

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- Denominación del Programa de Formación: Técnico en programación de Software
- Código del Programa de Formación: 233104-V2
- Nombre del Proyecto: DESARROLLO DE APLICACIONES DE SOFTWARE PARA EL SECTOR EMPRESARIAL. Fase del Proyecto: Diseño
- Actividad de Proyecto: Desarrollar el software de acuerdo con el diseño y a la plataforma de desarrollo.
- Competencia: 220501096 Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.
- Resultados de Aprendizaje Alcanzar: 669169 - Rap 1. Resolver procesos lógicos a través de la implementación de algoritmos y el lenguaje de programación seleccionado.
- Duración de la guía (24-horas)

2. PRESENTACIÓN

Una vez dominadas las estructuras secuenciales, el siguiente paso en la construcción de algoritmos es dotarlos de inteligencia para tomar decisiones y la eficiencia para repetir tareas. Las **estructuras de control** modifican el flujo de ejecución de un programa. Las **estructuras condicionales** ('Si-Sino', 'Según') permiten que un algoritmo elija entre diferentes caminos de acción basados en el cumplimiento de una condición lógica. Por otro lado, las **estructuras cíclicas o repetitivas** ('While', 'For') permiten ejecutar un bloque de instrucciones múltiples veces, ya sea un número determinado de veces o hasta que una condición deje de cumplirse. Se espera que el aprendiz adquiera la habilidad de traducir problemas simples a secuencias lógicas cíclicas de instrucciones.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividad de reflexión inicial.

Según la siguiente receta o lista de pasos secuenciales (algoritmo) para "Preparar Huevos Revueltos para una persona".

1. Romper 2 huevos en un tazón.
2. Agregar una pizca de sal.
3. Batir los huevos.
4. Calentar un sartén con aceite.
5. Verter los huevos en el sartén.
6. Cocinar por 3 minutos.
7. Servir en un plato.

Discusión 1. Discuta con todo el grupo los siguientes escenarios:

"Van a probar los huevos y se dan cuenta de que les falta sal. "

¿Qué parte del algoritmo tendrían que 'decidir' cambiar?

¿Qué harían si, por el contrario, ya estuvieran salados?"

SI (falta sal) **ENTONCES** (agregar un poco más).

Discusión 2. "El paso 3 dice 'batir los huevos'."

¿Es solo un movimiento y ya? ¿O lo hacen una y otra vez?

¿MIENTRAS los huevos no estén bien mezclados?

Discusión 3. "La receta es para una persona. Si tuvieran que hacer huevos para toda su familia (digamos, 4 personas):

¿repetirían todo el proceso 4 veces? ¿O qué harían de forma más inteligente?"

PARA cada una de las 4 personas, repetir ciertos pasos (como romper 2 huevos).

Realice la siguiente lectura <https://keepcoding.io/blog/que-son-estructuras-de-control-en-programacion/>.

Tiempo estimado: 25 minutos

3.2 Identificar los conocimientos necesarios para elaborar algoritmos con estructuras de control.

Las estructuras condicionales son la base de la toma de decisiones en un programa. Permiten ejecutar un bloque de código solo si se cumple una determinada condición.

Visualice el siguiente video:
https://www.youtube.com/watch?v=RaWfeVgkWbE&ab_channel=Cegamer

Preguntas para reflexionar:

1. ¿Cuál es la diferencia fundamental entre un condicional simple (SI...ENTONCES) y un condicional doble (SI...ENTONCES...SINO)?
2. Describe una situación de la vida real donde aplicarías un SI...ENTONCES...SINO.
3. En el diagrama de flujo, ¿qué representa la figura del rombo y por qué tiene dos salidas ("Verdadero" y "Falso")?

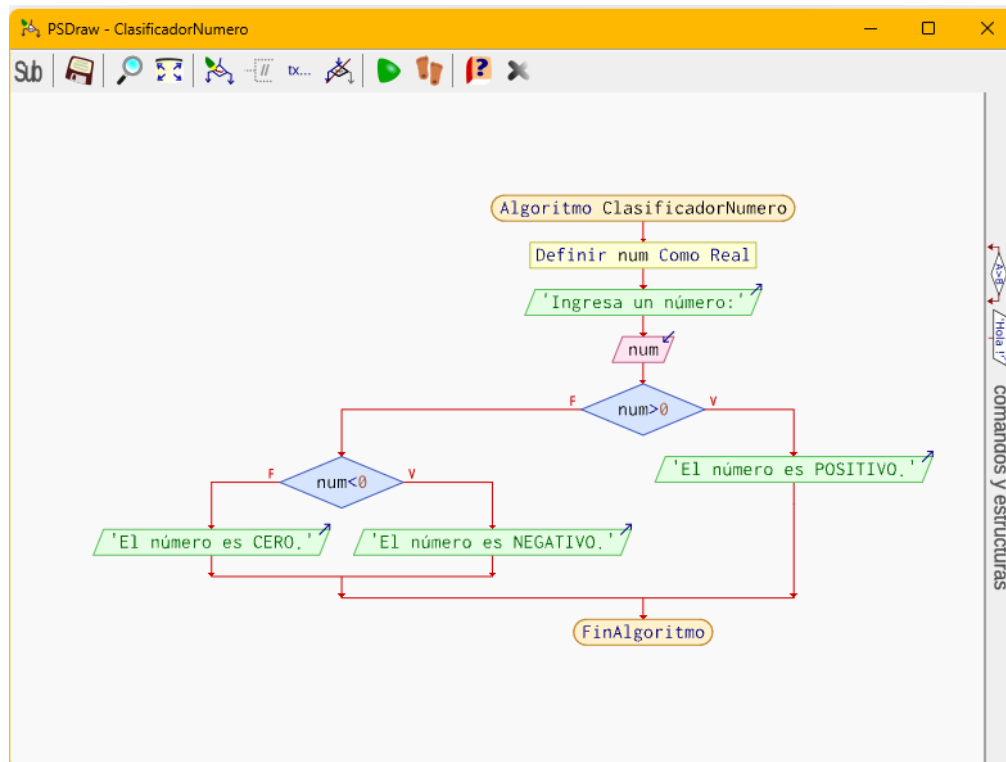
3.2.1 Actividades de apropiación del conocimiento (PSeInt)

Visualice el siguiente video:
https://www.youtube.com/watch?v=DIOBCZPFcaY&ab_channel=MarcosRios

- Siga el ejemplo e implemente en PSeInt el código para determinar si un aprendiz es o no mayor de edad.

3.2.1.1 Ejercicio Clasificador de Números

Objetivo: Implemente el siguiente diagrama de flujo que pide un número al usuario y determine si es positivo, negativo o cero.



2.2.1.2 Ejercicio Taquilla de Parque de Diversiones

Objetivo: Crea un algoritmo que calcule el precio de la entrada a un parque de diversiones. El programa debe solicitar la edad y la estatura (en metros) del visitante

Reglas de Precios:

- Menores de 4 años entran gratis.
- Entre 4 y 12 años, el precio base es de \$15.000. Pero, si miden menos de 1.30m, tienen un 10% de descuento.
- Mayores de 12 años pagan una tarifa única de \$25.000.

2.2.1.3 Ejercicio Recomendador de Menú

Objetivo: Diseña un algoritmo que funcione como un recomendador de menú para un restaurante. El programa debe hacer dos preguntas: "¿Qué tipo de comida prefieres? (Carne, Vegetariana, Pescado)" y "¿Cuál es tu presupuesto? (Bajo, Medio, Alto)".

Reglas de Recomendación:

- **Carne:**
 - Bajo: "Te recomendamos la Hamburguesa Clásica."
 - Medio: "Te recomendamos el Lomo a la Pimienta."
 - Alto: "Te recomendamos el Corte T-Bone importado."
- **Vegetariana:**
 - Bajo: "Te recomendamos la Ensalada del Chef."

- Medio: "Te recomendamos la Lasaña de Berenjenas."
- Alto: "Te recomendamos el Risotto de Champiñones y Trufa."
- **Pescado:**
- Bajo: "Te recomendamos los Tacos de Pescado."
- Medio: "Te recomendamos el Salmón a la Plancha."
- Alto: "Te recomendamos la Paella Marinera."

Si el usuario ingresa una opción no válida en cualquiera de las preguntas, el programa debe mostrar "Opción no válida en el menú."

3.2.1.4 Problema: Un profesor necesita un programa para analizar las calificaciones de un grupo de 5 aprendices. El programa debe hacer lo siguiente:

*Ingresar la calificación de cada aprendiz, por cada calificación ingresada, determinar si el estudiante "Aprobó" (calificación ≥ 3.0) o "Reprobó" (calificación < 3.0) y mostrar el mensaje inmediatamente.

Tiempo estimado: 4 horas.

