

Java

Taller de Programación en Java

UNICEN

Java

Ejemplo: Un Congreso

Congreso

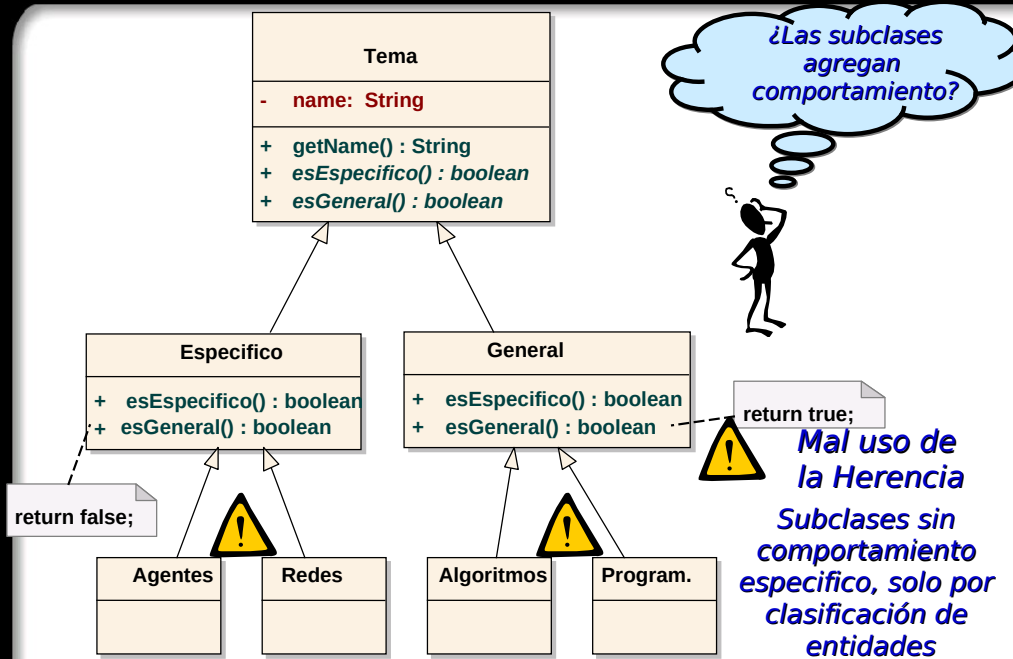
- El Comité de un Congreso de Ciencias de la Computación requiere de un sistema que le brinde los siguientes servicios:
 - ¿ que trabajos pueden ser asignados a un cierto evaluador ?
 - ¿ que evaluadores pueden evaluar un determinado trabajo?
 - ¿ un evaluador es común o experto ?
- Detalles:
 - Para determinar si un evaluador es apto para evaluar un trabajo, este evaluador debe haber declarado conocimiento en todos los temas del trabajo, especificados en sus palabras claves. Aunque en el caso de los posters basta que el evaluador tenga conocimiento en uno de los temas del poster
 - Evaluadores calificados por el comité organizador como comunes cuando sólo conocen algunos temas generales como algoritmos, lenguajes de programación, etc, o como evaluadores expertos cuando conocen temas específicos como agentes, visualización, redes de comunicación, etc.

Congreso

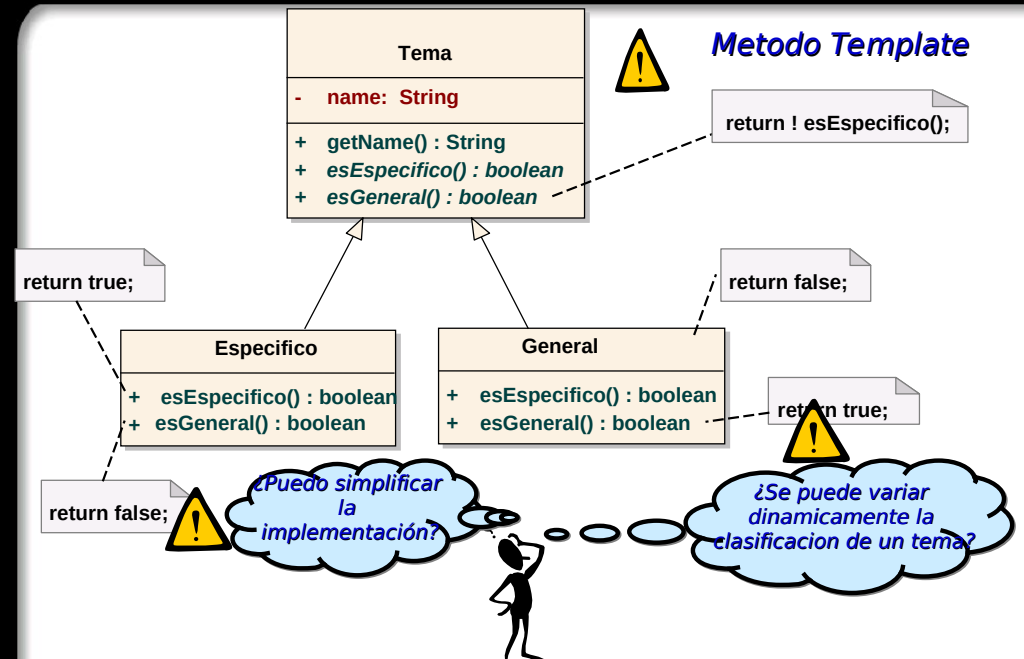
Analizando Alternativas



Temas



Temas



Temas

```

public class Tema {
    private String name;
    private boolean esEspecifico;

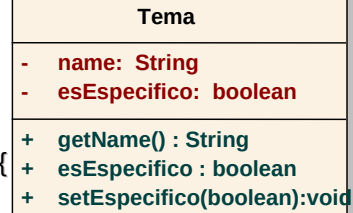
    public Tema (String name,boolean esEspecifico) {
        this.name=name;
        this.esEspecifico = esEspecifico;
    }

    public String getName() {
        return name; }

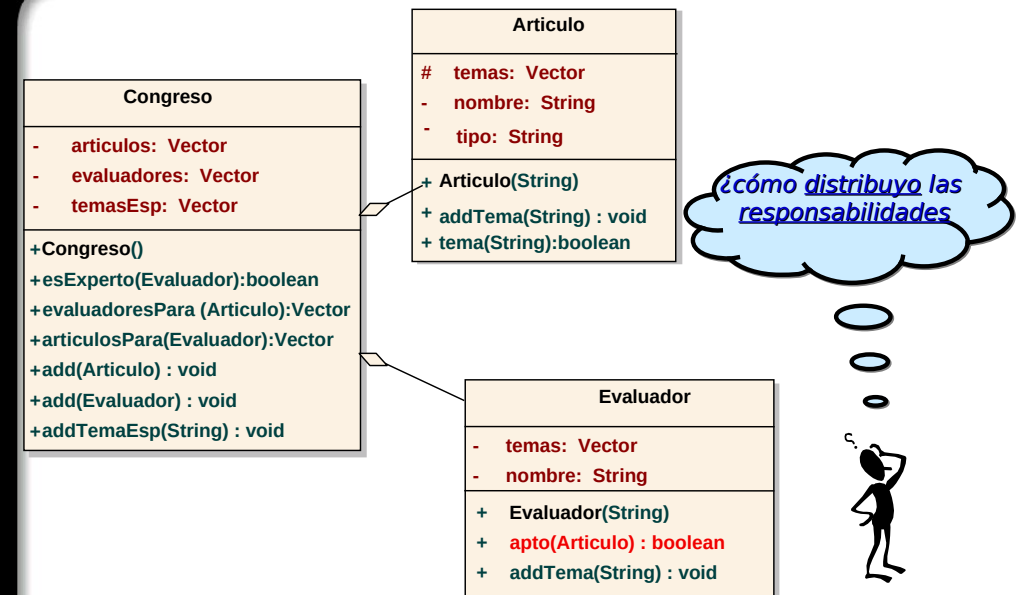
    public boolean esEspecifico() {
        return esEspecifico; }

    public void setEspecifico(boolean p) {
        esEspecifico =p; }
}
  
```

Incluso esta clase puede no ser necesaria



Congreso



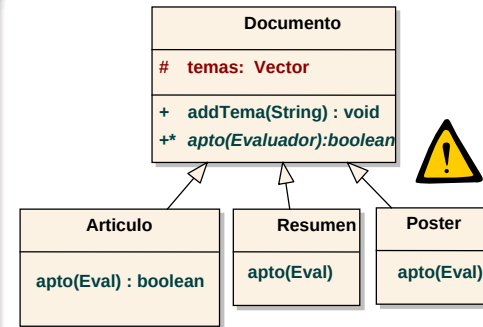
Congreso

```
public Vector articulosPara(Evaluador evaluador) { Congreso  
    Vector r= new Vector();  
    for (Enumeration e=articulos.elements(); e.hasMoreElements();) {  
        Articulo art=(Articulo )e.nextElement();  
        if (evaluador.apto(art)) r.add(art);}  
    return r;} 
```

```
public boolean apto(Articulo art) { Evaluador  
    if (art.tipo() == "articulo" || art.tipo() == "resumen" ) {  
        for (Enumeration e=art.temas().elements(); e.hasMoreElements();) {  
            if (! temas.includes (e.nextElement())) return false;    }  
        return true;    }  
    else if (art.tipo == "poster") {  
        for (Enumeration e=art.temas().elements(); e.hasMoreElements();) {  
            if (temas.includes (e.nextElement())) return true;    }  
        return false;    }  
    } else return false; }  
} Mal distribucion de responsabilidades.... ..se deberia hacer uso del polimorfismo
```

Taller de Programación en Java

Artículos



Mal uso de la Herencia:
Articulo y Resumen tienen el
mismo comportamiento



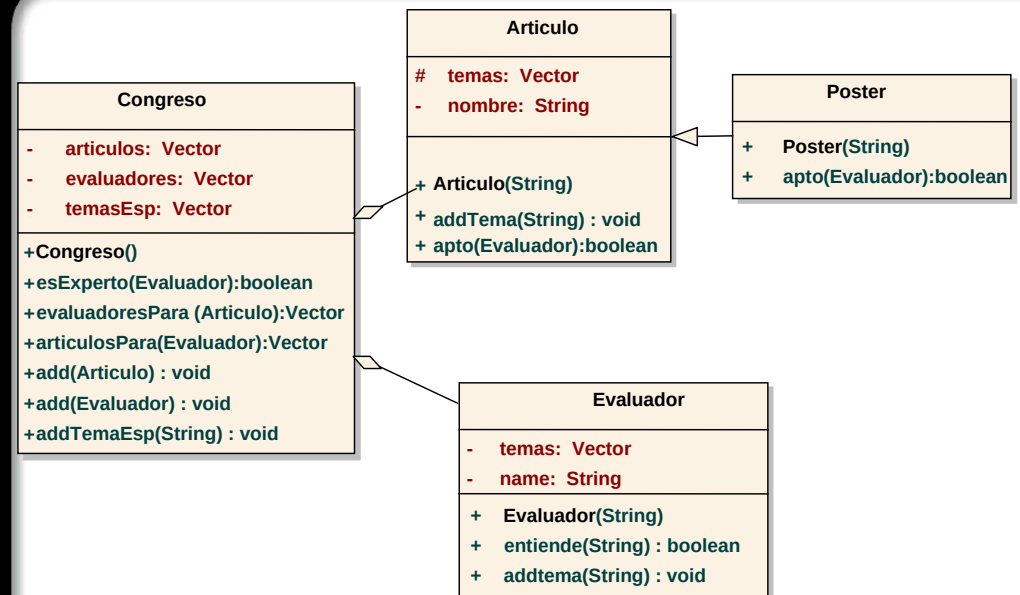
Taller de Programación en Java

10

Congreso

Una Solución

Congreso



Taller de Programación en Java

12

Congreso

```
public class Congreso {  
  
    private Vector articulos;           private Vector evaluadores;  
    private Vector temasEsp;  
  
    public Congreso () {  
        articulos =new Vector();   evaluadores =new Vector();  
        temasEsp =new Vector();   }  
  
    public void add(Evaluador eval) {  
        evaluadores.add(eval); }  
  
    public void add(Articulo a) {  
        articulos.add(a); }  
  
    public void add(String t) {  
        temasEsp.add(t); }  
}
```

Congreso

```
public boolean esExperto(Evaluador evaluador) {  
    for (Enumeration e=temasEsp.elements(); e.hasMoreElements();) {  
        if (evaluador.entiede(e.nextElement())) return true;}  
    return false;}  
  
    public Vector articulosPara(Evaluador evaluador) {  
        Vector r= new Vector();  
        for (Enumeration e=articulos.elements(); e.hasMoreElements();) {  
            Articulo art=(Articulo )e.nextElement();  
            if (art.apto(evaluador)) r.add(art);}  
        return r;}  
  
    public Vector evaluadoresPara(Articulo art) {  
        Vector r= new Vector();  
        for (Enumeration e=lotes.elements(); e.hasMoreElements();) {  
            Evaluador eval=(Evaluador )e.nextElement();  
            if (art.apto(eval)) r.add(eval); }  
        return r;}  
}
```

Evaluador

```
public class Evaluador {  
  
    private Vector temas;  
    private String name;  
  
    public Evaluador (String name) {  
        this.name=name;   temas =new Vector(); }  
  
    public boolean entiede(String t) {  
        return temas.contains(t); }  
  
    public void addTema(String t) {  
        temas.add(m); }  
}
```

Articulo

```
public class Articulo {  
    protected Vector temas;  
    private String titulo;  
  
    public void add(Tema t) {  
        temas.add(t); }  
  
    public boolean apto(Evaluador eval) {  
        for (Enumeration e=temas.elements(); e.hasMoreElements();) {  
            if (! eval.entiede(e.nextElement())) return false; }  
        return true; } }  
}
```

```
public class Poster extends Articulo {  
    public boolean apto(Evaluador eval) {  
        for (Enumeration e=temas.elements(); e.hasMoreElements();) {  
            if (eval.entiede(e.nextElement())) return true; }  
        return false; }  
}
```