Cheung Derek *11/Octubre/2016*
González Rico Diana Virginia
Neri González José Francisco

Proyecto de Inteligencia Artificial “Representación del Conocimiento”

1. Crear predicados para consultar:

a) La extensión de una clase (el conjunto de todos los objetos que pertenecen a la misma, ya sea porque se declaren directamente o porque están en la cerradura de la relación de herencia).

b) La extensión de una propiedad (mostrar todos los objetos que tienen una propiedad específica ya sea por declaración directa o por herencia, incluyendo su respectivo valor).

c) La extensión de una relación (mostrar todos los objetos que tienen una relación específica ya sea por declaración directa o por herencia, incluyendo con quién están relacionados).

d) Todas las clases a las que pertenece un objeto.

e) Todas las propiedades de un objeto o clase.

f) Todas las relaciones de un objeto o clase.

2. Crear predicados para añadir:

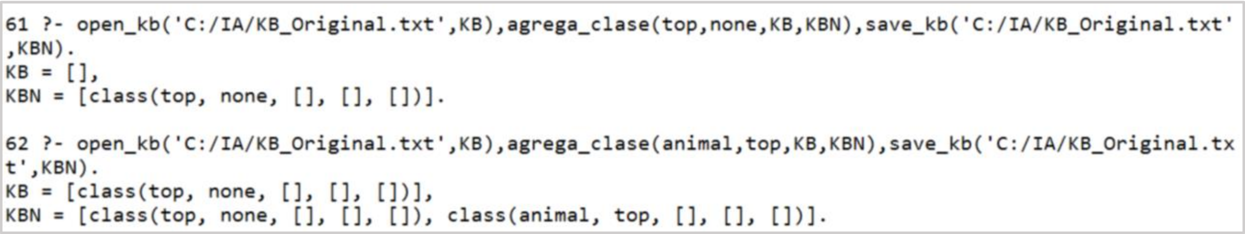
a) Clases u objetos.
Nombre: *agregra_clase(...)*

Descripción:
Agrega una nueva clase vacía a la base de conocimiento.

Argumentos:
NomClase: Nombre de la nueva clase.
Madre: Clase superior o madre de la nueva clase creada.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:
```
agrega_clase(NomClase,Madre,KB_Original,KB_Nuevo) :-
append(KB_Original,[class(NomClase,Madre,[],[],[])],KB_Nuevo).
```

Ejemplos de uso:



```
61 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_clase(top,none,KB,KBN),save_kb('C:/IA/KB_Original.txt'
,KBN).
KB = [],
KBN = [class(top, none, [], [], [])].

62 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_clase(animal,top,KB,KBN),save_kb('C:/IA/KB_Original.tx
t',KBN).
KB = [class(top, none, [], [], [])],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], [])].
```

Nombre: *agregra_objeto_clase(...)*

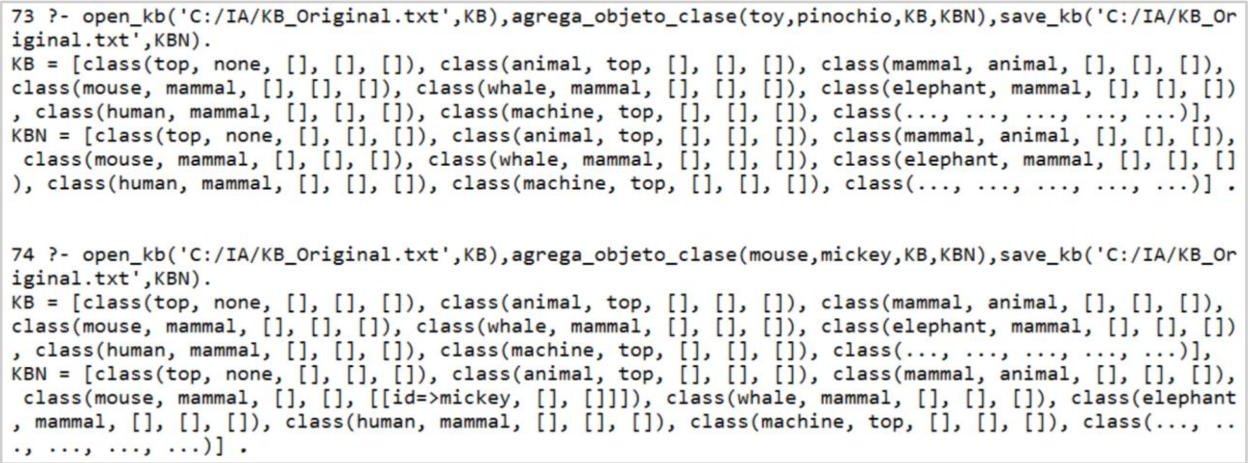
Descripción:
Agrega un nuevo objeto a una clase específica.

