# Universidad Nacional de San Agustín
# Escuela Profesional de Ingeniería de Sistemas
# Fundamentos de Programación II
# Tema N° 09:
# Programación Orientada a Objetos III

**Nombre: Jhonatan Benjamin Mamani Céspedes** **CUI: 20232188**

**Link de repositorio GitHub:** https://github.com/JBenjamin01/fp2-24b

## Ejercicio 1:

- Crear la clase Coordenada, posición en el plano (x, y), con su constructor y métodos necesarios
- 3 formas de calcular la distancia entre 2 coordenadas (distancia euclidiana)
  - Método en clase principal
  - Método en Coordenada con 2 parámetros
  - Método en Coordenada con 1 parámetro

### Clase Coordenada.java:

```java
public class Coordenada {
    private int x;
    private int y;

    public Coordenada(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public double calcularDistancia2(int x2, int y2) {
        return Math.sqrt(Math.pow((x2 - x), 2) + Math.pow((y2 - y), 2));
    }

    public double calcularDistancia3(Coordenada c) {
        return Math.sqrt(Math.pow((c.getX() - this.x), 2) + Math.pow((c.getY() - this.y), 2));
    }
}
```

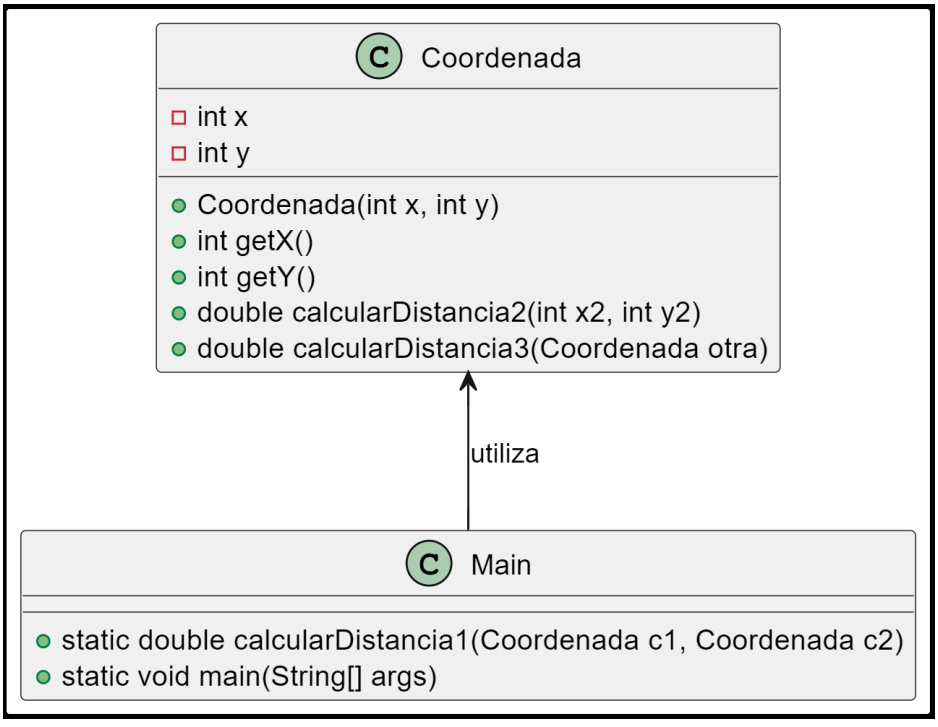
## Clase Main.java:

```java
public class Main {
    public static void main(String[] args) {
        Coordenada p1 = new Coordenada(11, 31);
        Coordenada p2 = new Coordenada(5, 16);

        double d1 = calcularDistancia1(p1, p2);
        System.out.println("Distancia (con el método en clase principal): " + d1);


        double d2 = p1.calcularDistancia2(5, 16);
        System.out.println("Distancia (con el método con 2 parámetros): " + d2);

        double d3 = p1.calcularDistancia3(p2);
        System.out.println("Distancia (con el método con 1 parámetro): " + d3);
    }

    public static double calcularDistancia1(Coordenada c1, Coordenada c2) {
        return Math.sqrt(Math.pow((c2.getX() - c1.getX()), 2) + Math.pow((c2.getY() - c1.getY()), 2));
    }
}
```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de S
& 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExcep
5fa918254aa1b6a5\redhat.java\jdt_ws\Ejercicio 1_da117150\bin' 'Main'
Distancia (con el método en clase principal): 16.15549442140351
Distancia (con el método con 2 parámetros): 16.15549442140351
Distancia (con el método con 1 parámetro): 16.15549442140351
```

## Diagrama UML:



## Ejercicio 2:

- Diagrama de clases UML y programa que utilice objetos de clases creadas
- Crear la clase Fracción. Constructores sobrecargados usando llamadas a `this` entre ellos.
- Implementar las operaciones de suma, resta, multiplicación, división y simplificación.
- 3 formas: método en clase principal, método de clase, método de instancia para cada operación. Evitar duplicación de código.

### Clase Fraccion.java:

