

GUÍA DE LABORATORIO N° 02

Escuela Profesional: Ingeniería de Sistemas.**Ciclo y Turno:** IV – Mañana**Docente:** Ing. Ivan Petrlik / Ing. Gustavo Coronel**Asignatura:** Metodología de la Programación.**Semestre Académico:** 2017 - II**Fecha:** 04 al 08 de Setiembre del 2017

Orientación a Objetos

INTRODUCCION

Una de las grandes utilidades de la serialización de objetos con el manejo de arraylist y manejo de archivos binarios es su capacidad de guardar cualquier tipo de datos y su manipulación está sujeta al uso de una clase principal con el manejo de un arreglo de objetos para manipular la información a través de posiciones.

Se implementaran aplicaciones en entorno visual con JFrame, utilizando como herramienta de desarrollo NetBeans 6.8 y además para demostrar los el uso de archivos binarios y arreglos de objetos.

I. OBJETIVOS

1. Desarrollo de ejercicios de Manejo de Cadenas usando Swing WT.
2. Codificar, compilar y ejecutar los ejercicios de aplicación.
3. Conocer la estructura de un programa visual en Java.
4. Se espera que el estudiante asocie los conocimientos nuevos con la "nueva plantilla" que se propone.

II. EQUIPOS Y MATERIALES

- ③ Computadora personal.
- ③ Programa NetBeans IDE 6.7.1 o 6.8 correctamente instalado.
- ③ Notas de los ejercicios resueltos en la clase.

III. METODOLOGIA Y ACTIVIDADES

- Codificar los ejercicios desarrollados en el aula.
- Presentar avances y ejecución de cada uno de los ejercicios al docente o jefe de práctica encargado para la calificación correspondiente.
- Guardar la carpeta de sus archivos a sus memorias.
- Apagar el computador y dejarla en buen estado al retirarse del laboratorio dejar todo en orden.

IV. OBSERVACION

- 🔗 El estudiante deberá crear una carpeta de trabajo con el nombre "Laboratorio_12", a donde deberá direccional su proyecto a crear.
- 🔗 Se recomienda que el estudiante haya planteado los ejercicios de la Guía de ejercicios 12 para que pueda comprender las soluciones que se proponen.

V. NUEVAS LIBRERIAS

Ver diapositiva GUI en Java-AWT & SWING.pdf

VI. EJERCICIO

Proyecto 01 Encontrar el Interés Compuesto

Enunciado el Problema

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:

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

Donde:

- 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

Datos

Los datos que normalmente se tienen, son:

- ✓ El capital
- ✓ La tasa de interés anual
- ✓ El número de meses

Abstracción

La abstracción consiste en capturar la información necesaria para dar solución al problema, en la POO creará clases con atributos y operaciones que representen la solución de problema.

Se necesita:

- ✓ Un objeto que tenga una operación (método) que permita obtener el importe acumulado (M) aplicando la fórmula del interés compuesto.
- ✓ Una interfaz con la cual debe interactuar el usuario para realizar el cálculo.

Ejercicio 01

Capital

Interés Mensual

Número de Meses

Importe Final

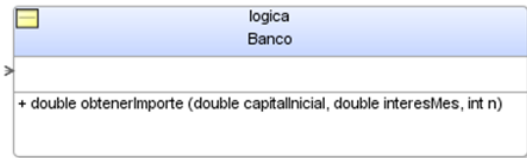


Diagrama de Secuencia

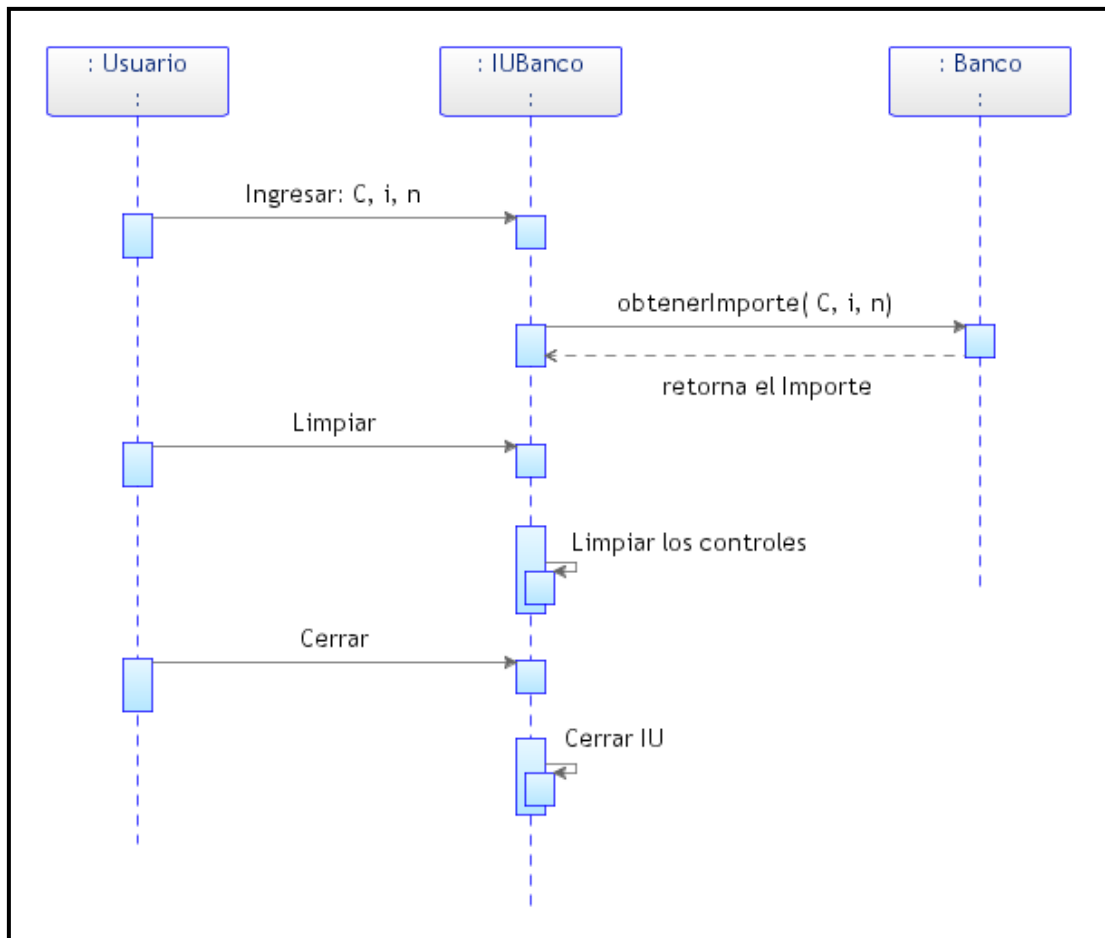
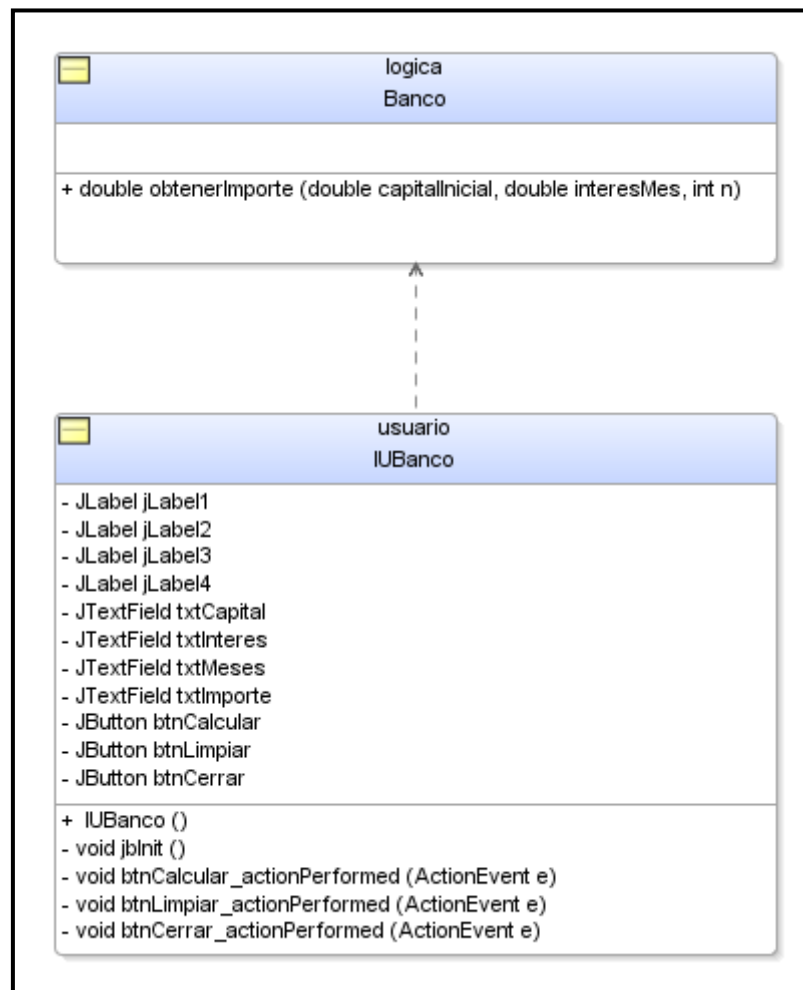


