



CÓDIGO: \_\_\_\_\_ NOMBRE: \_\_\_\_\_

**PARTE TEÓRICA (20%)**

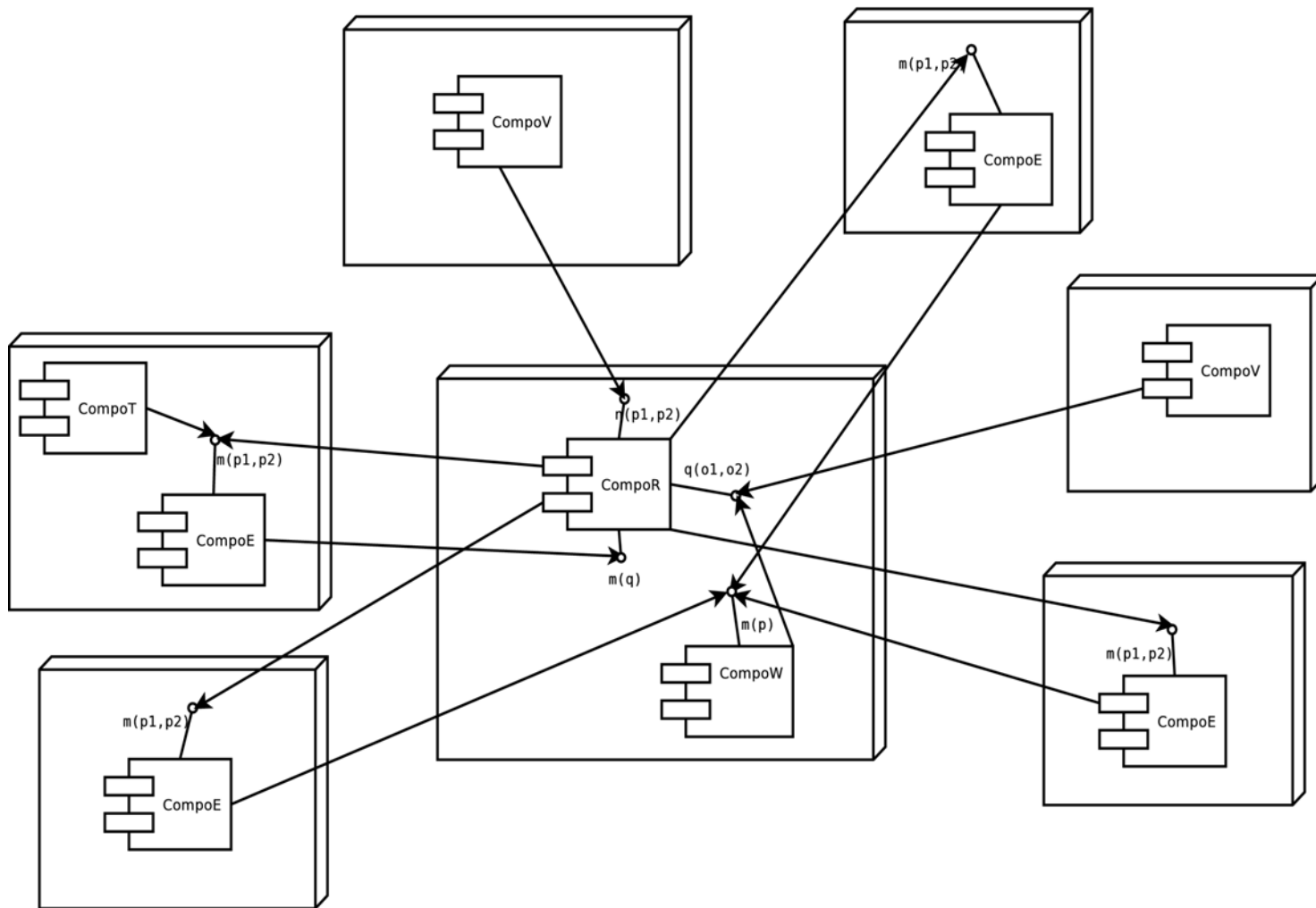
1. (15%) Explique cuáles son las dos mejores formas tanto de cohesión como de acoplamiento. Sustente su respuesta.

2. (5%) Explique por qué la cohesión y el acoplamiento están inversamente relacionadas entre ellas.

**PARTE APLICADA (80%)**

1. (40%) Para el diagrama UML presentado en la siguiente figura, explique:

- a) ¿Qué estilo de arquitectura implementa?
- b) ¿Cómo es el flujo de control entre los componentes?
- c) ¿Cómo es la transferencia de datos entre los componentes?