Argumentos:
NomClase: Nombre de la clase.
NomObjeto: Nombre del objeto nuevo.
Madre: Clase superior o madre de la clase.
Props: Propiedades actuales de la clase.
Rels: Relaciones actuales de la clase.
Insts: Instancias actuales de la clase.
Insts_New: Lista de instancias actuales concatenada con la nueva instancia añadida.

Código:

```
agrega_objeto_clase(NomClase,NomObjeto,[class(NomClase,Madre,Props,Rels,Insts)|T],[class(No
mClase,Madre,Props,Rels,Insts_New)|T]) :-
       append(Insts, [[id=>NomObjeto,[],[]]], Insts_New).
agrega_objeto_clase(NomClase,NomObjeto,[H|T],[H|R]) :-
       agrega_objeto_clase(NomClase, NomObjeto, T, R).
```

Ejemplos de uso:



```
73 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_objeto_clase(toy,pinochio,KB,KBN),save_kb('C:/IA/KB_Or
iginal.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
class(mouse, mammal, [], [], []), class(whale, mammal, [], [], []), class(elephant, mammal, [], [], [])
, class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
 class(mouse, mammal, [], [], []), class(whale, mammal, [], [], []), class(elephant, mammal, [], [], []
), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)] .


74 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_objeto_clase(mouse,mickey,KB,KBN),save_kb('C:/IA/KB_Or
iginal.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
class(mouse, mammal, [], [], []), class(whale, mammal, [], [], []), class(elephant, mammal, [], [], [])
, class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
 class(mouse, mammal, [], [], [[id=>mickey, [], []]]), class(whale, mammal, [], [], []), class(elephant
, mammal, [], [], []), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..
., ..., ..., ...)] .
```

b) Propiedades nuevas a clases u objetos.
Nombre: *agregra_propiedad_clase(...)*

Descripción:
Agrega una nueva propiedad a una clase en formato de átomo, Atributo => Valor, not (átomo) o not (Atributo => Valor).

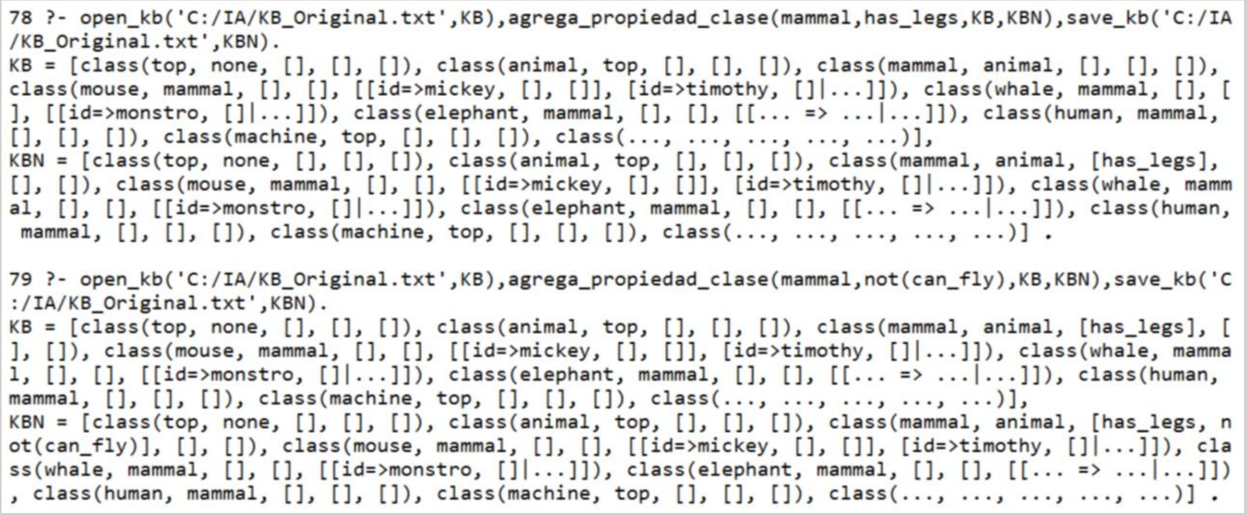
Argumentos:

NomClase: Nombre de la clase.
Propiedad: Nueva propiedad a añadir.
Madre: Clase superior o madre de la clase.
Props: Propiedades actuales de la clase.
Rels: Relaciones actuales de la clase.
Insts: Instancias actuales de la clase.
Props_New: Lista de propiedades actuales concatenada con la nueva propiedad añadida.

