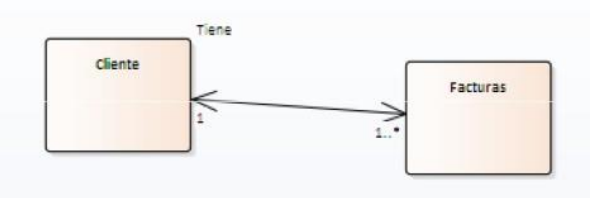
<u>Ejercicios generación de código</u>: Generar el código fuente java para las clases que se muestran en los siguientes diagramas de clases:

1. Asociación bidireccional:



Navigate Search Project Run Window Help

Factura.java

```java
public class Factura {

    public Cliente tiene;

}
```

Cliente.java

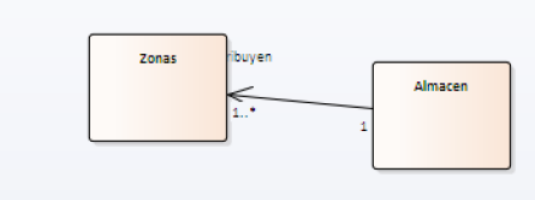
```java
import java.util.List;

public class Cliente {

    /**
     *
     *
     */
    public List<Factura> esTenida;

}
```

2. Asociación unidireccional:



Navigate Search Project Run Window Help

Zonas.java

```java
public class Zonas {


}
```

Almacen.java

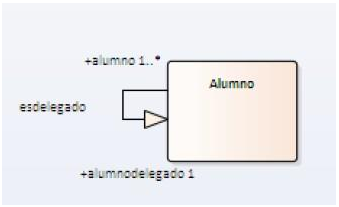
```java
import java.util.List;

public class Almacen {

    /**
     *
     *
     */
    public List<Zonas> zonas;

}
```

3. Asociación reflexiva:



Navigate Search Project Run Window Help

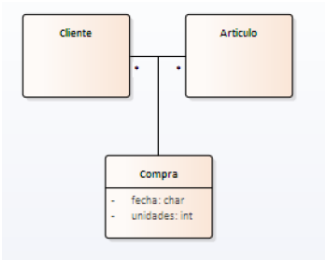
Alumno.java

```java
public class Alumno {

    public Alumno alumnoDelegado;
    /**
     *
     *
    */

}
```

4. Clase asociación:



Compra.java

```java
import java.util.List;

public class Compra {

  public date + fecha;

  public Integer + unidades;

    public List<Articulo> articulo;
```

Cliente.java

```java
import java.util.List;

public class Cliente {

    /**
     *
```

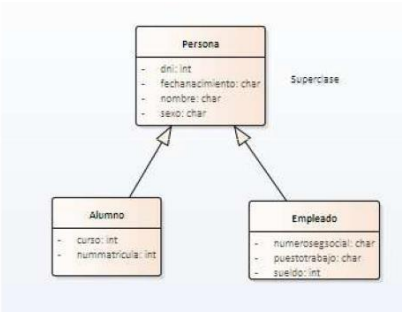
Articulo.java

```java
import java.util.List;

public class Articulo {

    /**
     *
```

5. Asociaciones – Herencia:



Empleado.java

```
public class Empleado extends Pe

  public Integer + numSS;

  public String + puestoTrabajo;

  public + sueldo: double;

}
```

Persona.java

```
public class Persona {

  public String + dni;

  public date + fechaNac;

  public String + nombre;

  public + sexo: char;

}
```

Alumno.java

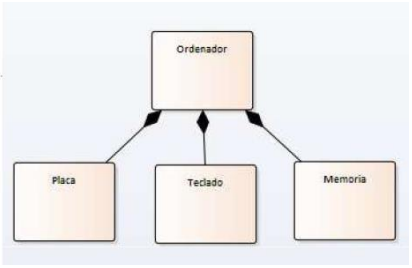
```
public class Alumno extends Persona

  public Integer + curso;

  public Integer + numMatricula;

}
```

6. Asociaciones – Composición:



Placa.java

```
import java.util.List;

public class Placa {

    public List<Ordenador> ordenador;

}
```

Teclado.java

```
import java.util.List;

public class Teclado {

    public List<Ordenador> ordenador;

}
```

Ordenador.java

```
import java.util.List;

public class Ordenador {

    public List<Placa> placa;
    public List<Teclado> teclado;
    public List<Memoria> memoria;

}
```

Memoria.java

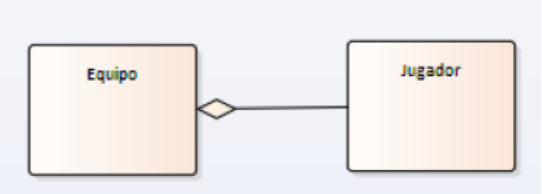
```
import java.util.List;

public class Memoria {

    public List<Ordenador> ordenador;

}
```

7. Asociaciones – Agregación:



Navigate Search Project Run Window Help

Jugador.java

```java
import java.util.List;

public class Jugador {

    public List<Equipo> equipo;

}
```

Equipo.java

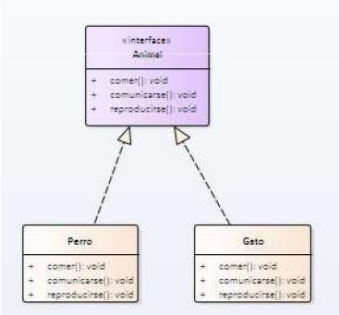
```java
import java.util.List;

public class Equipo {

    public List<Jugador> jugador;

}
```

8. Asociaciones – Realización:



Animal.java

```java
public interface Animal {

  public void + comer();

  public void + comunicarse();

  public void + reproducirse();

}
```

Perro.java

```java
public class Perro implements Animal {

  public void + comer() {
  }

  public void + comunicarse() {
  }

  public void + reproducirse() {
  }

}
```

Gato.java

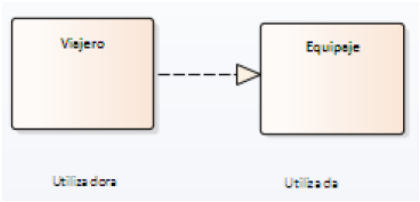
```java
public class Gato implements Animal {

  public void + comer() {
  }

  public void + comunicarse() {
  }

  public void + reproducirse() {
  }
```

9. Asociaciones – Dependencia:



Viajero.java

```
1 public class Viajero {
2 }
```

Equipaje.java

```
1 public class Equipaje {
2 }
```