Diagrama de Clases



Implementación de la Clase Banco

Clase: logica.Banco

```
package logica;

public class Banco {

    public double obtenerImporte(double capitalInicial, double interesMeses, int n){

        double importe;
        importe = capitalInicial * Math.pow(1 + interesMeses, n);
        return importe;

    }

}
```

Botón: Calcular

```
// Datos
double capital = Double.parseDouble(this.txtCapital.getText());
double interes = Double.parseDouble(this.txtInteres.getText());
int meses = Integer.parseInt(this.txtMeses.getText());

// Instanciar la clase Banco
logica.Banco obj = new logica.Banco();

// Obtener el importe
double importe = obj.obtenerImporte(capital, interes, meses);

// Redondear el importe a dos decimales
importe = importe * 100;
importe = Math.round(importe);
importe = importe / 100;

// Presentar el resultado
this.txtImporte.setText(String.valueOf(importe));
```

Botón: Limpiar

```
this.txtCapital.setText("");
this.txtInteres.setText("");
this.txtMeses.setText("");
this.txtImporte.setText("");
```

Botón: Cerrar

```
System.exit(0);
```

Proyecto 02

Encontrar el Promedio de un Alumno

El administrador de cursos de una institución educativa necesita un programa para calcular el promedio de un alumno, se debe tomar en cuenta lo siguiente:

- ✓ Son cuatro notas de practica, de donde se obtiene un promedio de practicas (PP).
- ✓ Se tiene también un examen parcial (EP) y un examen final (EF).
- ✓ El promedio final (PF) se obtiene aplicando la siguiente formula:

$$PF = PP * 0.30 + EP * 0.30 + EF * 0.40$$

Se pide plantear la solución e implementarla aplicando los conceptos de POO.

Proyecto 03

Encontrar el Importe de una Venta

El dueño de una tienda necesita un programa para encontrar el importe de una venta, se debe tomar en cuenta lo siguiente:

- ✓ Cada venta es un solo tipo de artículo.
- ✓ El cliente puede comprar varias unidades del artículo.
- ✓ En función a la cantidad de unidades que el cliente está compra, el vendedor puede hacerle un descuento que resulta de una negociación en el acto con el cliente.

Se pide plantear la solución e implementarla aplicando los conceptos de POO.