Código:

```
agrega_propiedad_clase(NomClase,Propiedad,[class(NomClase,Madre,Props,Rels,Insts)|T],[class
(NomClase,Madre,Props_New,Rels,Insts)|T]) :-
	append(Props, [Propiedad], Props_New).
agrega_propiedad_clase(NomClase,Propiedad,[H|T],[H|R]) :-
	agrega_propiedad_clase(NomClase, Propiedad, T, R).
```

Ejemplos de uso:



```
78 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_propiedad_clase(mammal,has_legs,KB,KBN),save_kb('C:/IA
/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale, mammal, [], [
], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]]), class(human, mammal,
[], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs],
[], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale, mamm
al, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]]), class(human,
 mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)] .

79 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_propiedad_clase(mammal,not(can_fly),KB,KBN),save_kb('C
:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs], [
], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale, mamma
l, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]]), class(human,
mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]])
, class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)] .
```

Nombre: *agregra_propiedad_objeto(...)*

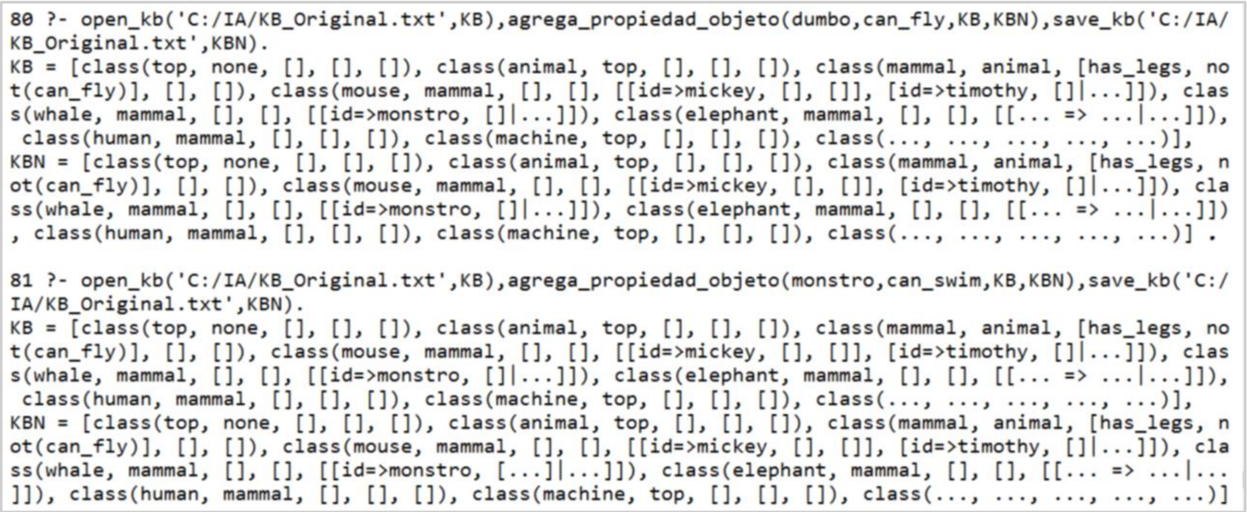
Descripción:
Agrega una nueva propiedad a un objeto en formato de átomo, Atributo => Valor, not (átomo) o not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre del objeto.
Propiedad: Nueva propiedad a añadir.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
agrega_propiedad_objeto(NomObjeto,Propiedad,KB_Original,KB_Nuevo) :-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Nuevo),
	verifica_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],Insts),
	reemplaza_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],[id=>NomObjeto,PropsObjeto
_New,RelsObjeto],Insts,Insts_New),
	append(PropsObjeto,[Propiedad],PropsObjeto_New).
```

Ejemplos de uso:



```
80 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_propiedad_objeto(dumbo,can_fly,KB,KBN),save_kb('C:/IA/
KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]]),
 class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]])
, class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)] .