2. (40%) La oficina del tránsito del municipio de Santiago de Cali ha solicitado el desarrollo de una funcionalidad en java que permita calcular el valor a pagar por concepto de impuesto automotor de un vehículo. Para el cálculo del impuesto, el usuario debe suministrar la información del vehículo y la fecha en la que desea realizarse el pago. El sistema debe verificar que la placa del vehículo corresponde al rango de placas que puede atenderse el día de la semana que corresponde a la fecha suministrada. El valor del impuesto se calcula con base en un porcentaje que se aplica al valor comercial del vehículo, según su modelo.

El siguiente código es la primera aproximación a la solución que ha desarrollado un programador desprevenido. Note que la tabla de los porcentajes para el cobro del impuesto y la clasificación de días por número de placa están definidos en el código\* y explicados en los comentarios. Por lo tanto, la especificación de los casos de prueba debe realizarse a partir de esta definición.

- a) Usando las técnicas de prueba de valor límite, tabla de decisión y partición de equivalencia, diseñe los casos de prueba que permitan establecer la correctitud del código si se quisiera cubrir la totalidad de los casos que pueden presentarse en la ejecución del código (ver \*).
- b) Sobre el texto del código, indique en palabras las modificaciones que tendrían que hacerse al método del cálculo de impuesto, de manera que maneje correctamente todos los casos que se pueden presentar en su ejecución.

```

package paquete;

public class Vehiculo {

    private String numPlaca;

    private String ciudad;

    private int modelo;

    private double valorComercial;

    public Vehiculo(String nPlaca, String city, int model, double valor)
    {
        numPlaca=nPlaca;

        ciudad=city;

        modelo=model;

        valorComercial=valor;
    }

    public String getNumPlaca()
    {
        return numPlaca;
    }

    public int getModelo()
    {
        return modelo;
    }

    public double getValorComercial()
    {
        return valorComercial;
    }

    public boolean equals(Vehiculo auto)
    {
        if(numPlaca.equals(auto.getNumPlaca()))

            return true;

        return false;
    }

    public int hashCode()
    {
        return (Integer.getInteger(numPlaca)).intValue();
    }
}

```

```

package paquete;
import java.util.*;
import java.text.*;
public class Transito {

    //Matriz que almacena el límite inferior y superior de los rangos de
    placas(filas) para cada día de la semana(columnas)
    private static String[][]
    limitesPlaca={{ "000", "223", "445", "667", "889" }, {"222", "444", "666", "888", "999"
    }};

    //Vector con los códigos de los días de la semana que usa el sistema
    private static String[] diasSemana={"mon", "tue", "wed", "thu", "fri"};

    //Vector con los tres porcentajes de impuesto que se aplican según el modelo
    private static double[] porcImpuesto={0.1,0.08,0.05};

    private static int[] limitesModelo={1990,2000};

    public static void main(String[] args) {

        Vehiculo auto=new Vehiculo("395", "Cali", 2005, 15000000);

        double valorImpuesto=calcularImpuesto(auto,"05/12/2009");

        System.out.println("El valor a pagar por el impuesto del vehículo es
        $" +valorImpuesto);

    }
    public static double calcularImpuesto(Vehiculo auto, String fechaPago)
    {
        double porcentaje=0.0;
        try {

            DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);

            Date fecha;

            fecha = df.parse(fechaPago);

            SimpleDateFormat formato = new SimpleDateFormat("E");

            String dia=formato.format(fecha);

            System.out.println(dia);

            for(int cont=0;cont<diasSemana.length;cont++)
            {
                if(dia.equalsIgnoreCase(diasSemana[cont]))
                {
                    if (auto.getNumPlaca().compareTo(limitesPlaca[0][cont])>=0
                    && auto.getNumPlaca().compareTo(limitesPlaca[1][cont])<0)
                    {
                        if (auto.getModelo()<=limitesModelo[0])
                        {
                            porcentaje=porcImpuesto[0];
                        }
                        else
                        {
                            if(auto.getModelo()<=limitesModelo[1])
                            {
                                porcentaje=porcImpuesto[1];
                            }
                            else
                            {
                                porcentaje=porcImpuesto[2];
                            }
                        }
                    }
                }
            }
        } catch (ParseException e) {
            e.printStackTrace();
        }

        return auto.getValorComercial()*porcentaje;
    }
}

```