



**Universidad Nacional Autónoma de
México**

**Dirección General de Cómputo de
Tecnologías de Información y
Comunicación**

-

**Diplomado
Desarrollo de Sistemas con Tecnología Java**

**Explicación
“Asociaciones”**

Mtro. ISC. Miguel Ángel Sánchez Hernández

Tabla de contenido

1. Asociaciones	3
2. Asociación entre clases	3
3. Rol	4
4. Acceso.....	4
5. Multiplicidad.....	5
6. Asociación Reflexiva.....	6
7. Composición.....	7

1. Asociaciones

En los lenguajes que son orientados a objetos no existe en si el concepto de asociación, por lo que su implementación debe ser con un mecanismo existente en el lenguaje a ocupar, en nuestro caso Java.

Para representar la asociación en Java la describiremos mediante las referencias entre otras clases, donde esta referencia se almacena como atributo de la clase.

2. Asociación entre clases

Supongamos que tenemos dos clases su asociación entre clases es la siguiente en UML.



Código Java:

```
class Clase_A{
    Clase_B ref;
}
class Clase_B{
    Clase_A ref;
}
```

3. Rol

Para indicar el concepto de la asociación ocuparemos el rol que es el nombre que le indicamos a la asociación.



Código Java:

```
class Clase_A{
    Clase_B rol2;
}
class Clase_B{
    Clase_A rol1;
}
```

4. Acceso

El concepto de acceso es indicar en que dirección queremos la asociación.



Código Java:

```
class Clase_A{
    Clase_B rol2;
}
class Clase_B{
}
```



Código Java:

```
class Clase_A{
}
class Clase_B{
    Clase_A rol1;
}
```

5. Multiplicidad

En todos los ejemplos que tenemos arriba la multiplicidad es de uno a uno, pero ahora veamos necesitamos uno a muchos y mucho a muchos, con esta multiplicidad indicamos el número de referencias que necesitamos asociar.



Código Java:

```
class Clase_A{
    Clase_B rol2;
}
class Clase_B{
    Clase_A rol1[];
}
```

Ahora si tuviéramos de muchos a muchos.

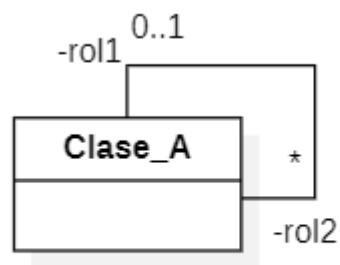


Código Java:

```
class Clase_A{
    Clase_B rol1[];
}
class Clase_B{
    Clase_A rol1[];
}
```

6. Asociación Reflexiva

Esta se toma como una asociación de una clase consigo misma.



Código Java:

```
class Clase_A{
    Clase_A rol1;
    Clase_A rol[];
}
```

7. Composición

Es una extensión del concepto de composición y en Java como no tiene un mecanismo para ocuparla, solo extendemos una referencia.



Código Java:

```
class Clase_A{
    Clase_B ref;
}
class Clase_B{
    Clase_A ref;
}
```