

NOMBRES: NICOLÁS PACHECO, MARCELO HENRIQUEZ
Actividad en Clase: Análisis de código y Dependencia

i. Identifique las clases y lo que éstas representan. Luego, establezca una descripción textual breve del contexto problema.

R: Hay dos clases, una llamada **Calculadora** y la otra llamada **CarroCompra**.

ii. Analice los atributos y métodos de cada clase, luego, identifique las relaciones existentes entre las clases identificadas y establezca una descripción textual breve del contexto problema..

R: Análisis de atributos y métodos:

Clase Calculadora

Atributos	Métodos
n1: int n2: int	Calculadora() Calculadora(num1: int, num2: int) sumar(): int multiplicar(): int setN1(num1: int): void setN2(num2: int): void

Clase CarroCompra

Atributos	Métodos
productos: int[][]	CarroCompra() calcularTotal(): int subTotal(cant: int, precio: int): int mostrarTotal(): void

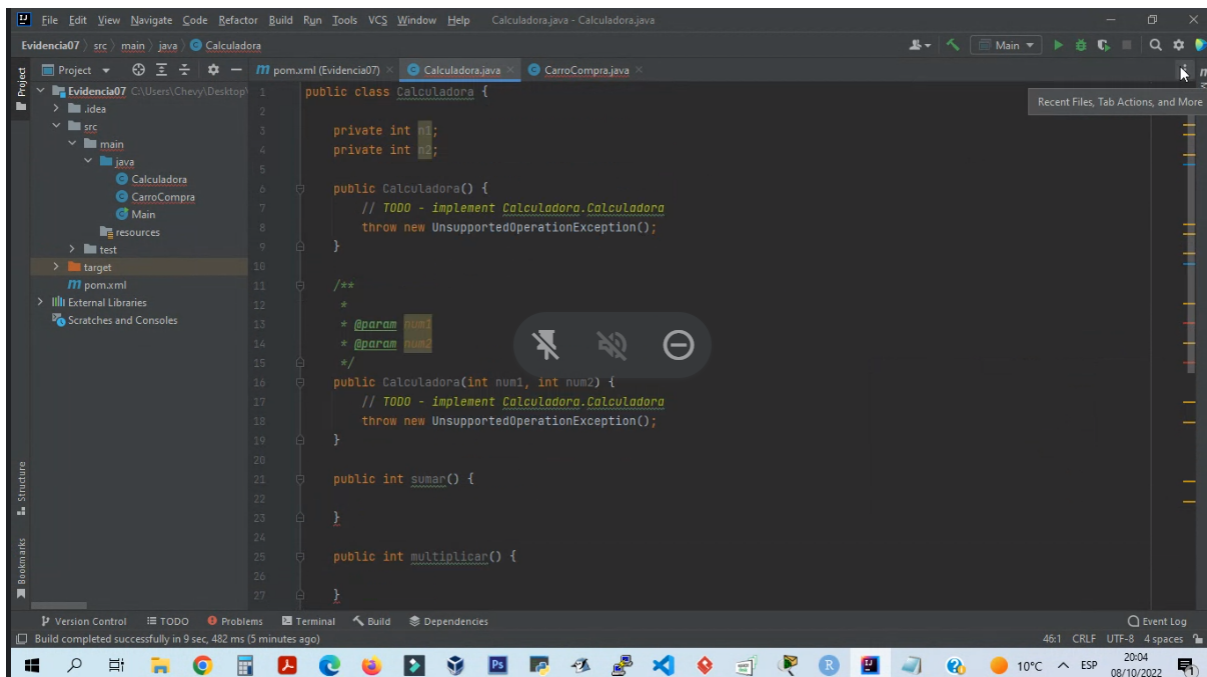
R: Relación entre Clases y descripción: Existe dependencia entre las dos clases. En la clase CarroCompra se crea un objeto perteneciente a la clase Calculadora para utilizar el método multiplicar().

iii. De lo anterior, establezca una representación detallada del código fuente, usando un diagrama de clases UML y la herramienta de modelado Visual Paradigm.



iv. Genere un código fuente Java a partir de su modelo de clases.

General (posteriormente agregamos una clase main para poder probar el código):



Clase CarroCompra:

```
1 public class CarroCompra {
2
3     private int[][] productos;
4
5     public CarroCompra() {
6         // TODO - implement CarroCompra.CarroCompra
7         throw new UnsupportedOperationException();
8     }
9
10    public int calcularTotal() {
11
12    }
13
14    /**
15     *
16     * @param cant
17     * @param precio
18     */
19    public int subTotal(int cant, int precio) {
20
21    }
22
23    public void mostrarTotal() {
24        // TODO - implement CarroCompra.mostrarTotal
25        throw new UnsupportedOperationException();
26    }
27 }
```

Clase Calculadora:

```
1  public class Calculadora {
2
3      private int n1;
4      private int n2;
5
6      public Calculadora() {
7          // TODO - implement Calculadora.Calculadora
8          throw new UnsupportedOperationException();
9      }
10
11     /**
12      *
13      * @param num1
14      * @param num2
15      */
16     public Calculadora(int num1, int num2) {
17         // TODO - implement Calculadora.Calculadora
18         throw new UnsupportedOperationException();
19     }
20
21     public int sumar() {
22
23     }
24
25     public int multiplicar() {
26
27     }
```

```
29     /**
30      *
31      * @param num1
32      */
33     public void setN1(int num1) {
34         this.n1 = num1;
35     }
36
37     /**
38      *
39      * @param num2
40      */
41     public void setN2(int num2) {
42         this.n2 = num2;
43     }
44 }
```

Ejecucion

```
1  ▶ public class Main {  
2  ▶      public static void main(String[] args) {  
3      CarroCompra carroCompra = new CarroCompra();  
4      carroCompra.mostrarTotal();  
5      }  
6  }
```

Main ×

↑ <C:\Users\Chevy\.jdk\openjdk-17.0.2\bin\java.exe> "-javaag
↓ El total de la compra es: \$5000
= Process finished with exit code 0
-1