

# Diagramas UML y Java

## Implementar un diagrama de Clases

Trasladar el diagrama de clases a código requiere detalles específicos. Veamos varios ejemplos que nos ayuden a entenderlo.

### 1. Ejemplo 1

Invoice
+ amount : Real + date : Date = Current date + customer : String <u>- number of invoices : Integer = 0</u>

Se llevaría al siguiente código:

```
public class Invoice {  
    public double amount;  
    public Date date = new Date();  
    public String customer;  
    static private int number_of_invoices = 0;  
  
    // Constructor  
    public Invoice () {  
        // inicializaciones  
        number_of_invoices++;  
    }  
  
    // Métodos  
} ;
```

### 2. Ejemplo 2:

Figure
- x : Integer = 0 - y : Integer = 0
+ draw ()

En código:

```
public class Figure {  
    private int x = 0;  
    private int y = 0;  
  
    public void draw(){  
        // codigo para dibujar la figura  
    }  
};
```

Para usar esta clase en java:

```
Figure fig1 = new Figure();  
Figure fig2 = new Figure();  
fig1.draw();  
fig2.draw();
```

### 3. Ejemplo 3:

Si estamos en presencia de dos clases asociadas:

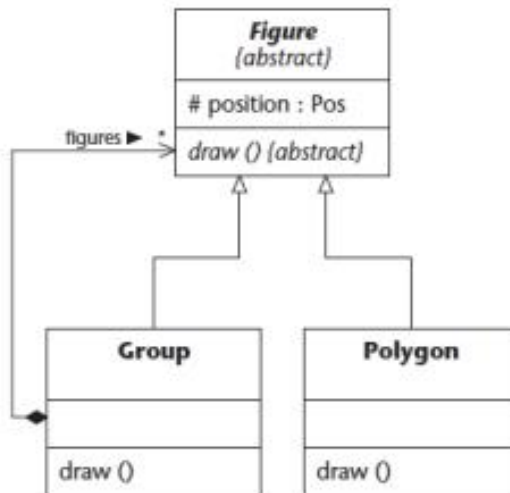


El código sería:

```
public class Insurance_company {  
  
    // Vector es una clase Java que  
    // representa la asociación con *  
  
    private Vector<Insurance_contract> contracts =  
        new Vector<Insurance_contract>(6);  
}  
  
public class Insurance_contract {  
    //asociación de 1 a 1  
  
    private Insurance_company refers_to;  
}
```

## 4. Ejemplo 4:

Ejemplo con clases abstractas:



Su implementación sería:

```
abstract public class Figure {
    protected Pos position;
    abstract public void draw();
}

public class Group extends Figure {
    // Vector es una clase de Java
    Int n = 5;
    private Vector<Figure> figures = new Vector<Figure>(n);

    public Group() {
        // inicializaciones
        for (int i = 1; i <= n; i++)
            figures.add(new Polygon());
    }

    public void draw() {
        for (int i = 0; i < figures.size(), i++){
            figures.get(i).draw();
        }
    }
}
```

```

public class Polygon extends Figure {
    public void draw () {
        /* Draw polygon code */
    }
}

```

Y para usarlo sería:

```

Group group = new Group();
group.draw();

```

## 5. Ejemplo 5

Ahora supongamos esta línea de código:

```

(usuario.getFechaNacimiento()).imprimir();

```

Si llevamos este código a un diagrama de clases vemos que deberían existir por lo menos dos Clases:

- El objeto guardado en el identificador *usuario* debe tener un tipo asociado a una clase, vamos a llamarla Usuario, que al menos tiene un método (getFechaNacimiento). Este método debe retornar una fecha.
- La fecha retornada debería ser un objeto de una clase, llamémosla Fecha. Esta clase debería tener al menos un método llamado imprimir.

Entonces, el diagrama de clase que se desprende de estas instrucciones sería:

