



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formatos para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC y LSC	2003-1	2531	Programacion Orientada a Objetos I

PRÁCTICA No.	LABORATORIO DE	Ingeniero en Computacion	DURACIÓN (HORA)
6	NOMBRE DE LA PRÁCTICA	Herencia	2

### 1. INTRODUCCIÓN

La herencia, es uno de los puntos principales en la programación orientada a objetos, ya que es una herramienta que permite la reutilización de código. Se considera la aplicación de clases abstractas que permiten establecer ligadura dinámica que da la base para el polimorfismo.

### 2. OBJETIVO (COMPETENCIA)

Realizar programas en java utilizando correctamente la herencia con el fin de reutilizar código.

### 3. FUNDAMENTO

La herencia nos permite definir objetos nuevos en base a los que ya existen. En java se utiliza como una extensión de un clase. A la clase ya definida que proporciona la herencia se le llama superclase, a la clase derivada de esta se le conoce como subclase o clase extendida .Java soporta únicamente herencia sencilla .

#### Ventajas de utilizar herencia

Permite la reutilización del código

Fácil modificación de código

Adaptación de programas para trabajar en situaciones similares.

Extracción de elementos comunes de clases diferentes usando clases abstractas.

Organización de objetos en jerarquías

#### Definición de una clase extendida (subclase)

Formuló LSC. Lourdes Ramirez Fernandez e M.I.María Luisa González Ramírez	Revisó MC. Gloria Etelbina Chavez Valenzuela	Aprobó	Autorizó MC. MIGUEL ANGEL MARTINEZ ROMERO
Maestro	Coordinador de Programa Educativo	Gestión de Calidad	Director de la Facultad



## Formatos para prácticas de laboratorio

```
class Nombre_subclase extends Nombre_de_la_Superclase
{
//cuerpo de la clase extendida
}
```

Un clase extendida hereda todos los miembros de su superclase menos los constructores y agrega nuevos constructores que le pertenecen. La llamada del constructor de la superclase es invocando el método `super()`; donde `super` es una palabra clave de Java.

### La superclase

Cualquier definición de clase puede ser una superclase, si esta tiene elementos en común para otra clase, pero también las clases abstractas que por su naturaleza son superclases.

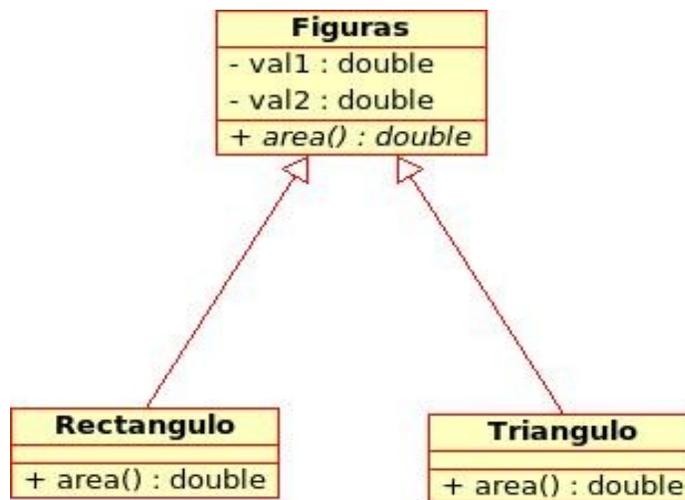
### Clases Abstractas

Es una clase que contiene uno o más métodos abstractos y por lo cual no es posible crear objetos de esa clase. Se utilizan para representar conceptos generales, las características comunes de un serie de objetos. Se definen para que otras clases puedan heredarlas y hacerlas concretas (que se pueden instanciar), implementando los métodos abstractos.

### Definición de una clase abstracta

```
abstract class Nombre_de_la_Superclase
{ //atributos
//métodos
//métodos abstractos
}
```

Ejemplo de una clase abstracta.





