INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Tecnologia de Objetos				
TÍTULO DE LA PRÁCTICA:	*Guia 2*				
NÚMERO DE PRÁCTICA:	*2*	**AÑO LECTIVO:**	*2025-B*	**NRO. SEMESTRE:**	*VI*
FECHA DE PRESENTACIÓN	*20/09/25*	**HORA DE PRESENTACIÓN**	*8:40*		
INTEGRANTE (s): Alvaro Suasaca Pacompia				**NOTA:**	
DOCENTE(s): *Edith Cano*					

SOLUCIÓN Y RESULTADOS

I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

Problema propuesto:

Vuelto. Contando el vuelto o cambio.
Escriba una función recursiva que cuente de cuántas maneras diferentes puede dar cambio para una determinada cantidad y de acuerdo a una lista de denominaciones de monedas.
Por ejemplo, hay 3 maneras de dar cambio si la cantidad=4 si tienes monedas con denominación 1 y 2:
1+1+1+1,
1+1+2,
2+2.
Realiza este ejercicio implementando la función countChange en Scala.
Esta función toma una cantidad a cambiar y una lista de denominaciones únicas para las monedas.
Su definición es la siguiente:
def countChange(money: Int, coins: List[Int]): Int
Puedes usar las funciones isEmpty, head y tail en la lista de monedas enteras.

main.scala — Share — Run — Output

```scala
// Online Scala Editor for free
// Write, Edit and Run your Scala code using Scala Online Compiler

object JDoodle {
  import scala.collection.mutable

  // Creamos un cache para guardar repetidos
  val memo: mutable.Map[(Int, List[Int]), Int] = mutable.Map()

  def contarFormas(monedas: List[Int], total: Int): Int = {
    memo.clear()
    contarDesde(monedas, total)
  }
```

```
N?mero de formas de obtener 20: 29

=== Code Execution Successful ===
```

main.scala — Share — Run — Output

```scala
    // clave
    val clave = (restante, monedas)
    // verificar si ya existe en nuestro map
    if (memo.contains(clave)) return memo(clave)
    val usarHead = contarDesde(monedas, restante - monedas.head)
        // restamos a la moneda actual
    val sinHead  = contarDesde(monedas.tail, restante)
        // pasamos a la sgte moneda

    val totalFormas = usarHead + sinHead
    memo(clave) = totalFormas
    totalFormas
  }

  def main(args: Array[String]): Unit = {
    val monedas = List(1, 2, 5)
    val total = 20

    val formas = contarFormas(monedas, total)
    println(s"Número de formas de obtener $total: $formas")
```

```
N?mero de formas de obtener 20: 29

=== Code Execution Successful ===
```

II. SOLUCIÓN DEL CUESTIONARIO

1. Explique el caso base implementado.

Para el problema del cambio de monedas primero se penso en que de una manera no deban de repetirse el vuelto asi que se hizo un map donde se guarda la informacion del vuelto con la clave de el restante y las monedas luego para los casos de pare o base tenemos cuando ya llego a 0, cuando se haya pasado del vuelto para no tener numero negativos y por si no hay monedas con la cual hacer el vuelto luego creamos una clave la cual nos ayudara para guardarlo en nuestro map, luego tenemos el llamado a nuestra recursion uno es en el cual usamos el valor actual y le restamos el head y llamamos a nuestra recursion y la otra llamada es cuando usamos la sgte moneda no restamos nada pues estamos usando otra moneda, por ultimo sumamos luego las dos recursiones guardamos el resultado con su clave y devolvemos el valor de la funcion.

2. ¿Este problema se resuelve mejor en el modelo de programación funcional?

A mi parecer si pues ya que es de manera recursiva el codigo debe de ser más limpio evitando los errores al momento de hacer las llamadas nuevamente de la funcion

3. ¿Qué beneficios se podrían obtener para este problema utilizando memoización?

Es de gran ayuda pues con esta ya tenemos en cuenta los casos que ya se resolvio o tenemos evitando la redundancia y obteniendo una mejor lista de resultados los cuales no se repiten.

III. CONCLUSIONES

- Bueno el lenguaje scala es nuevo para mi pero tiene muchos parecidos con otros lenguajes como python pues segun su definicion todo es un objeto, tambien el parecido con algunos lenguajes como c++ y java los cuales tambien estan orientado a objetos, ademas de que nos dicen que su compilador puede interpretar las clases java y utilizas su biblioteca y herramientas de java.

RETROALIMENTACIÓN GENERAL