81 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_propiedad_objeto(monstro,can_swim,KB,KBN),save_kb('C:/
IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, []|...]]), class(elephant, mammal, [], [], [[... => ...|...]]),
 class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [], [[... => ...|...
]]), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)]
```

c) Relaciones nuevas a clases u objetos.
Nombre: *agregra_relacion_clase(...)*

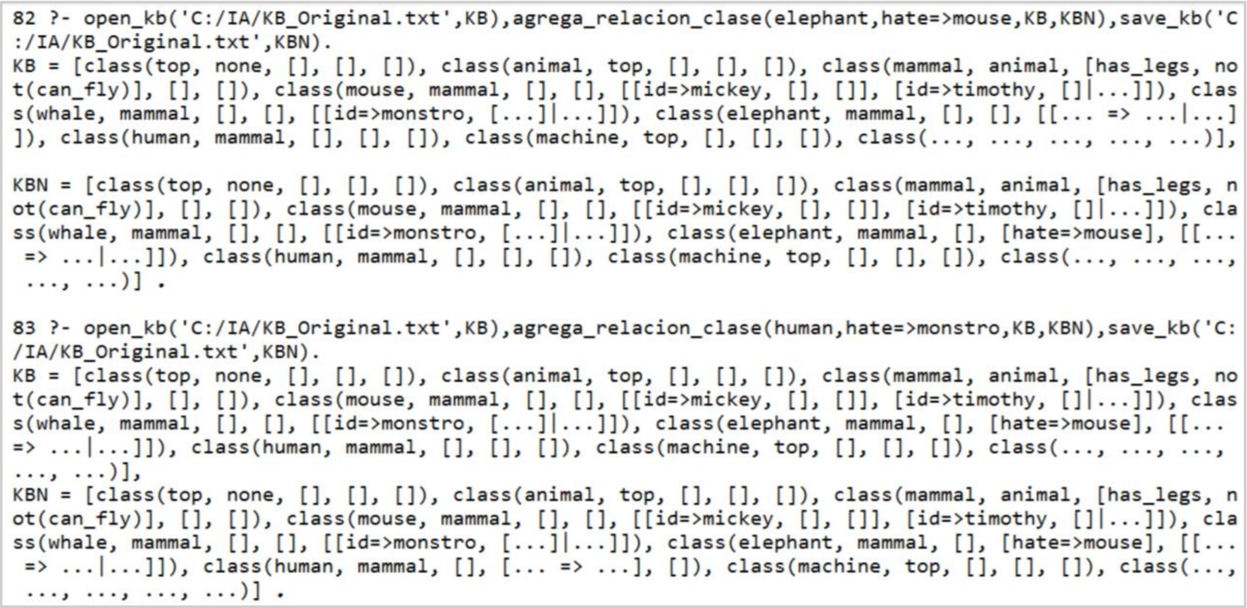
Descripción:
Agrega una nueva relación a una clase en formato Atributo => Valor y not (Atributo => Valor).

Argumentos:
NomClase: Nombre de la clase.
Relacion: Nueva relación a añadir.
Madre: Clase superior o madre de la clase.
Props: Propiedades actuales de la clase.
Rels: Relaciones actuales de la clase.
Insts: Instancias actuales de la clase.
Rels_New: Lista de relaciones actuales concatenada con la nueva relación añadida.

Código:

```
agrega_relacion_clase(NomClase,Relacion,[class(NomClase,Madre,Props,Rels,Insts)|T],[class(N
omClase,Madre,Props,Rels_New,Insts)|T]) :-
       append(Rels, [Relacion], Rels_New).
agrega_relacion_clase(NomClase,Relacion,[H|T],[H|R]) :-
       agrega_relacion_clase(NomClase, Relacion, T, R).
```

Ejemplos de uso:



```
82 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_relacion_clase(elephant,hate=>mouse,KB,KBN),save_kb('C
:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [], [[... => ...|...]
]), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)],

KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
 => ...|...]]), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ...,
 ..., ...)] .

83 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_relacion_clase(human,hate=>monstro,KB,KBN),save_kb('C:
/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
=> ...|...]]), class(human, mammal, [], [], []), class(machine, top, [], [], []), class(..., ..., ...,
..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
 => ...|...]]), class(human, mammal, [], [... => ...], []), class(machine, top, [], [], []), class(...,
 ..., ..., ..., ...)] .
```

Nombre: *agregra_relacion_objeto(...)*

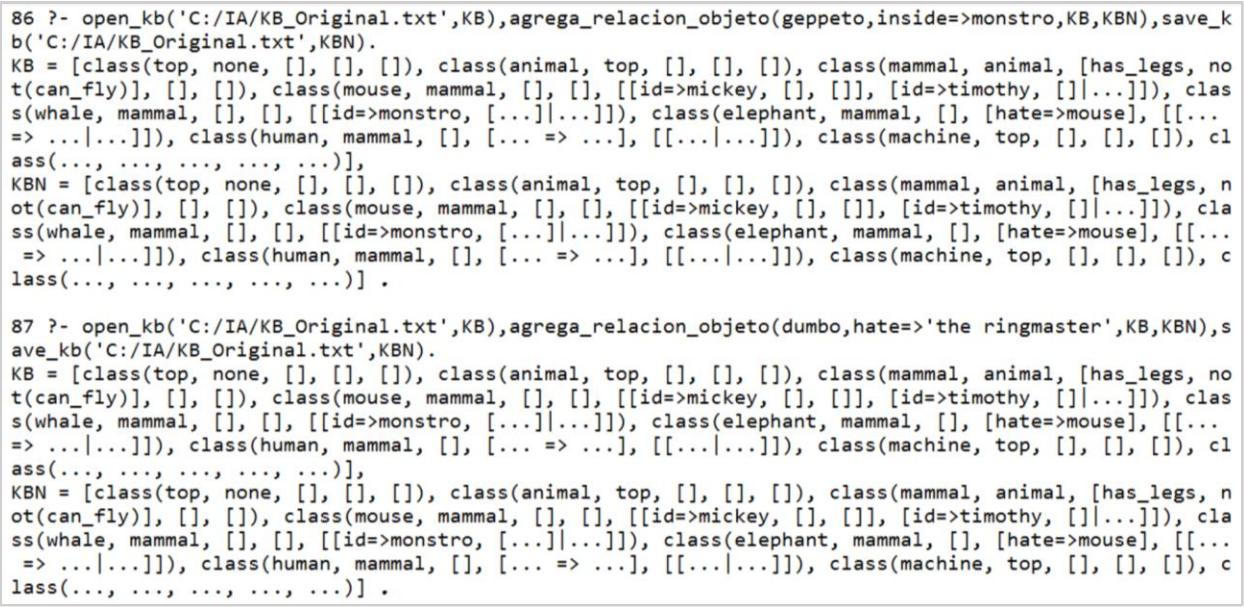
Descripción:
Agrega una nueva relación a un objeto en formato Atributo => Valor y not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre del objeto
Relacion: Nueva relación a añadir.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
agrega_relacion_objeto(NomObjeto,Relacion,KB_Original,KB_Nuevo) :-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Nuevo),
	verifica_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],Insts),
	reemplaza_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],[id=>NomObjeto,PropsObjeto
,RelsObjeto_New],Insts,Insts_New),
	append(RelsObjeto,[Relacion],RelsObjeto_New).
```

