



# **Laboratorio 1: SOLUCIÓN DEL PROBLEMA 2**

**CURSO: Programador Java**

**DOCENTE: Gustavo Coronel Castillo**

**ALUMNAS:**

- **ALTAMIRANO PACHECO, ANGIE CELESTE**
- **ATÚNCAR QUISPE, MARISOL**
- **JAQUE PAJUELO, JUSCELLY NATIVIDAD**

# INDICE

INDICE	2
1. Resumen	3
2. Desarrollo del tema:	3
3. Caso desarrollado:	4
4. Conclusiones	7
5. Recomendaciones	7

## 1. Resumen

Se tiene un problema donde se busca calcular el importe de préstamos que ofrece una financiera, donde tenemos 3 entradas el capital, la tasa y el número de periodos. Para ello usaremos el concepto de encapsulación donde accedemos a los objetos o instancias de las clases a través de sus métodos.

Para ello realizamos la creación de clases ejecutables y una interface de fácil uso que brinda además de la función de calcular, la función de volver a realizar el cálculo y cerrar la ventana.

## 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 02

Una institución financiera necesita de un programa que le permita encontrar el importe que deben pagar sus clientes por los préstamos que realiza, se sabe que se trata de un interés compuesto, capitalizable mensualmente. La fórmula que debe aplicarse es:

$$M = C(1 + i)^t$$

Dónde:

C: Capital

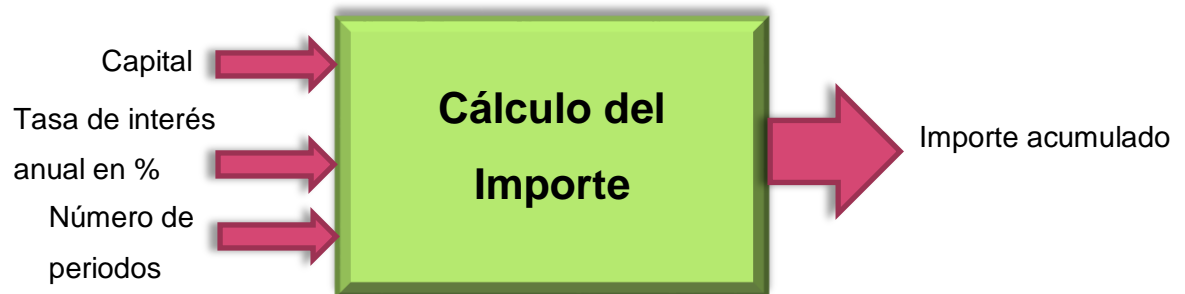
i: Tasa de interés por periodo, por ejemplo, puede ser mensual

n: Número de periodos

M: Importe acumulado en el número de periodos

**Solución planteada:**

**Servicio:**



**Código Fuente:****Creación de clases:**

```

1   package Prestamos;
2
3   import prestamo.View.PrestamoView;
4
5   /**
6    *
7    * @author Sistema
8    */
9   public class Prestamos {
10
11       /**
12        * @param args the command line arguments
13        */
14       public static void main(String[] args) {
15           // TODO code application logic here
16           PrestamoView.main(args);
17       }
18   }
19
20 }

```

```

1   package prestamo.service;
2
3   /**
4    *
5    * @author Sistema
6    */
7   public class PrestamoService {
8
9       public double CalcImporte(double capital,double tasa,double CantPeriod){
10           double importe;
11           tasa=tasa/100;
12           importe=capital*Math.pow(1+tasa, CantPeriod);
13           return importe;
14       }
15
16   }

```

```

1  package prueba;
2
3  import prestamo.service.PrestamoService;
4
5
6  public class Prueba01 {
7      public static void main(String[] args) {
8          //Variables
9          double importe, capital, tasa, CantPeriod;
10         //Datos
11         capital=1000;
12         tasa=10;
13         CantPeriod=2;
14         //Proceso
15         PrestamoService service=new PrestamoService();
16         importe=service.CalcImporte(capital, tasa, CantPeriod);
17         //Reporte
18         System.out.println("Importe: "+importe);
19     }
20 }
21
22

```

### Paquete View

PRESTAMOS	RESULTADO
Capital: <input type="text"/>	Importe a pagar: <input type="text"/>
Tasa(%): <input type="text"/>	
Numero de periodos: <input type="text"/>	
<b>ACCIONES</b> <div> <input type="button" value="Calcular"/> <input type="button" value="Nuevo"/> <input type="button" value="Salir"/> </div>	

### Ejecución

FINANCIERA - PRACTICA 02

PRESTAMOS	RESULTADO
Capital: <input type="text"/>	Importe a pagar: <input type="text"/>
Tasa(%): <input type="text"/>	
Numero de periodos: <input type="text"/>	
<b>ACCIONES</b> <div> <input type="button" value="Calcular"/> <input type="button" value="Nuevo"/> <input type="button" value="Salir"/> </div>	

#### 4. Conclusiones

- El encapsulamiento nos permite implementar al método Prestamo.Servicio en la clase Préstamo.view de manera más ordenada.
- La clase pueba.java nos permite probar el servicio creado en este caso CalcImporte antes de implementarlo en nuestra clase vista, para ver si hay un error, en ese caso poder hacer las respectivas modificaciones.
- La alternativa planteada, es la ejecución de un programa que busca agilizar el proceso, permitiéndole a la Financiera mostrar el importe del pago que deben efectuar sus clientes por los préstamos solicitados, utilizando el lenguaje de programación Java.

#### 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.

