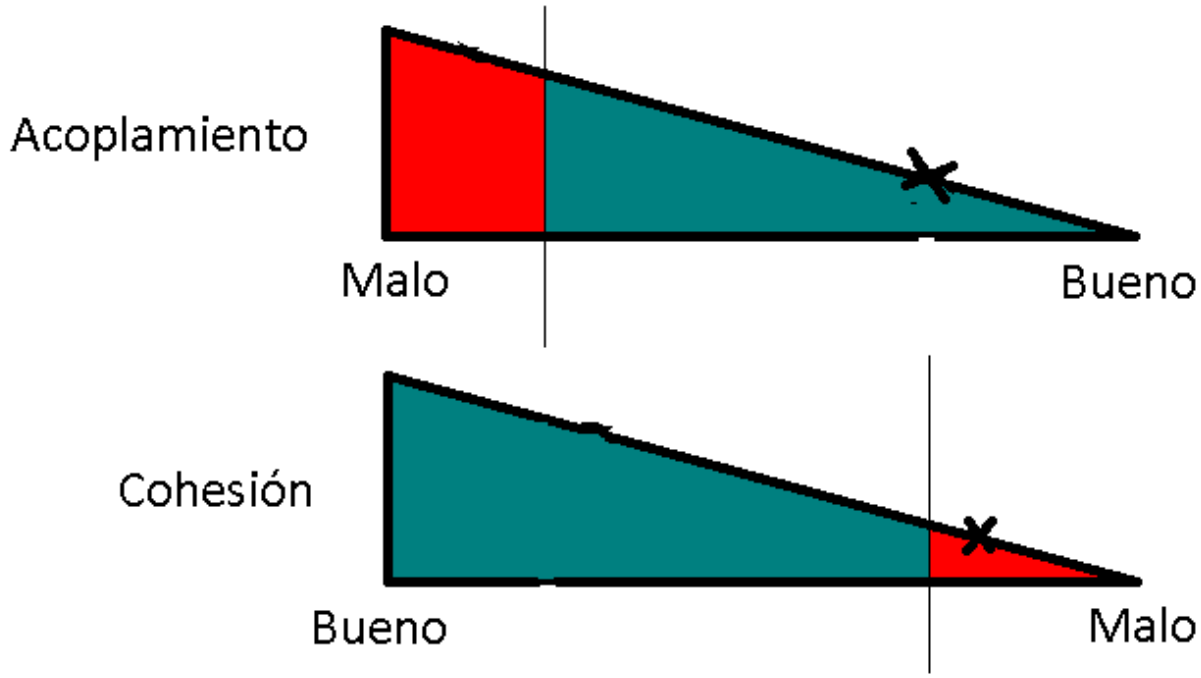
Enunciado
Modelar el juego Piedra Papel Tijera Lagarto Spock, con sus reglas

Irep elegido:
- No hay reglas repetidas.
- Las reglas no tienen orden, pero <e1,e2> se lee
  "e1 le gana a e2"
- No hay tuplas de reglas "simétricas" Si existe e1 R e2, entonces no existe e2 R e1.
- No hay tuplas que se indiquen que un elem se gane a si mismo (!reflexivas). e1 !R e1
- Los jugadores pueden jugar la misma jugada y en ese caso se declara empate. //TODO:
- La lista de reglas, no es exhasutiva (no estan todos con todos).
- Si la regla no esta definida, se lo considera un empate.

Version 1

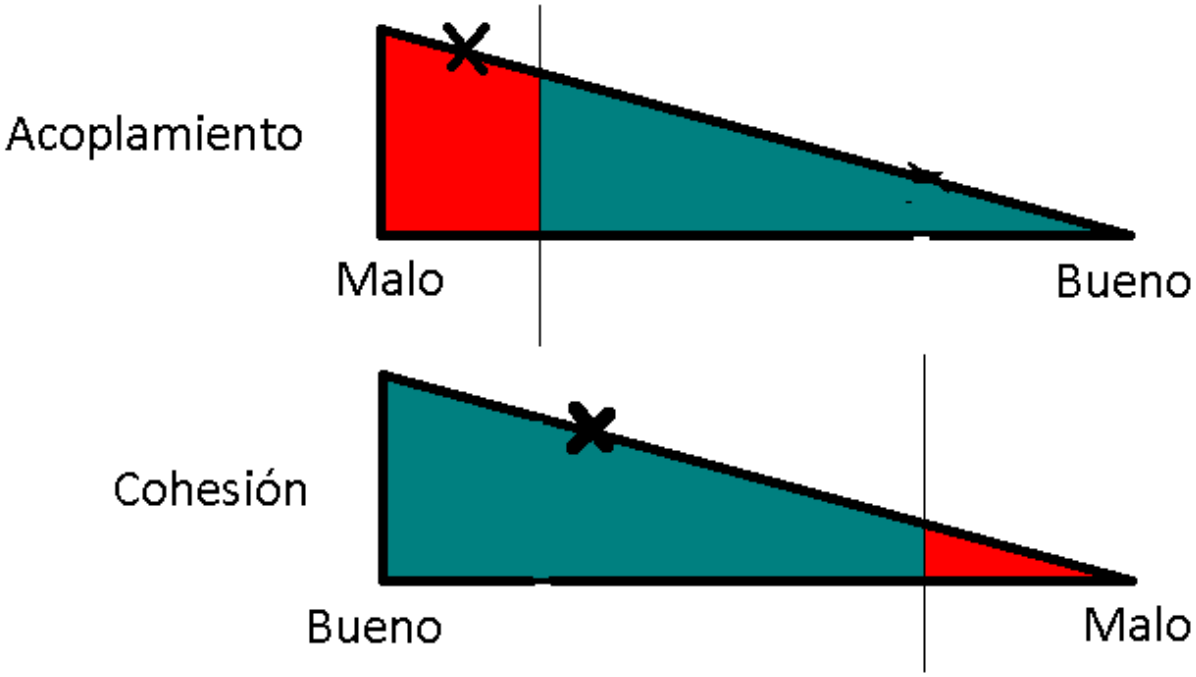
```
TAD PPTLS
ArrayList<String> elems;
Tupla<String,String> j1; <"Juan",Tijera>
Tupla<String,String> j2; <"Pedro", Papel>

//reglas
Set<Tupla<String,String>> datos;


String jugar()...
```