Ejemplos de uso:



```
86 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_relacion_objeto(geppeto,inside=>monstro,KB,KBN),save_k
b('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
=> ...|...]]), class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), cl
ass(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
 => ...|...]]), class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), c
lass(..., ..., ..., ..., ...)] .

87 ?- open_kb('C:/IA/KB_Original.txt',KB),agrega_relacion_objeto(dumbo,hate=>'the ringmaster',KB,KBN),s
ave_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
=> ...|...]]), class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), cl
ass(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
 => ...|...]]), class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), c
lass(..., ..., ..., ..., ...)] .
```

3. Crear predicados para eliminar:

a) Clases u objetos

Nombre: *elimina_clase(...)*

Descripción:

Elimina una clase de la base de conocimiento.

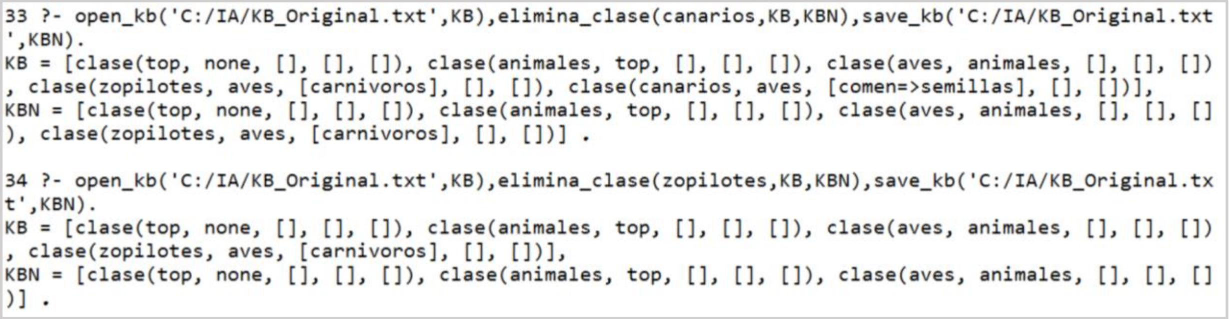
Argumentos:

NomClase: Nombre de la nueva clase a eliminar

Código:

```
elimina_clase(NomClase, [clase(NomClase,_,_,_,_)|T], T).
elimina_clase(NomClase, [H|T], [H|R]) :- elimina_clase(NomClase, T,R).
```

Ejemplos de uso:



```
33 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_clase(canarios,KB,KBN),save_kb('C:/IA/KB_Original.txt
',KBN).
KB = [clase(top, none, [], [], []), clase(animales, top, [], [], []), clase(aves, animales, [], [], [])
, clase(zopilotes, aves, [carnivoros], [], []), clase(canarios, aves, [comen=>semillas], [], [])],
KBN = [clase(top, none, [], [], []), clase(animales, top, [], [], []), clase(aves, animales, [], [], []
), clase(zopilotes, aves, [carnivoros], [], [])] .

34 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_clase(zopilotes,KB,KBN),save_kb('C:/IA/KB_Original.tx
t',KBN).
KB = [clase(top, none, [], [], []), clase(animales, top, [], [], []), clase(aves, animales, [], [], [])
, clase(zopilotes, aves, [carnivoros], [], [])],
KBN = [clase(top, none, [], [], []), clase(animales, top, [], [], []), clase(aves, animales, [], [], []
)] .
```

Nombre: *elimina_objeto(...)*

Descripción:

Elimina un objeto de la base de conocimiento.

Argumentos:

NomClase: Nombre del objeto a eliminar.

