



Laboratorio 2: SOLUCIÓN DEL PROBLEMA 2

CURSO: Programador Java

DOCENTE: Gustavo Coronel Castillo

ALUMNA:

JAQUE PAJUELO, JUSCELLY NATIVIDAD

INDICE

INDICE	2
1. Resumen	3
2. Desarrollo del tema:	3
3. Caso desarrollado:	4
4. Conclusiones	19
5. Recomendaciones	19

1. Resumen

Se tienen dos problemas donde se busca reforzar los conocimientos de encapsulación, entender las diferencias entre los conceptos de variables, clase y métodos.

2. Desarrollo del tema:

Para desarrollar este caso se utilizó los conceptos de encapsulamiento y el concepto de clases y métodos.

3. Caso desarrollado:

ENUNCIADO: PRACTICA 04

Objetivo

Aplicar la sobrecarga para disponer de diversas versiones de métodos y constructores que se puedan aplicar dependiendo de las necesidades que se tengan o se proyecten tener.

Enunciado

La empresa EduTec necesita de una librería que permita calcular el promedio de un conjunto de números. Se sabe que pueden ser 2, 3, 4 o 5 números. A usted se le ha encargado que desarrolle la librería que necesita Edutec y construya una aplicación de prueba.

Solución planteada:

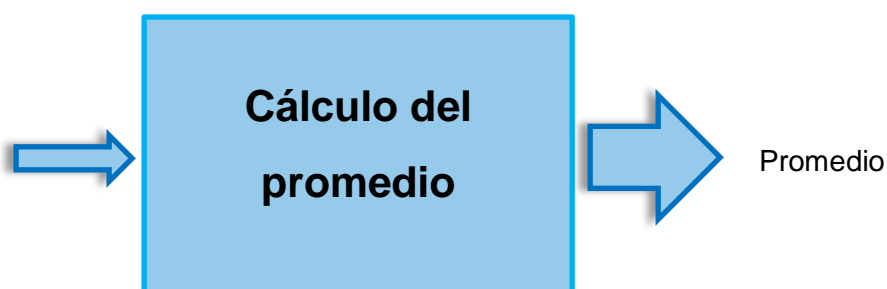
Servicio:

Número 1 Por el concepto de sobrecarga podemos crear
 Número 2 varias instancias de métodos, dependiendo de la
 cantidad de entradas que tengamos

Número 1
 Número 2
 Número 3

Número 1
 Número 2
 Número 3
 Número 4

Número 1
 Número 2
 Número 3
 Número 4
 Numero 5



Código Fuente:**Creación de clases:**

```

1  package edutec;
2
3  import edutec.View.EdutecVista;
4
5  public class EduTec {
6
7      /**
8       * @param args the command line arguments
9       */
10     public static void main(String[] args) {
11         // TODO code application logic here
12         EdutecVista.main(args);
13     }
14
15 }

```

Clase EditecDto para hacer la encapsulacion:

```

1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package edutec.Service;
7
8  /**
9   *
10   * @author Sistema
11   */
12  public class EdutecDto {
13      private double n1;
14      private double n2;
15      private double n3;
16      private double n4;
17      private double n5;
18      private double promedio;
19
20      public double getPromedio() {
21          return promedio;
22      }
23
24      public void setPromedio(double promedio) {
25          this.promedio = promedio;
26      }
27

```

Clase EdutecModel para la sobrecarga de métodos:

```

7      public class EdutecModel {
8          private double n1;
9          private double n2;
10         private double n3;
11         private double n4;
12         private double n5;
13         private double promedio;
14
15
16         public EdutecModel(double n1,double n2){
17             this.n1=n1;
18             this.n2=n2;
19         }
20
21         public EdutecModel(double n1,double n2,double n3){
22             this.n1=n1;
23             this.n2=n2;
24             this.n3=n3;
25         }
26
27         public EdutecModel(double n1,double n2,double n3,double n4){
28             this.n1=n1;
29             this.n2=n2;
30             this.n3=n3;
31             this.n4=n4;
32
33
34         public EdutecModel(double n1,double n2,double n3,double n4,double n5){
35             this.n1=n1;
36             this.n2=n2;
37             this.n3=n3;
38             this.n4=n4;
39             this.n5=n5;
40         }