```java
public class Fraccion {
    private int numerador;
    private int denominador;

    public Fraccion() {
        this(0, 1);
    }

    public Fraccion(int numerador) {
        this(numerador, 1);
    }

    public Fraccion(int numerador, int denominador) {
        if (denominador == 0) {
            throw new IllegalArgumentException("El denominador no puede ser cero");
        }
        this.numerador = numerador;
        this.denominador = denominador;
        simplify();
    }

    public int getNumerador() {
        return numerador;
    }

    public int getDenominador() {
        return denominador;
    }

    private void simplify() {
        int mcd = mcd(numerador, denominador);
        numerador /= mcd;
        denominador /= mcd;
    }
```

```java
    private int mcd(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    // Métodos de instancia
    public Fraccion sumar(Fraccion f) {
        int newNumerador = this.numerador * f.denominador + f.numerador * this.denominador;
        int newDenominador = this.denominador * f.denominador;
        return new Fraccion(newNumerador, newDenominador);
    }

    public Fraccion restar(Fraccion f) {
        int newNumerador = this.numerador * f.denominador - f.numerador * this.denominador;
        int newDenominador = this.denominador * f.denominador;
        return new Fraccion(newNumerador, newDenominador);
    }

    public Fraccion multiplicar(Fraccion f) {
        int newNumerador = this.numerador * f.numerador;
        int newDenominador = this.denominador * f.denominador;
        return new Fraccion(newNumerador, newDenominador);
    }

    public Fraccion dividir(Fraccion f) {
        if (f.numerador == 0) {
            throw new IllegalArgumentException("No es posible dividir entre cero");
        }
        int newNumerador = this.numerador * f.denominador;
        int newDenominador = this.denominador * f.numerador;
        return new Fraccion(newNumerador, newDenominador);
    }

    // Métodos de clase
    public static Fraccion sumar(Fraccion a, Fraccion b) {
        return a.sumar(b);
    }

    public static Fraccion restar(Fraccion a, Fraccion b) {
        return a.restar(b);
    }

    public static Fraccion multiplicar(Fraccion a, Fraccion b) {
        return a.multiplicar(b);
    }

    public static Fraccion dividir(Fraccion a, Fraccion b) {
        return a.dividir(b);
    }
}
```

## Clase Main.java:

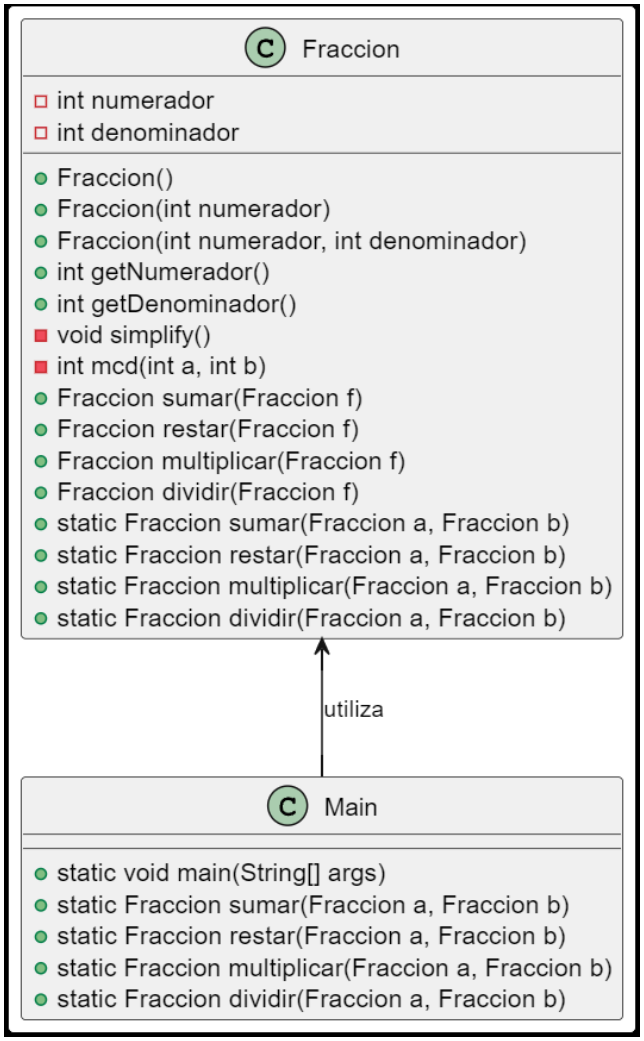
```java
public class Main {
    public static void main(String[] args) {
        Fraccion f1 = new Fraccion(7, 9);
        Fraccion f2 = new Fraccion(5, 13);