3.3 Identificar los conocimientos necesarios para elaborar algoritmos con estructuras cíclicas condicionadas.

El ciclo "while" o ciclo "mientras" es un "**ciclo controlado por condición inicial**", lo que significa que el proceso se repite solo si una condición es verdadera, evaluándose esta antes de cada ejecución del cuerpo del ciclo.

Realice la siguiente lectura <https://dcodingames.com/el-ciclo-while/> y responda:

- ¿Cómo funciona el ciclo while?
- ¿Cuáles son las partes fundamentales del ciclo while?
- ¿Por qué es crucial que la variable de control cambie su valor dentro del cuerpo de un ciclo while?
- ¿Cuándo es más apropiado utilizar un ciclo while en comparación con otros tipos de ciclos como do-while?

3.3.1 Actividades de apropiación del conocimiento (PSeInt)

Visualice el siguiente video: https://www.youtube.com/watch?v=CNdd2np4zHE&ab_channel=MarcosRios

- Siga el ejemplo para crear un programa en PSeInt que simule un "login". Debe pedir una contraseña hasta que el usuario ingrese la correcta.

3.3.1.1 Ejercicio 2 Simulador de Cajero Automático

Objetivo: Diseña un algoritmo que simule un menú de cajero automático. El programa debe permitir al usuario consultar saldo, retirar dinero y salir.

- Reglas:
 1. El programa debe iniciar con un saldo de \$500.000.
 2. El menú principal se debe mostrar repetidamente mientras el usuario no elija la opción "Salir".
 3. Si el usuario elige "Retirar dinero", el programa debe pedir el monto a retirar. Se debe usar otro ciclo Mientras para validar que el monto ingresado no sea mayor al saldo disponible. Si lo es, debe mostrar un error y volver a pedir el monto.
 4. Una vez validado, se debe restar el monto del saldo y mostrar el nuevo saldo.

5. Si elige "Consultar saldo", solo debe mostrar el saldo actual.

3.3.1.2 Ejercicio 3 Juego de Adivinar el Número

Objetivo: Implementa un juego donde el programa "piensa" un número al azar entre 1 y 100, y el usuario debe adivinarlo.

- Reglas:
 1. El programa debe pedir un numero para adivinar.
 2. Usando un ciclo Mientras, el programa debe pedir al usuario que ingrese un número hasta que lo adivine.
 3. Después de cada intento, el programa debe dar una pista, indicando si el número secreto es "más alto" o "más bajo" que el número ingresado por el usuario.
 4. El programa también debe contar el número de intentos que le tomó al usuario adivinar.
 5. Cuando el usuario adivine, el programa debe felicitarlo y mostrarle cuántos intentos necesitó.

3.3.1.3 Problema: Un profesor necesita un programa para analizar las calificaciones de un grupo de 5 estudiantes. El programa debe hacer lo siguiente:

Usar un **ciclo Para** para pedir la calificación (de 0.0 a 5.0) de cada uno de los 5 estudiantes.

Tiempo estimado: 4 horas.

3.4 Identificar los conocimientos necesarios para elaborar algoritmos con cíclicas.

El ciclo "for" es una estructura de control de flujo repetitiva que se caracteriza por ser un **ciclo controlado por contador**. Esto significa que se sabe con exactitud cuántas veces se va a repetir una determinada tarea. Se define una variable de control (el contador) que se incrementa o decrementa de uno en uno, y las acciones dentro del ciclo se repiten hasta que esta variable alcanza un valor final predeterminado. Aunque la palabra "for" se traduce como "para", el autor sugiere que "desde" es una traducción más adecuada para su funcionamiento en pseudocódigo, ya que describe la repetición de acciones "desde" un valor inicial "hasta" un valor final.

Realice la siguiente lectura <https://dcodingames.com/el-ciclo-for/> y responda:

- ¿Qué tipos de datos puede tener la variable de control en un ciclo for?
- ¿Qué tipos de valores se pueden usar como límites (inicial y final) en un ciclo for?
- ¿Por qué no se debe modificar la variable de control dentro del cuerpo del ciclo for?
- ¿Qué sucede si se termina el encabezado del ciclo for con un punto y coma (;)?

3.4.1 Actividades de apropiación del conocimiento (PSeInt)

Visualice el siguiente video:

https://www.youtube.com/watch?v=3G7AXyLvRuo&ab_channel=MarcosRios

3.4.1.4 Problema: Un profesor necesita un programa para analizar las calificaciones de un grupo de 5 estudiantes. El programa debe hacer lo siguiente:

Usar un **ciclo Para** para pedir la calificación (de 0.0 a 5.0) de cada uno de los 5 estudiantes.