```

Clase Prueba:

```

7      *
8      * @author Sistema
9      */
10     public class EdutecPrueba {
11     public static void main(String[] args) {
12         // Variable
13         double n1,n2,n3,n4,n5;
14         EdutecDto dto= new EdutecDto();
15
16         n1=10;
17         n2=20;
18         n3=30;
19         n4=40;
20         n5=50;
21
22         // Proceso
23         EdutecService service = new EdutecService();
24         dto=service.EdutecModel(n1,n2);
25         System.out.println("Promedio de los 2 numeros es: " + dto.getPromedio());
26         dto=service.EdutecModel(n1,n2,n3);
27         System.out.println("Promedio de los 3 numeros es: " +dto.getPromedio());
28         dto=service.EdutecModel(n1,n2,n3,n4);
29         System.out.println("Promedio de los 4 numeros es: " + dto.getPromedio());
30         dto=service.EdutecModel(n1,n2,n3,n4,n5);
31         System.out.println("Promedio de los 5 numeros es: " + dto.getPromedio());
32
33
34
35

```

Clase EdutecService:

```

11     public class EdutecService {
12
13     public EdutecDto EdutecModel(double n1,double n2) {
14         double promedio;
15         promedio=(n1+n2)/2;
16         EdutecDto dto=new EdutecDto();
17         dto.setPromedio(promedio);
18         return dto;
19     }
20
21     public EdutecDto EdutecModel(double n1,double n2,double n3){
22         double promedio;
23         promedio=(n1+n2+n3)/3;
24         EdutecDto dto=new EdutecDto();
25         dto.setPromedio(promedio);
26         return dto;
27     }
28
29     public EdutecDto EdutecModel(double n1,double n2,double n3,double n4){
30         double promedio;
31         promedio=(n1+n2+n3+n4)/4;
32         EdutecDto dto=new EdutecDto();
33         dto.setPromedio(promedio);
34         return dto;
35     }
36
37     public EdutecDto EdutecModel(double n1,double n2,double n3,double n4,double n5){
38         double promedio;
39         promedio=(n1+n2+n3+n4+n5)/5;
40         EdutecDto dto=new EdutecDto();
41         dto.setPromedio(promedio);
42         return dto;
43     }
44

```

Clase Formato para darle formato:

```
6 package edutec.View;
7
8 import java.text.DecimalFormat;
9
10 /**
11  *
12  * @author Sistema
13  */
14 public class Formato {
15     public static String format(double numero) {
16         String pattern = "###,##0.00";
17         DecimalFormat decimalFormat = new DecimalFormat(pattern);
18         return decimalFormat.format(numero);
19     }
20 }
21
```

Clase Vista:

The screenshot shows a Java Swing window titled "--EDUTEC--". The window has a light gray background and a vertical separator line. On the left side, there are five input fields labeled "Nro de Datos:", "Numero1:", "Numero2:", "Numero3:", "Numero4:", and "Numero5:". Below these is a "Resultado:" label and a dark green rectangular output area. On the right side, there are four buttons: "Validar" (green border), "Calcular" (blue border), "Nuevo" (yellow border), and "Salir" (red border).

Ejecutando nuestro programa:

The image shows a Java Swing window titled "--EDUTEC--". The window has a teal header bar with the title. Below the header, the form is organized into two columns. The left column contains labels for input fields: "Nro de Datos:", "Numero1:", "Numero2:", "Numero3:", "Numero4:", "Numero5:", and "Resultado:". The right column contains buttons: "Validar", "Calcular", "Nuevo", and "Salir". The "Validar" button is green, "Calcular" is blue, "Nuevo" is yellow, and "Salir" is red. The "Nro de Datos:" field contains the value "3". The "Numero1:" field contains "4", "Numero2:" contains "5", and "Numero3:" contains "6". The "Numero4:" and "Numero5:" fields are empty. The "Resultado:" field contains the value "5,00".

Label	Value	Button
Nro de Datos:	3	Validar
Numero1:	4	Calcular
Numero2:	5	
Numero3:	6	
Numero4:		Nuevo
Numero5:		
Resultado:	5,00	Salir