        // Usando los métodos de instancia
        Fraccion suma = f1.sumar(f2);
        Fraccion diferencia = f1.restar(f2);
        Fraccion producto = f1.multiplicar(f2);
        Fraccion cociente = f1.dividir(f2);

        // Usando los métodos de clase
        Fraccion sumaStatic = Fraccion.sumar(f1, f2);
        Fraccion diferenciaStatic = Fraccion.restar(f1, f2);
        Fraccion productoStatic = Fraccion.multiplicar(f1, f2);
        Fraccion cocienteStatic = Fraccion.dividir(f1, f2);

        // Usando los métodos de la clase Main
        Fraccion sumaMain = sumar(f1, f2);
        Fraccion diferenciaMain = restar(f1, f2);
        Fraccion productoMain = multiplicar(f1, f2);
        Fraccion cocienteMain = dividir(f1, f2);

        System.out.println("* Resultados de los métodos de instancia:");
        System.out.println("   - Suma: " + suma.getNumerador() + "/" + suma.getDenominador());
        System.out.println("   - Diferencia: " + diferencia.getNumerador() + "/" + diferencia.getDenominador());
        System.out.println("   - Producto: " + producto.getNumerador() + "/" + producto.getDenominador());
        System.out.println("   - Cociente: " + cociente.getNumerador() + "/" + cociente.getDenominador());

        System.out.println("\n* Resultados de los métodos de clase:");
        System.out.println("   - Suma: " + sumaStatic.getNumerador() + "/" + sumaStatic.getDenominador());
        System.out.println("   - Diferencia: " + diferenciaStatic.getNumerador() + "/" + diferenciaStatic.getDenominador());
        System.out.println("   - Producto: " + productoStatic.getNumerador() + "/" + productoStatic.getDenominador());
        System.out.println("   - Cociente: " + cocienteStatic.getNumerador() + "/" + cocienteStatic.getDenominador());

        System.out.println("\n* Resultados de los métodos de la clase princial (main):");
        System.out.println("   - Suma: " + sumaMain.getNumerador() + "/" + sumaMain.getDenominador());
        System.out.println("   - Diferencia: " + diferenciaMain.getNumerador() + "/" + diferenciaMain.getDenominador());
        System.out.println("   - Producto: " + productoMain.getNumerador() + "/" + productoMain.getDenominador());
        System.out.println("   - Cociente: " + cocienteMain.getNumerador() + "/" + cocienteMain.getDenominador());
    }
```

```java

    public static Fraccion sumar(Fraccion a, Fraccion b) {
        return a.sumar(b);
    }

    public static Fraccion restar(Fraccion a, Fraccion b) {
        return a.restar(b);
    }

    public static Fraccion multiplicar(Fraccion a, Fraccion b) {
        return a.multiplicar(b);
    }

    public static Fraccion dividir(Fraccion a, Fraccion b) {
        return a.dividir(b);
    }
}

```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional
ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jhona\AppData\Ro
* Resultados de los métodos de instancia:
   - Suma: 136/117
   - Diferencia: 46/117
   - Producto: 35/117
   - Cociente: 91/45

* Resultados de los métodos de clase:
   - Suma: 136/117
   - Diferencia: 46/117
   - Producto: 35/117
   - Cociente: 91/45

* Resultados de los métodos de la clase princial (main):
   - Suma: 136/117
   - Diferencia: 46/117
   - Producto: 35/117
   - Cociente: 91/45
```