Se puede observar que tenemos baja cohesión, porque todo el TAD está todo en una única clase.
La clase PPTLS tiene demasiada (toda) la responsabilidad.
El bajo acoplamiento se “garantiza” artificialmente, por la baja cohesión PPTLS conoce todas las estructuras de datos.

Version 2

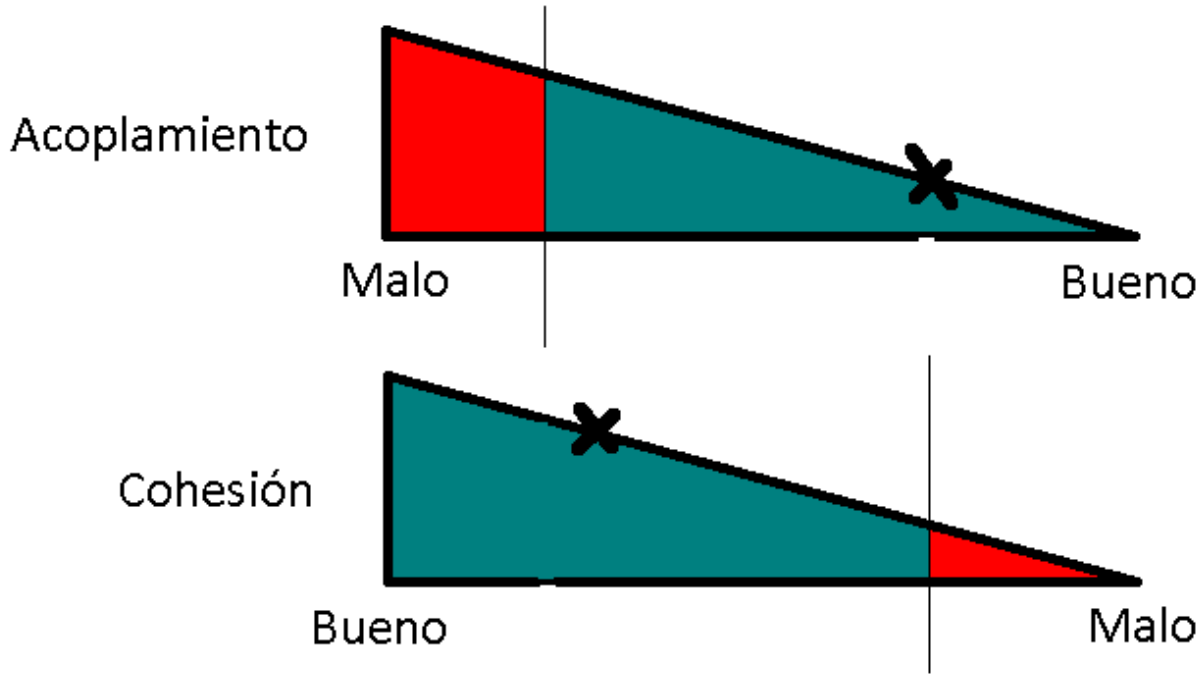
```
TAD PPTLS
ArrayList<String> elems;
Tupla<String,String> j1; <"Juan",Tijera>
Tupla<String,String> j2; <"Pedro", Papel>
Regla reglas;

// aumente la cohesion pero hay
// acoplamiento en la estructura de reglas
private String Jugar1( )
{
     Tupla tGanoJ1 = new Tupla(j1.getY(),j2.getY());
     Tupla tGanoJ2 = new Tupla(j2.getY(),j1.getY());

     if(this.reglas.pertenece(tGanoJ1))
          return j1.getX();

     if(this.reglas.pertenece(tGanoJ2))
          return j2.getX();

     return "Empate";
}

 ------------------------------------------------------------
TAD Regla
Set<Tupla<String,String>> datos;
{<Piedra,Tijera>,<Tijera,Papel>,<Lagarto,Spock>,.....}

private boolean pertenece(Tupla<String,String> jugada)
     return this.datos.contains(jugada);

boolean gano (String e1, String e2)
     Tupla aux = new Tupla(e1,e2)
     return pertenece(aux)
```



Si bien aumenta la cohesión al separar el TAD en dos, generamos acoplamiento porque PPTLS utiliza Reglas conociendo la implementación:
-Como “se” que Reglas internamente es un conjunto de Tupla, voy a utilizarlo de esa manera. Sin embargo la funcionalidad era saber si e1 le gana a e2, sin depender de cómo una Regla este implementado.

Version 3

```
PPTLS
private String Jugar2( )
{
     if(this.reglas.gano(j1.getY(),j2.getY())
          return j1.getX();

     if(this.reglas.gano(j2.getY(),j1.getY())
          return j2.getX();

     return "Empate";
}
```



Finalmente en esta última versión logramos una alta cohesión y un bajo acoplamiento