ENUNCIADO: PRACTICA 05**Objetivo**

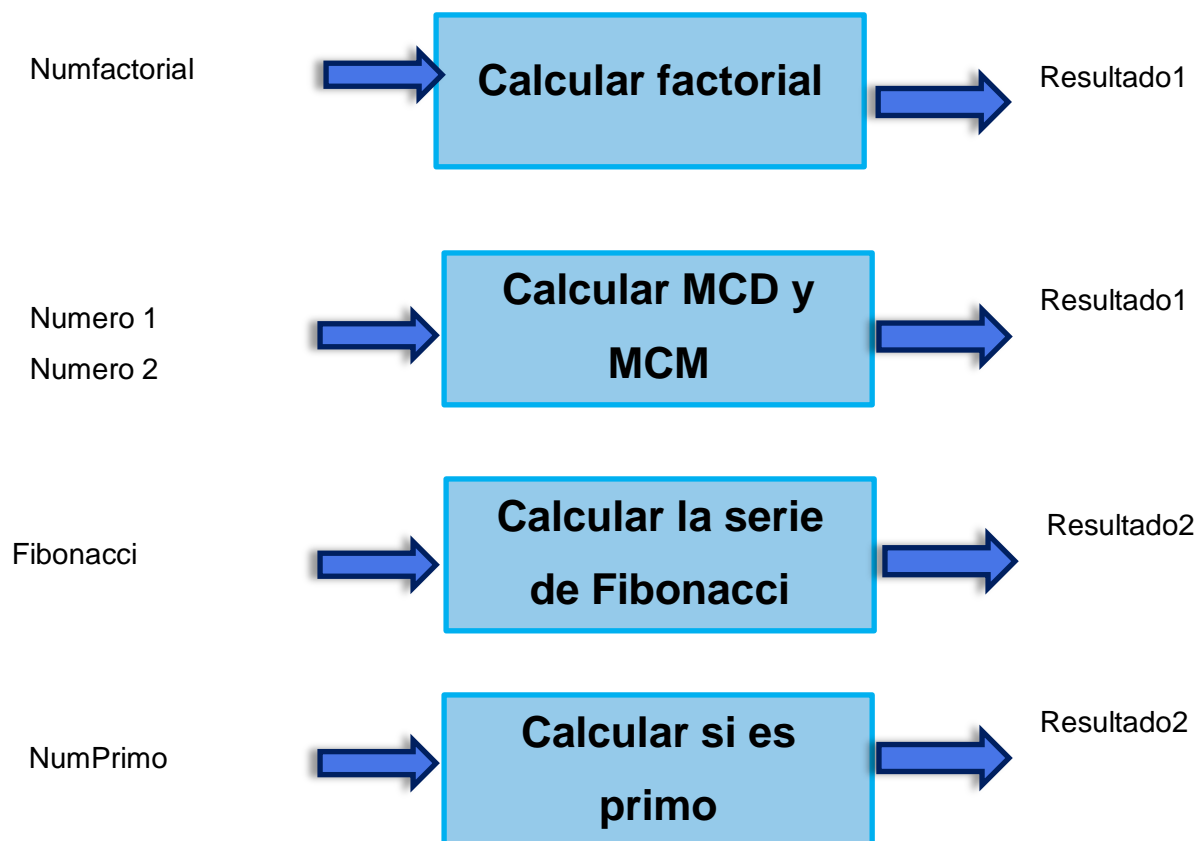
Entender la diferencia entre variables y métodos de instancia y de clase. Aplicar correctamente variables y métodos de instancia y de clase.

Enunciado

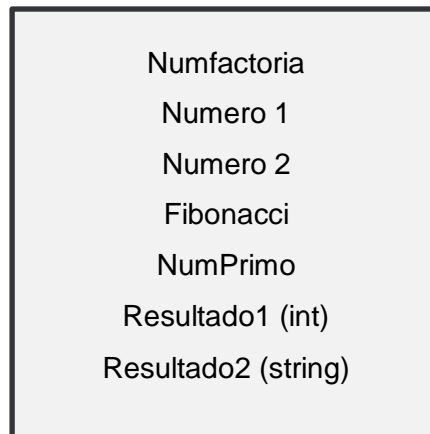
El colegio "Ángeles del Cielo" está solicitando un programa en Java para que los alumnos de primaria verifiquen sus ejercicios de matemáticas referidos a:

- Calculo de factorial
- Calculo del MCD y MCM de dos números
- La serie de Fibonacci
- Número primo

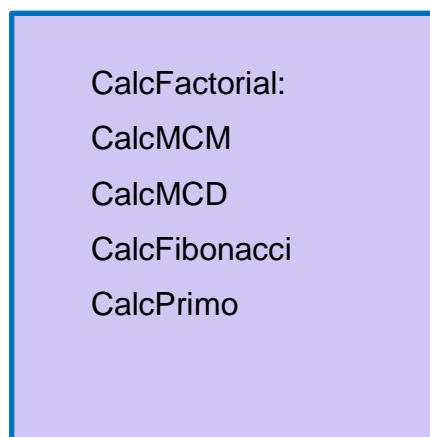
La programación de estos cálculos matemáticos debe estar implementados como métodos de clase en una clase de nombre MyMath.