## Diagrama UML:



## Ejercicio 3:

- Diagrama de clases UML y programa que utilice objetos de las clases creadas
- Crear la clase Imaginario con constructores sobrecargados, usar this para llamadas entre ellos. Crear sets, gets. También que permita mostrar en diferentes formatos: (a, b) a+bi. Y que permita retornar la suma, resta y multiplicación (el objeto llamado queda intacto). Además que permita retornar la módulo de un imaginario y su conjugado. Evitar duplicar código. Usar 3 formas de implementación para cada operación.

```
suma = i1.sumar(i2);

suma = Imaginario.sumar(i1, i2);

suma = sumar(i1, i2);
```

- Menú:

    1. Crear ArrayList de Imaginarios

    2. Mostrar todos los Imaginarios creados

    3. Realizar operaciones (de 2 primeros Imaginarios almacenados): sumar, restar y multiplicar. Módulo y conjugado (del primer Imaginario). En cada operación mostrar el resultado usando todas las formas posibles

    4. Salir

### Clase Imaginario.java:

```java
public class Imaginario {
    private double real;
    private double imaginario;

    // Constructores sobrecargados
    public Imaginario() {
        this(0, 0);
    }

    public Imaginario(double real) {
        this(real, 0);
    }

    public Imaginario(double real, double imaginario) {
        this.real = real;
        this.imaginario = imaginario;
    }

```

```java
    // Getters Y setters
    public double getReal() {
        return real;
    }

    public double getImaginario() {
        return imaginario;
    }

    public void setReal(double real) {
        this.real = real;
    }

    public void setImaginario(double imaginario) {
        this.imaginario = imaginario;
    }

    // Formatos para mostrar el numero imaginario
    public String formato1() {
        return "(" + real + ", " + imaginario + ")";
    }

    public String formato2() {
        return real + " + " + imaginario + "i";
    }

    // Operaciones con los primeros números del arreglo
    // Métodos de instancia
    public Imaginario sumar(Imaginario i) {
        return new Imaginario(this.real + i.real, this.imaginario + i.imaginario);
    }

    public Imaginario restar(Imaginario i) {
        return new Imaginario(this.real - i.real, this.imaginario - i.imaginario);
    }

    public Imaginario multiplicar(Imaginario i) {
        double parteReal = this.real * i.real - this.imaginario * i.imaginario;
        double parteImaginaria = this.real * i.imaginario + this.imaginario * i.real;
        return new Imaginario(parteReal, parteImaginaria);
    }

    // Métodos de clase
    public static Imaginario sumar(Imaginario i1, Imaginario i2) {
        return new Imaginario(i1.real + i2.real, i1.imaginario + i2.imaginario);
    }
```

```java

    public static Imaginario restar(Imaginario i1, Imaginario i2) {
        return new Imaginario(i1.real - i2.real, i1.imaginario - i2.imaginario);
    }

    public static Imaginario multiplicar(Imaginario i1, Imaginario i2) {
        double parteReal = i1.real * i2.real - i1.imaginario * i2.imaginario;
        double parteImaginaria = i1.real * i2.imaginario + i1.imaginario * i2.real;
        return new Imaginario(parteReal, parteImaginaria);
    }

    // Otros métodos
    public double modulo() {
        return Math.sqrt(real * real + imaginario * imaginario);
    }

    public Imaginario conjugado() {
        return new Imaginario(real, -imaginario);
    }
}
```

## Clase Main.java:

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Imaginario> imaginarios = new ArrayList<>();
        int opcion;

        do {
            System.out.println("Menú:");
            System.out.println("1. Crear ArrayList de Imaginarios");
            System.out.println("2. Mostrar todos los Imaginarios creados");
            System.out.println("3. Realizar operaciones (de 2 primeros Imaginarios almacenados)");
            System.out.println("4. Salir");
            System.out.print("Seleccione una opción: ");
            opcion = sc.nextInt();

            switch (opcion) {
                case 1:
                    System.out.print("Ingrese la parte real: ");
                    double real = sc.nextDouble();
                    System.out.print("Ingrese la parte imaginaria: ");
                    double imaginario = sc.nextDouble();
                    imaginarios.add(new Imaginario(real, imaginario));
                    break;
                case 2:
                    for (Imaginario i : imaginarios) {
                        System.out.println(i.formato1() + " o " + i.formato2());
                    }
                    break;
                case 3:
                    if (imaginarios.size() < 2) {
                        System.out.println("Se necesitan al menos 2 Imaginarios para realizar operaciones.");
                        break;
                    }
                    Imaginario i1 = imaginarios.get(0);
                    Imaginario i2 = imaginarios.get(1);
```