## Formatos para prácticas de laboratorio

//clase abstracta figuras, esta clase sera la super clase. Contiene el metodo area que es un método abstracto

```
abstract class Figuras {
private double val1;
private double val2;
//constructor de la clase figuras
public Figuras(double a,double b)
{ val1 =a;
val2 =b;}
//métodos publicos que permiten el acceso a los atributos privados
double getVal1()
{ return val1;}
double getVal2()
{ return val2;}
public abstract double area();
} //fin de la clase Figuras
```

//La clase Rectangulo es una subclase de la clase Figuras y debe implementar el codigo en el metodo area

```
class rectangulo extends Figuras
{
public rectangulo(double a,double b)
{ super(a,b); }
public double area()
{ double x=getVal1();
double y=getVal2();
return x *y;
}
}
```

//La clase Triangulo es una subclase de la clase Figuras y debe implementar el codigo en el metodo area

```
class triangulo extends Figuras
{
public triangulo(double a,double b)
{ super(a,b); }
public double area()
{ double x=getVal1();
double y=getVal2();
return x *y /2;
}
}
```

```
public class figura
{
public static void main(String args [])
{
rectangulo r =new rectangulo(9,5);
triangulo t =new triangulo(10,8);
```



## Formatos para prácticas de laboratorio

```
double ar,at;
ar=r.are ();
at=t.are ();
System.out.println("Área del rectángulo =" +ar);
System.out.println("Área del triángulo =" +at);
}
}
```

### 4. PROCEDIMIENTO (DESCRIPCIÓN)

**A) EQUIPO NECESARIO**  
computadora con SO Linux

**MATERIAL DE APOYO**  
Practica impresa

Pregunte a su maestro cual de los siguientes programas deberá realizar. En cada uno de ellos se aplicara herencia y el uso de clases y métodos abstractos.

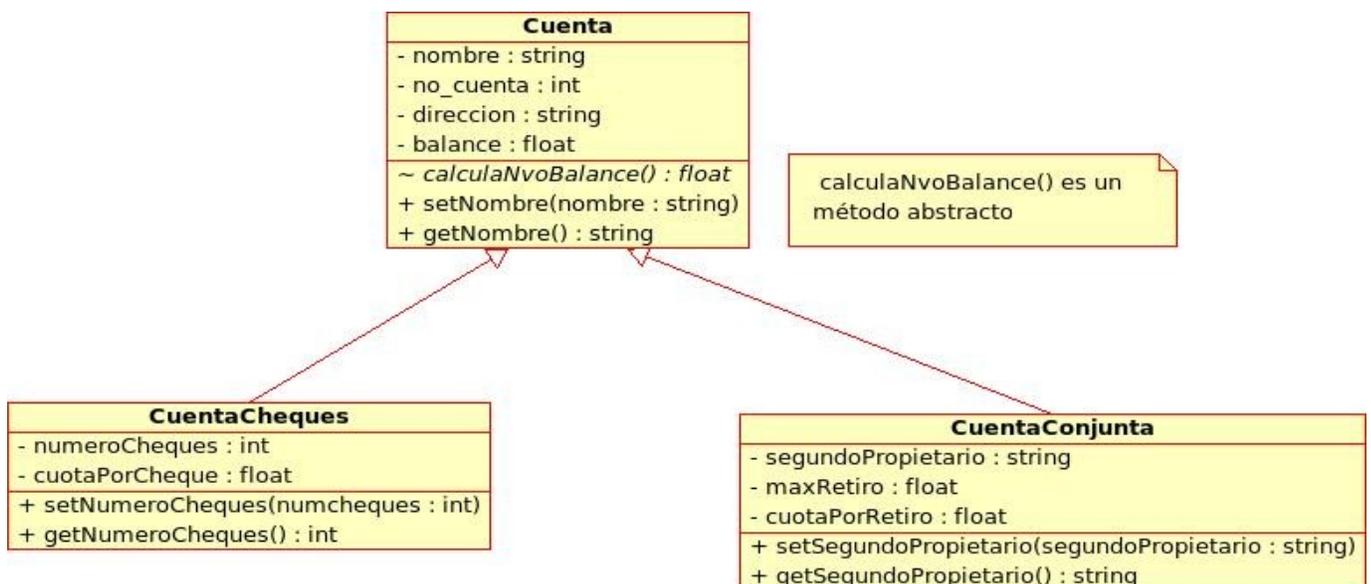
De acuerdo a los siguientes diagramas realice un programa que implemente las clases mostradas y utilice vectores de objetos para manipular los objetos.

En el diagrama solo se muestra un ejemplo de los métodos set y get, deberá implementar los métodos para todos los atributos de las clases.

En el método main comprobara el funcionamiento de las clases, con un menú de opciones que realicen lo siguiente:

- altas: dar de alta un objeto
- bajas: utilizando una llave dar de baja un objeto
- mostrar: mostrar los datos de todos los objetos dados de alta.

Problema 1.

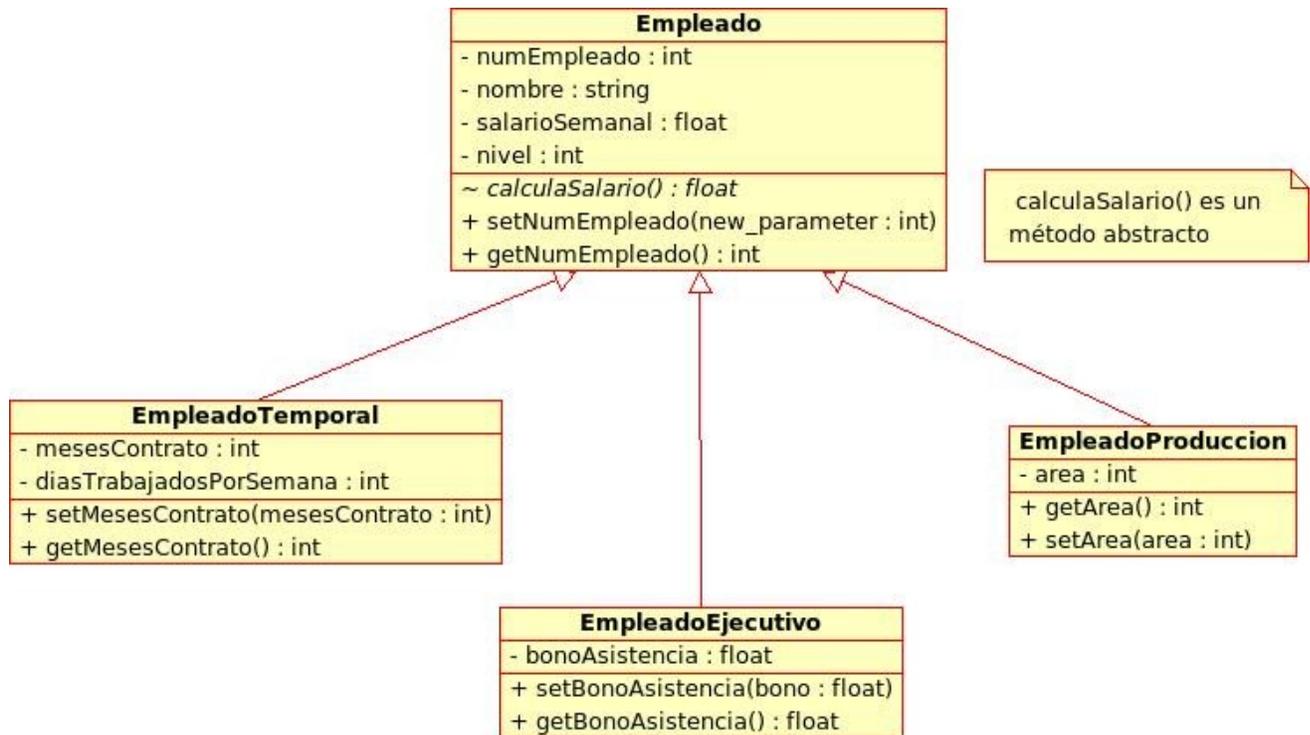




## Formatos para prácticas de laboratorio

El método `calculaBalance()` toma en consideración los atributos de la clase `CuentaCheques` para calcular el balance, y en la clase de `CuentaConjunta` utilizar la `coutaPorRetiro` para hacer el calculo del balance.