Solución planteada:**Servicio:**

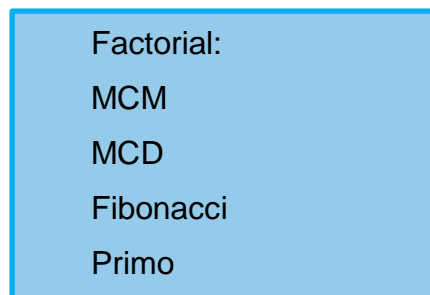
Encapsulación:



Definido 5 servicios:



Definir métodos:



Código Fuente:**Creación de clases:****Clase principal:**

```

6      package colegio;
7
8      import colegio.View.ColegioView;
9
10     /**
11      *
12      * @author Sistema
13      */
14     public class Colegio {
15
16         /**
17          * @param args the command line arguments
18          */
19         public static void main(String[] args) {
20             // TODO code application logic here
21             ColegioView.main(args);
22         }
23
24     }
25

```

Clase ColegioDto:

```

6      package colegio.dto;
7
8      /**
9      *
10     * @author Sistema
11     */
12     public class ColegioDto {
13         private int Numfactorial;
14         private int Num1;
15         private int Num2;
16         private int Fibonacci;
17         private int NumPrimo;
18
19         public int getNumfactorial() {
20             return Numfactorial;
21         }
22
23         public void setNumfactorial(int Numfactorial) {
24             this.Numfactorial = Numfactorial;
25         }
26
27         public int getFibonacci() {
28             return Fibonacci;
29         }
30

```

```

31  [ ] public void setFibonacci(int Fibonacci) {
32      [ ]     this.Fibonacci = Fibonacci;
33      [ ] }
34
35  [ ] public int getNumPrimo() {
36      [ ]     return NumPrimo;
37      [ ] }
38
39  [ ] public void setNumPrimo(int NumPrimo) {
40      [ ]     this.NumPrimo = NumPrimo;
41      [ ] }
42
43  [ ] public int getNum1() {
44      [ ]     return Num1;
45      [ ] }
46
47  [ ] public void setNum1(int Num1) {
48      [ ]     this.Num1 = Num1;
49      [ ] }
50
51  [ ] public int getNum2() {
52      [ ]     return Num2;
53      [ ] }
54
55  [ ] public void setNum2(int Num2) {
56      [ ]     this.Num2 = Num2;
57      [ ] }
58
59  private int Resultado1;
60  private String Resultado2;
61
62  [ ] public int getResultado1() {
63      [ ]     return Resultado1;
64      [ ] }
65
66  [ ] public void setResultado1(int Resultado1) {
67      [ ]     this.Resultado1 = Resultado1;
68      [ ] }
69
70  [ ] public String getResultado2() {
71      [ ]     return Resultado2;
72      [ ] }
73
74  [ ] public void setResultado2(String Resultado2) {
75      [ ]     this.Resultado2 = Resultado2;
76      [ ] }
77

```

Clase MyMath:

```

9  public class MyMath {
10     private int factorial, MCM, MCD, Fibonacci, Primo, Num1, Num2, n3, n4, max, min, maximo, minimo;
11     private int resultado1;
12     private String resultado2;
13
14     public MyMath() {
15         this.resultado1=0;
16         this.resultado2="";
17     }
18
19     public int Factorial(int Numfactorial) {
20         int acumulador=1;
21         if(Numfactorial>0) {
22             for(int i=1; i<=Numfactorial; i++) {
23                 acumulador=acumulador*i;
24                 resultado1=acumulador;
25             }
26         } else if (Numfactorial==0) {
27             resultado1=0;
28         } else {
29             JOptionPane.showMessageDialog(null, "Datos incorrectos");
30         }
31         return resultado1;
32
33     public int MCM(int Num1, int Num2) {
34         MCD=0;
35         max=Math.max(Num1, Num2);
36         min=Math.min(Num1, Num2);
37         maximo=max;
38         minimo=min;
39         do {
40             MCD=minimo;
41             minimo= maximo%minimo;
42             maximo=MCD;
43         } while (minimo!=0);
44         MCM= (max/MCD) *min;
45         resultado1=MCM;
46         return resultado1;
47     }
48
49     public int MCD (int Num1, int Num2) {
50         MCD=0;
51         max=Math.max(Num1, Num2);
52         min=Math.min(Num1, Num2);
53         maximo=max;
54         minimo=min;
55         do {
56             MCD=minimo;
57             minimo= maximo%minimo;
58             maximo=MCD;
59         } while (minimo!=0);
60         resultado1=MCD;
61         return resultado1;