```
                    // Suma
                    Imaginario suma1 = i1.sumar(i2);
                    Imaginario suma2 = Imaginario.sumar(i1, i2);
                    Imaginario suma3 = sumar(i1, i2);
                    System.out.println("Suma: " + suma1.formato2() + ", "
                                       + suma2.formato2() + ", " + suma3.formato2());

                    // Resta
                    Imaginario resta1 = i1.restar(i2);
                    Imaginario resta2 = Imaginario.restar(i1, i2);
                    Imaginario resta3 = restar(i1, i2);
                    System.out.println("Resta: " + resta1.formato2() + ", "
                                       + resta2.formato2() + ", " + resta3.formato2());

                    // Multiplicación
                    Imaginario multiplicacion1 = i1.multiplicar(i2);
                    Imaginario multiplicacion2 = Imaginario.multiplicar(i1, i2);
                    Imaginario multiplicacion3 = multiplicar(i1, i2);
                    System.out.println("Multiplicación: " + multiplicacion1.formato2() + ", "
                                       + multiplicacion2.formato2() + ", " + multiplicacion3.formato2());

                    // Módulo
                    System.out.println("Módulo de " + i1.formato2() + ": " + i1.modulo());

                    // Conjugado
                    System.out.println("Conjugado de " + i1.formato2() + ": " + i1.conjugado().formato2());
                    break;
                case 4:
                    System.out.println("Saliendo...");
                    break;
                default:
                    System.out.println("Opción no válida.");
            }
        } while (opcion != 4);
    }

    public static Imaginario sumar(Imaginario i1, Imaginario i2) {
        return i1.sumar(i2);
    }

    public static Imaginario restar(Imaginario i1, Imaginario i2) {
        return i1.restar(i2);
    }

    public static Imaginario multiplicar(Imaginario i1, Imaginario i2) {
        return i1.multiplicar(i2);
    }
}
```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín\2nd Year\Segundo Semestre\Fundamentos
ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jhona\AppData\Roaming\Code\User\workspaceStorage\b60f3f0c84de27c16315a
Menú:
1. Crear ArrayList de Imaginarios
2. Mostrar todos los Imaginarios creados
3. Realizar operaciones (de 2 primeros Imaginarios almacenados)
4. Salir
Seleccione una opción: 1
Ingrese la parte real: 11
Ingrese la parte imaginaria: 4.6
Menú:
1. Crear ArrayList de Imaginarios
2. Mostrar todos los Imaginarios creados
3. Realizar operaciones (de 2 primeros Imaginarios almacenados)
4. Salir
Seleccione una opción: 1
Ingrese la parte real: 15.15
Ingrese la parte imaginaria: 24.1
```

```
Menú:
1. Crear ArrayList de Imaginarios
2. Mostrar todos los Imaginarios creados
3. Realizar operaciones (de 2 primeros Imaginarios almacenados)
4. Salir
Seleccione una opción: 2
(11.0, 4.6) o 11.0 + 4.6i
(15.15, 24.1) o 15.15 + 24.1i
Menú:
1. Crear ArrayList de Imaginarios
2. Mostrar todos los Imaginarios creados
3. Realizar operaciones (de 2 primeros Imaginarios almacenados)
4. Salir
Seleccione una opción: 3
Suma: 26.15 + 28.700000000000003i, 26.15 + 28.700000000000003i, 26.15 + 28.700000000000003i
Resta: -4.15 + -19.5i, -4.15 + -19.5i, -4.15 + -19.5i
Multiplicación: 55.790000000000006 + 334.79i, 55.790000000000006 + 334.79i, 55.790000000000006 + 334.79i
Módulo de 11.0 + 4.6i: 11.923086848631105
Conjugado de 11.0 + 4.6i: 11.0 + -4.6i
Menú:
1. Crear ArrayList de Imaginarios
2. Mostrar todos los Imaginarios creados
3. Realizar operaciones (de 2 primeros Imaginarios almacenados)
4. Salir
Seleccione una opción: 4
Saliendo...
```

## Diagrama UML: