

**UNIVERSIDAD ESTATAL DE BOLIVAR**

**NIVELACIÓN SOFTWARE**



**TEMA: ANALISIS DE UN PROBLEMA**

MATERIA: ALGORITMOS Y LÓGICA DE PROGRAMACIÓN

INTEGRANTES:

* ARIEL ALEJANDRO CALDERON
* ANGELES XIOMARA PUNINA

**INTRODUCCIÓN**

La resolución de problemas mediante algoritmos es fundamental en el campo de la programación y la informática. En este contexto, el diseño de algoritmos eficientes y estructurados es esencial para abordar situaciones cotidianas que requieren una solución lógica y precisa. En este documento, se abordará el desarrollo de un algoritmo para calcular el cambio a entregar a un cliente, minimizando el número de monedas utilizadas. Siguiendo los pasos propuestos por el Msc. Marcelo Baño, se analizará el problema, se diseñará una solución lógica, se construirá un diagrama de flujo y se implementará el algoritmo en un formato cercano al código. Este proceso permitirá comprender la importancia de seguir una estructura definida para la resolución de problemas mediante algoritmos, brindando una base sólida para el desarrollo de soluciones informáticas efectivas.

**PROBLEMA** **PLANTEADO**

Disponemos de una máquina que puede dar cambio con 5 tipos de billetes distintos: 20, 10, 5, 2 y 1. Dado el precio del artículo y la cantidad entregada por el consumidor indicar el cambio a entregar empleando el menor número posible de monedas.

**ENTENDER EL PROBLEMA**

**• Define claramente el problema que estás tratando de resolver.**

En este caso, el problema es calcular la vuelta a entregar al cliente utilizando el menor número posible de monedas, dados el precio del artículo y la cantidad entregada por el consumidor.

**• Identifica las entradas (datos de entrada) y las salidas (resultados esperados).**

Valor de cantidad entregada y valor del objeto o articulo

Valores de salida serian la cantidad de billetes que hay que entregar por cada denominación.

**ANALISIS DEL PROBLEMA**

**• Descompón el problema en partes más pequeñas y manejables.**

En este caso, se puede dividir el problema en los siguientes pasos:

1. Calcular el cambio a entregar al cliente.
2. Determinar la cantidad de monedas necesarias para dar el cambio.
3. Seleccionar las monedas adecuadas para dar el cambio.

**• Identifica los pasos o procesos principales necesarios para resolver el problema.**

* Realizamos la resta para calcular el vuelto
* Seccionamos el problema para realizar una división entre cada billete y el vuelto
* El sobrante se seguirá dividiendo para los billetes hasta que nos quede un valor de cero

**DISEÑAR LA SOLUCIÓN**

**• Selecciona las estructuras de datos y algoritmos apropiados para resolver cada parte del problema.**

**Estructuras de datos**

* **Variables enteras:** Utiliza variables para almacenar el precio, la cantidad entregada, el vuelto y la cantidad de cada tipo de billete.

**Algoritmos:**

* **Resta simple:** Calcula el vuelto restando el precio de la cantidad entregada.
* **División entera y módulo:** Utiliza la división entera y el módulo (% en muchos lenguajes) para obtener la cantidad de billetes de cada denominación.
* **Bucle o estructuras condicionales:** Puedes utilizar un bucle o estructuras condicionales para verificar cuántos billetes de cada denominación son necesarios.

**• Decide sobre la secuencia de pasos que compondrán tu algoritmo.**

* Definir variables
* Entrada de datos
* Cálculo del vuelto
* Cálculo de billetes
* Salida de resultados

**ESCRIBIR EL PSEUDOCÓDIGO**

**•Traduce tu diseño a un formato más cercano al código:**

- Leer el precio del artículo y la cantidad entregada por el cliente. - Calcular el cambio a entregar al cliente. - Determinar la cantidad de monedas necesarias para dar el cambio. - Seleccionar las monedas adecuadas para dar el cambio. – Imprimir la cantidad de monedas necesarias para dar el cambio.

**REVISAR Y REFINAR**

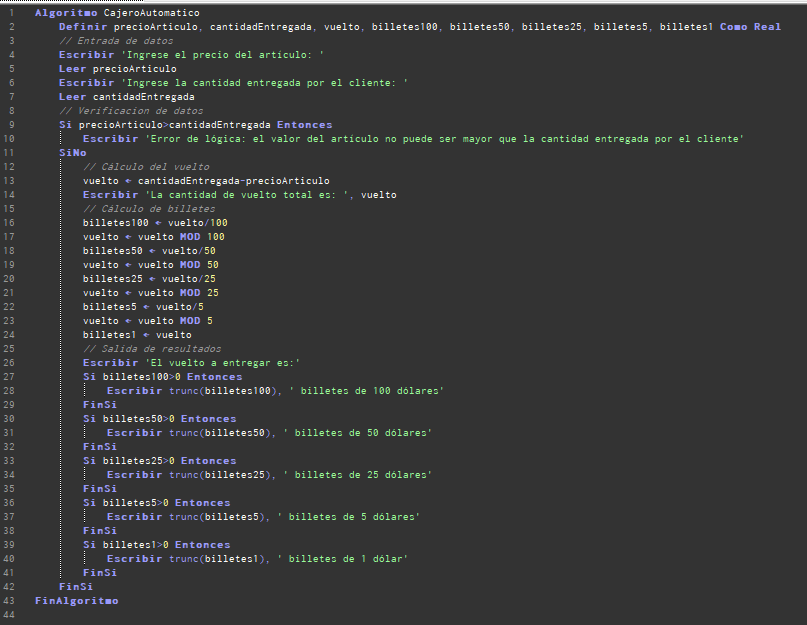
En nuestro caso, hubo que agregar ciertas funcionalidades:

* Una operación extra al momento de entregar el resultado ya que cada billete debería ser representado solo por enteros y no por sus decimales (derivados del vuelto que queda por calcular). La función **trunc( )** nos ayudara con el problema ya que elimina la parte decimal de un valor**.**
* Una verificación de datos ingresados. El valor del articulo **siempre será menor** a la cantidad entregada por el cliente, de lo contario no existiría vuelto alguno.

**CONSTRUIR EL DIAGRAMA DE FLUJO**



**IMPLEMENTAR EL CÓDIGO**



**PROBAR Y DEPURAR**