```

```

64 public String Fibonacci(int Fibonacci){
65     int num1=0,num2=1,num3=0;
66     for (int i = 1; num3 < Fibonacci; i++) {
67         resultado2=resultado2+num3+" || ";
68         num3 = num1 + num2;
69         num1 = num2;
70         num2 = num3;
71     }
72     return resultado2;
73 }
74 public String Primo(int NumPrimo){
75     int cont,i;
76     cont=0;
77     resultado2="";
78     for(i=1;i<=NumPrimo;i++){
79         if( (NumPrimo%i)==0){
80             cont++;
81         }
82     }
83     if(cont<=2 & NumPrimo>1){
84         resultado2="Es primo";
85     }else{
86         resultado2="No es primo";
87     }
88     return resultado2;
89 }
90 }

```

Clase de prueba:

```

3 import colegio.Service.ColegioService;
4 import colegio.dto.ColegioDto;
5
6 /**
7  *
8  * @author Sistema
9  */
10 public class ColegioPrueba1 {
11     public static void main(String[] args) {
12
13         //Variables
14         ColegioDto dto= new ColegioDto();
15         //Datos
16         dto.setNumfactorial(10);
17         //Proceso
18         ColegioService service=new ColegioService();
19         dto=service.CalcFactorial(dto);
20
21         System.out.println(""+dto.getResultado1());
22     }
23 }
24
25

```

```

3  import colegio.Service.ColegioService;
4  import colegio.dto.ColegioDto;
5
6  /**
7   *
8   * @author Sistema
9   */
10 public class ColegioPrueba2 {
11     public static void main(String[] args) {
12         //Variables
13         ColegioDto dto= new ColegioDto();
14         //Datos
15         dto.setNum1(14);
16         dto.setNum2(7);
17         //Proceso
18         ColegioService service=new ColegioService();
19         dto=service.CalcMCM(dto);
20
21         System.out.println(""+dto.getResultado1());
22     }
23 }

```

```

3  import colegio.Service.ColegioService;
4  import colegio.dto.ColegioDto;
5
6  /**
7   *
8   * @author Sistema
9   */
10 public class ColegioPrueba3 {
11     public static void main(String[] args) {
12         //Variables
13         ColegioDto dto= new ColegioDto();
14         //Datos
15         dto.setFibonacci(14);
16         //Proceso
17         ColegioService service=new ColegioService();
18         dto=service.CalcFibonacci(dto);
19
20         System.out.println(""+dto.getResultado2());
21     }
22 }
23 }
24

```



```
3  import colegio.Service.ColegioService;
4  import colegio.dto.ColegioDto;
5
6  /**
7   *
8   * @author Sistema
9   */
10 public class ColegioPrueba4 {
11     public static void main(String[] args) {
12         //Variables
13         ColegioDto dto= new ColegioDto();
14         //Datos
15         dto.setNumPrimo(14);
16         //Proceso
17         ColegioService service=new ColegioService();
18         dto=service.CalcPrimo(dto);
19
20         System.out.println(""+dto.getResultado2());
21     }
22 }
23
24
```

Paquete View

Ejecución

The screenshot shows a Java application window titled "Colegio 'Angeles del Cielo'". It contains four distinct calculator sections, each with input fields, a "Calcular" button, and a "Limpiar" button.

- Factorial Calculator:** The input field "Calcular el factorial de:" contains the value 8. The "Resultado" field displays 40320.
- LCM and MCD Calculator:** The input field "Calcular el MCM y MCD d..." contains 25 and 15. The "MCM:" field displays 75, and the "MCD:" field displays 5.
- Fibonacci Series Calculator:** The input field "Hallar la serie de Fibonacci hasta:" contains 30. The "Resultado" field displays the sequence 0 || 1 || 2 || 3 || 5 || 8 || 13 || 21 ||.
- Primality Test Calculator:** The input field "Calcular si es primo:" contains 33. The "Resultado" field displays "No es primo".

4. Conclusiones

- El encapsulamiento nos permite almacenar las entradas y salidas de manera que se reduce el número de llamadas a los servicios.
- Para el primer problema se utiliza la sobrecarga de métodos, donde podemos tener varios métodos, con distintas cantidad de entradas, lo cual nos permite de una forma mejor personalizar nuestro código, y ya no hacer mas métodos, si no que reutilizamos nuestro código.
- La alternativa planteada para el segundo, es la ejecución de un programa que busca agilizar el proceso, usando las instancias de métodos, junto con la encapsulación, los métodos y las clases.

5. Recomendaciones

- Implementar la clase prueba nos permite probar nuestro código antes de implementarlo en nuestra clase view.
- Debemos de tener en cuenta cuales son las entradas y las salidas de nuestra clase servicio para poder cumplir con los requerimientos solicitados.