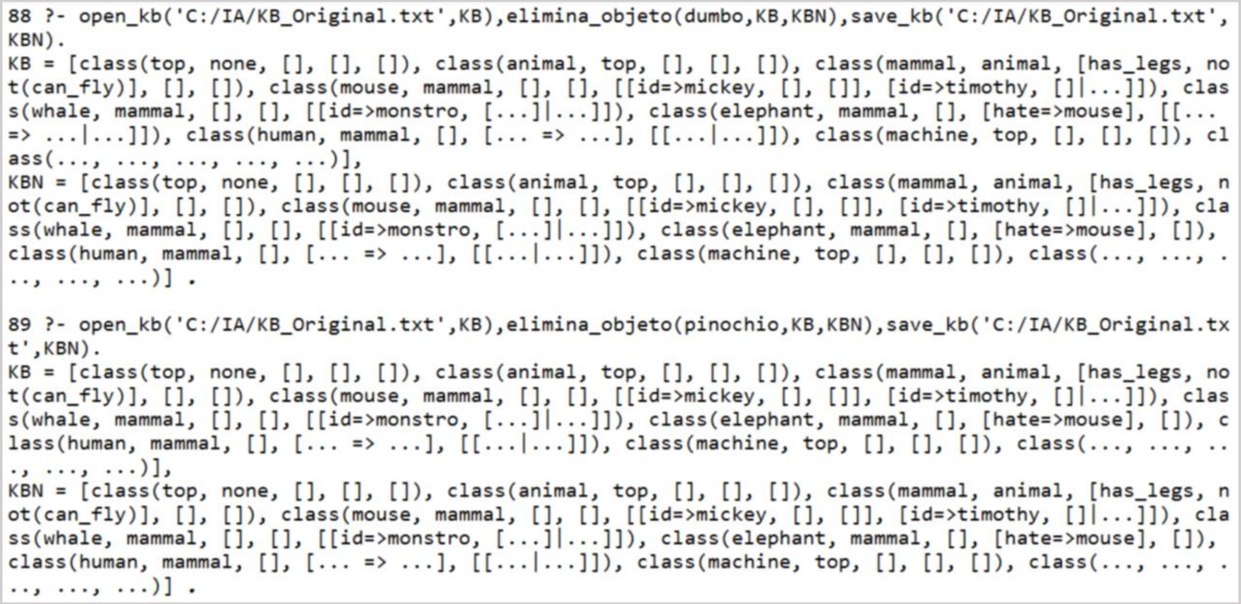
KB_Original: Base de conocimiento (Input) en formato lista.

KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
elimina_objeto(NomObjeto,KB_Original,KB_Nuevo) :-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Aux1),
	verifica_elemento([id=>NomObjeto,_,_],Insts),
	elimina_elemento([id=>NomObjeto,_,_],Insts,Insts_New),
	elimina_toda_relacion(NomObjeto,KB_Aux1,KB_Nuevo).
```

Ejemplos de uso:



```
88 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_objeto(dumbo,KB,KBN),save_kb('C:/IA/KB_Original.txt',
KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], [[...
=> ...|...]]), class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), cl
ass(..., ..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []),
class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., .
.., ..., ...)] .

89 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_objeto(pinochio,KB,KBN),save_kb('C:/IA/KB_Original.tx
t',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []), c
lass(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., ..
., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, n
ot(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), cla
ss(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []),
class(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., .
.., ..., ...)] .
```

b) Propiedades específicas de clases u objetos

Nombre: *elimina_propiedad_clase(...)*

Descripción:

Elimina una propiedad de una clase específica en formato átomo, Atributo => Valor, not (átomo) o not (Atributo => Valor).

Argumentos:

NomClase: Nombre de la clase.

Propiedad: Propiedad a eliminar.

Madre: Clase superior o madre de la clase.

Props: Propiedades actuales de la clase.

Rels: Relaciones actuales de la clase.

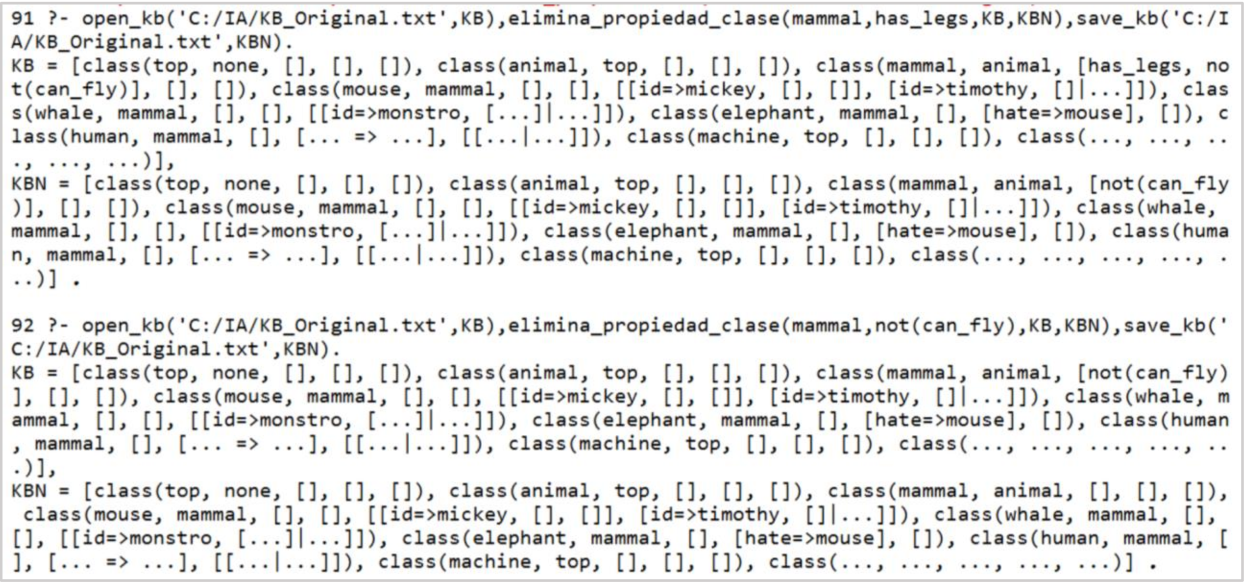
Insts: Instancias actuales de la clase.

Props_New: Lista de propiedades restantes después de eliminar la propiedad indicada.

Código:

```
elimina_propiedad_clase(NomClase,Propiedad,[class(NomClase,Madre,Props,Rels,Insts)|T],[clas
s(NomClase,Madre,Props_New,Rels,Insts)|T]) :-
	elimina_elemento(Propiedad, Props, Props_New).
elimina_propiedad_clase(NomClase,Propiedad,[H|T],[H|R]) :-
	elimina_propiedad_clase(NomClase, Propiedad, T, R).
```

Ejemplos de uso:



```
91 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_propiedad_clase(mammal,has_legs,KB,KBN),save_kb('C:/I
A/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [has_legs, no
t(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), clas
s(whale, mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []), c
lass(human, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., ..
., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [not(can_fly
)], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale,
mammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []), class(huma
n, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., ..., ..., .
..)] .

92 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_propiedad_clase(mammal,not(can_fly),KB,KBN),save_kb('
C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [not(can_fly)
], [], []), class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale, m
ammal, [], [], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []), class(human
, mammal, [], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., ..., ..., ..
.)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(mammal, animal, [], [], []),
 class(mouse, mammal, [], [], [[id=>mickey, [], []], [id=>timothy, []|...]]), class(whale, mammal, [],
[], [[id=>monstro, [...]|...]]), class(elephant, mammal, [], [hate=>mouse], []), class(human, mammal, [
], [... => ...], [[...|...]]), class(machine, top, [], [], []), class(..., ..., ..., ..., ...)] .
```

Nombre: *elimina_propiedad_objeto(...)*

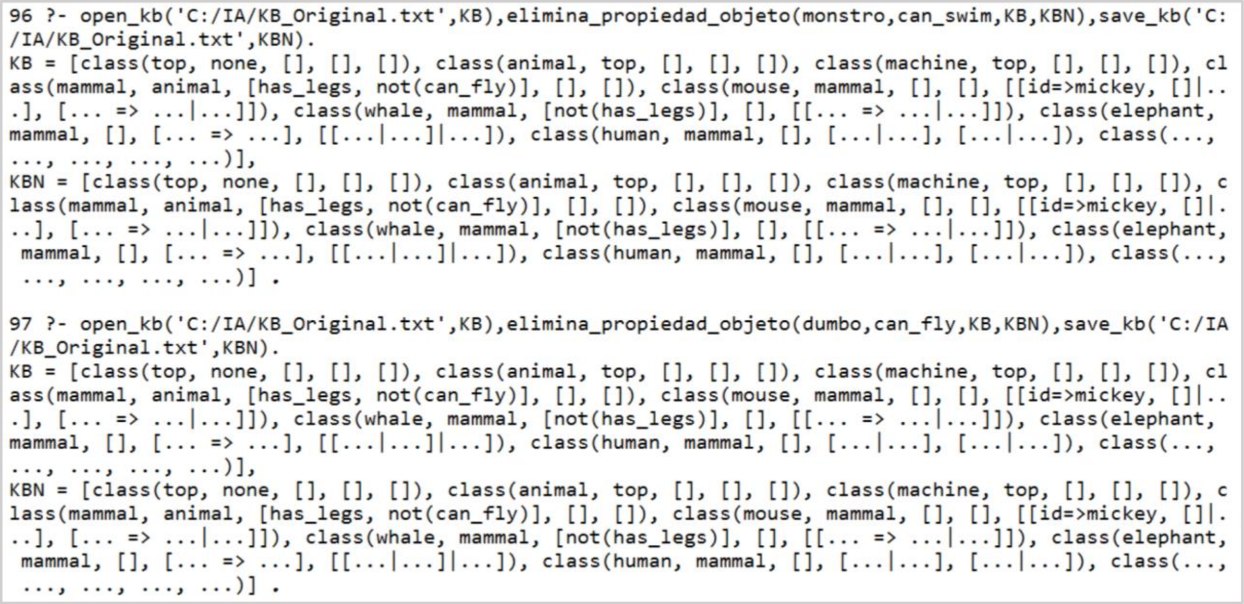
Descripción:
Elimina una propiedad de un objeto en formato átomo, Atributo => Valor, not (átomo) o not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre del objeto.
Propiedad: Propiedad a eliminar.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
elimina_propiedad_objeto(NomObjeto,Propiedad,KB_Original,KB_Nuevo) :-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Nuevo),
	verifica_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],Insts),
	reemplaza_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],[id=>NomObjeto,PropsObjeto
_New,RelsObjeto],Insts,Insts_New),
	elimina_elemento(Propiedad,PropsObjeto,PropsObjeto_New).
```