Tiempo estimado: 4 horas.

4. ACTIVIDADES DE EVALUACIÓN (consolidación de conocimientos)

De forma individual diseñe el pseudocódigo, el diagrama de flujo y posteriormente implementa en PSeInt el algoritmo que solucione el siguiente problema:

Analizador de Calificaciones de Estudiantes

Problema: Un profesor necesita un programa para analizar las calificaciones de un grupo de 5 estudiantes. El programa debe hacer lo siguiente:

1. Usar un **ciclo Para** para pedir la calificación (de 0.0 a 5.0) de cada uno de los 5 estudiantes.
2. Dentro del ciclo, por cada calificación ingresada, debe usar una **estructura condicional (Si...Entonces...SiNo)** para determinar si el estudiante "Aprobó" (calificación ≥ 3.0) o "Reprobó" (calificación < 3.0) y mostrar el mensaje inmediatamente.
3. El programa debe contar cuántos estudiantes aprobaron y cuántos reprobaron.
4. El programa debe calcular la suma total de todas las calificaciones (usando un acumulador).
5. Al finalizar el ciclo, el programa debe mostrar en pantalla:
 - El número total de estudiantes que aprobaron.
 - El número total de estudiantes que reprobaron.
 - El promedio de calificación de todo el grupo (Suma de calificaciones / 5).

4.1 Actividad de Evaluación Final (Proyecto): Simulador de Venta de Boletos de Cine

En parejas, diseñe el pseudocódigo, el diagrama de flujo y la implementación en PSeInt para el siguiente proyecto.

Problema: Un cine local necesita un sistema para su taquilla que permita vender boletos y llevar un registro de las ventas del día.

Funcionalidades Requeridas:

1. **Menú Principal Interactivo:**
 - El programa debe mostrar un menú con tres opciones:
 1. Vender Entradas
 2. Ver Reporte de Ventas
 3. Salir
2. **Módulo "Vender Entradas":**
 - Al elegir esta opción, el sistema debe preguntar: "¿Cuántas entradas desea comprar?".
 - Para cada entrada, debe solicitar la **edad** del cliente.
 - El precio de la entrada se determina:
 - Menores de 12 años (niños): \$7.000
 - Entre 12 y 60 años (adultos): \$10.000
 - Mayores de 60 años (tercera edad): \$5.000 (descuento especial)
 - El sistema debe calcular el costo total de la transacción actual y mostrarlo.
3. **Módulo "Ver Reporte de Ventas":**
 - Al elegir esta opción, el sistema debe mostrar en pantalla:
 - Cantidad total de entradas vendidas (sumando todas las transacciones).
 - Cantidad de entradas vendidas por categoría (niños, adultos, tercera edad).
 - Dinero total recaudado.
4. **Módulo "Salir":**
 - Debe mostrar un mensaje de despedida como "Cerrando sistema. ¡Hasta pronto!" y finalizar el programa.

Variables Para Considerar:

- Necesitarás **acumuladores** para el total de dinero y para el total de entradas vendidas.
- Necesitarás **contadores** para cada categoría de entrada (niños, adultos, tercera edad).

Tiempo estimado: 8 horas.

5. GLOSARIO DE TERMINOS

Estructura de Control (Condicional): Instrucción que permite alterar el flujo de ejecución de un programa basándose en la evaluación (Verdadero o Falso) de una condición. La principal es Si...Entonces...SiNo (If...Then...Else).

Estructura Cíclica (Bucle o Iteración): Construcción que permite la ejecución repetida de un bloque de código. Las principales son Mientras (While) y Para (For).

Condición: Expresión lógica (que utiliza operadores relacionales como >, <, ==) que, al ser evaluada, produce un resultado booleano (Verdadero o Falso), determinando el flujo del programa.

Bucle Mientras (While): Ejecuta un bloque de código repetidamente *mientras* una condición especificada se mantenga verdadera. Su número de iteraciones es indeterminado.

Bucle Para (For): Ejecuta un bloque de código un número predefinido de veces. Utiliza una variable de control (contador) que se incrementa o decrementa en cada iteración.

Contador: Variable, típicamente de tipo entero, utilizada para llevar la cuenta del número de veces que se ha ejecutado un bucle.

Acumulador: Variable utilizada para almacenar la suma o el resultado consolidado de una serie de valores a lo largo de las iteraciones de un bucle.

Bucle Infinito: Un ciclo cuya condición de terminación nunca se cumple, provocando que el programa se ejecute indefinidamente y se bloquee. Es un error de lógica común.