

➤ **MANUAL DE PRÁCTICA**

Resolución de problemas complejos
utilizando algoritmos

Introducción

En esta actividad, pondrás en práctica tus habilidades para la solución de problemas utilizando algoritmos. Para ello, te plantearemos situaciones ordinarias que encontrarás cuando escribas código.

Por principio de cuentas, deberás analizar el problema. Luego, será necesario descomponerlo en fracciones más pequeñas y diseñar algoritmos que resuelvan cada una de esas secciones. Será oportuno utilizar diferentes estructuras de datos y algoritmos, hasta llegar a la solución total del problema.

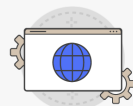
Empezaremos con ejercicios sencillos que te ayudarán a poner en práctica los conocimientos adquiridos. Paso a paso llegaremos a planteamientos más complejos, donde será necesario activar tu pensamiento lógico y tu creatividad.

Al igual que en la Unidad 3, utilizarás el software PseInt para la codificación. El objetivo es que llegues al resultado esperado. En otro momento, revisarás los algoritmos que has desarrollado y trabajarás en nuevas modificaciones para que se ejecuten con mayor eficiencia. ¡Empecemos!

Recursos requeridos



Computadora portátil o de escritorio con procesador mínimo Intel Core i3 o equivalente, 8 GB de memoria RAM, 100 MB disponibles de almacenamiento. El sistema operativo puede ser Windows, Linux o MacOS con conexión a internet.



PSeInt (sourceforge.net)

<https://pseint.sourceforge.net/>



Lápiz y hoja de papel para planear el algoritmo si así lo deseas.

Antes de iniciar, es necesario que identifiques los problemas que vas a resolver a lo largo de esta práctica.



Desarrollo de la práctica

1. **Lee** detenidamente cada problema, pues irán avanzando en su nivel de dificultad. El propósito, es que te vayas adaptando a la herramienta y su semántica.
2. **Utiliza** tu pensamiento crítico para idear un plan de trabajo que te conduzca a la solución de los problemas. Es importante tener en cuenta todas las restricciones que se presentan según su planteamiento. Aquí puedes utilizar lápiz y papel para anotar los macroprocesos que después deberás desarrollar en algoritmos.
3. **Lanza** la aplicación PSeInt para desarrollar uno a uno los algoritmos que darán solución a los siguientes problemas, justo como lo hiciste en la práctica de la Unidad 3.
4. **Ejecuta** un algoritmo a la vez, siguiendo la secuencia y realiza pruebas que satisfagan al menos los casos propuestos en cada problema.

Problema 1

Duplicados

Debes encontrar valores duplicados en un arreglo de números enteros y regresar un arreglo con los números que se repitan exactamente 2 veces.



Notas

Prueba tu algoritmo para los siguientes casos:

- $[4,3,2,7,8,2,3,1] = [3,2]$
- $[1,1,2] = [1]$
- $[1] = []$

Problema 2

Cadena a la inversa

Regresa una cadena de texto en sentido inverso.



Notas

- El programa debe solicitar una cadena de texto al usuario
- Prueba tu algoritmo con las siguientes cadenas:
 - galleta = atellag
 - procesador = rodasecorp
 - vehículo = olucíhev
- Una vez que hayas conseguido lo anterior, agrega una validación para regresar Verdadero si la palabra es un palíndromo. Prueba tu algoritmo con los valores:
 - rotor = rotor
 - sometemos = sometemos
 - reconocer = reconocer

Problema 3

Calculadora de edad

Trabajas en una compañía que ofrece seguros de vida y de gastos médicos. La empresa está desarrollando una herramienta tecnológica, para que los agentes puedan calcular el monto de una póliza. Para dicho cálculo, se consideran varios factores de riesgo, entre los que se encuentra

la edad. Por ello, el contratante debe llenar un formulario, incluyendo su fecha de nacimiento. Como primer reto, deberás calcular la edad del cliente al momento de la cotización.



Notas

- La captura de la fecha de nacimiento debe ser en una sola cadena de texto con formato DD-MM-AAA
- Debes validar que es una fecha válida; para simplificar el algoritmo tomaremos el día 29 de febrero como inválido
- El año de nacimiento debe ser mayor a 1900
- No olvides validar si el cliente ya cumplió años en la fecha de captura o aún no

Problema 4

Dispensadora de billetes

Has sido contratado por un banco que está en un proceso de actualización de su red de cajeros automáticos. Tu trabajo es desarrollar el algoritmo con el cual va a operar la dispensadora de billetes. Para que el cajero realice esta tarea, deberás considerar las siguientes premisas:



- Entregar siempre la menor cantidad de billetes
- Mantener un inventario que indique, cuál es la cantidad de billetes disponible para cada denominación
- Las denominaciones disponibles son 20, 50, 100, 200, 500 y 1000
- Si el inventario no cuenta con una combinación de billetes, suficiente para satisfacer el importe solicitado, no dispensará ningún billete
- Para efectos de esta práctica, siempre que inicie el algoritmo, deberá haber en inventario, 10 billetes de cada denominación

Problema 5

Máquina expendedora

Eres un emprendedor entusiasta y tienes la oportunidad de instalar una máquina expendedora de botanas en un auto lavado cercano a tu casa. Te corresponde desarrollar el algoritmo para que cuente las monedas que el usuario va insertando, sumando el importe en una variable hasta que el usuario introduzca un código de algunos de los productos que están disponibles. debes validar que el importe pagado sea suficiente. Hay que tener en cuenta las siguientes restricciones:



- El algoritmo deberá aceptar los siguientes comandos:
 - [NÚMERO]. Será un valor numérico que representa la moneda que está entrando al sistema
 - [IDENTIFICADOR]. Será una clave para identificar los productos que están disponibles con el formato [LETRA][NÚMERO], por ejemplo, D4, A3, etc.
 - RETIRAR. Si se recibe este comando, se devolverá el importe al usuario cancelando la operación.
 - PARA. Este comando será para detener la ejecución
- La máquina sólo acepta monedas de 1, 2, 5, 10 y 20 pesos. Cualquier otra moneda que entre al sistema será rechazada
- No te preocupes por llevar el detalle de las monedas que el cliente ha introducido, sólo el importe.
- En este problema, no ocupas tomar en cuenta el inventario disponible de cada producto
- El algoritmo debe reiniciarse cada vez que se haya dispensado un producto y deberá mantenerse corriendo hasta que sea introducido el comando para salir
- Intenta dividir tu programa en funciones para que tu código sea más comprensible