### Problema 2



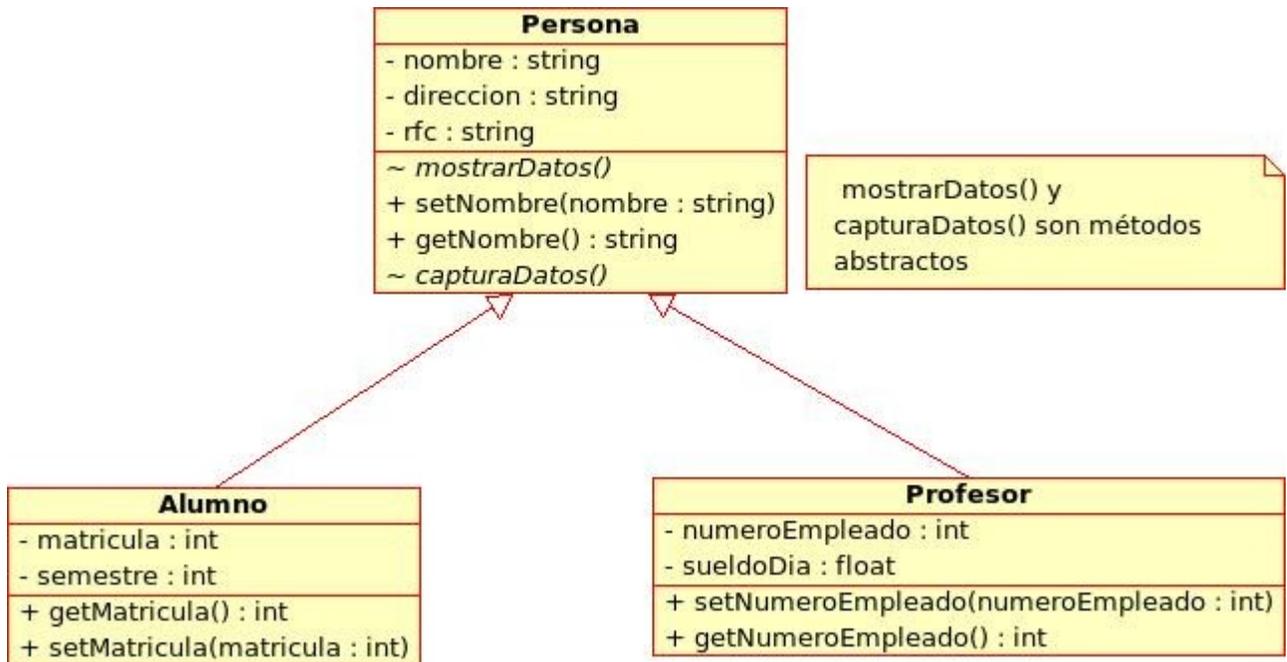
El método `calculaSalario` toma en consideración los atributos de `diasTrabajadosPorSemana` de la clase `EmpleadoTemporal` para realizar el calculo. En la clase `EmpleadoEjecutivo` considera el `bonoAsistencia` para el calculo y en `EmpleadoProduccion` según el `area` el salario por semana cambia, tome en consideración la siguiente tabla:

Área	Incremento de salario
1	10.00%
2	11.00%
3	13.00%
3	15.00%



## Formatos para prácticas de laboratorio

### Problema 3



Los métodos mostrarDatos() y capturaDatos() son métodos abstractos de la clase Persona, y varían de acuerdo a la subclase que se este implementando.

### C) CÁLCULOS Y REPORTE

### 5. RESULTADOS Y CONCLUSIONES

### 6. ANEXOS

### 7. REFERENCIAS