Ejemplos de uso:



```
96 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_propiedad_objeto(monstro,can_swim,KB,KBN),save_kb('C:
/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

97 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_propiedad_objeto(dumbo,can_fly,KB,KBN),save_kb('C:/IA
/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

c) Relaciones específicas de clases u objetos

Nombre: *elimina_relacion_clase(…)*

Descripción:

Elimina una relación de una clase en formato Atributo => Valor o not (Atributo => Valor).

Argumentos:

NomClase: Nombre de la clase.

Relacion: Relación a eliminar.

Madre: Clase superior o madre de la clase.

Props: Propiedades actuales de la clase.

Rels: Relaciones actuales de la clase.

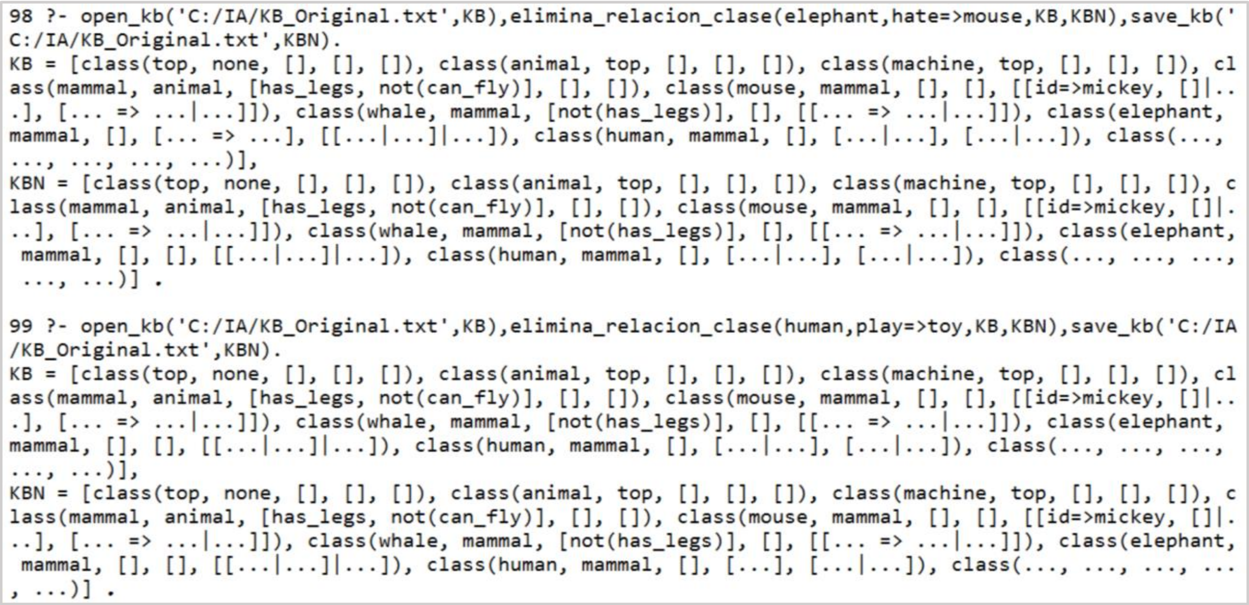
Insts: Instancias actuales de la clase.

Rels_New: Lista de relaciones restantes después de eliminar la relación indicada.

Código:

```
elimina_relacion_clase(NomClase,Relacion,[class(NomClase,Madre,Props,Rels,Insts)|T],[class(
NomClase,Madre,Props,Rels_New,Insts)|T]) :-
       elimina_elemento(Relacion, Rels, Rels_New).
elimina_relacion_clase(NomClase,Relacion,[H|T],[H|R]) :-
       elimina_relacion_clase(NomClase, Relacion, T, R).
```

Ejemplos de uso:



```
98 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_relacion_clase(elephant,hate=>mouse,KB,KBN),save_kb('
C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(..., ..., ...,
 ..., ...)] .

99 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_relacion_clase(human,play=>toy,KB,KBN),save_kb('C:/IA
/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(..., ..., ...,
..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [], [[...|...]|...]), class(human, mammal, [], [...], [...|...]), class(..., ..., ..., ...
, ...)] .
```

Nombre: *elimina_relacion_objeto(...)*

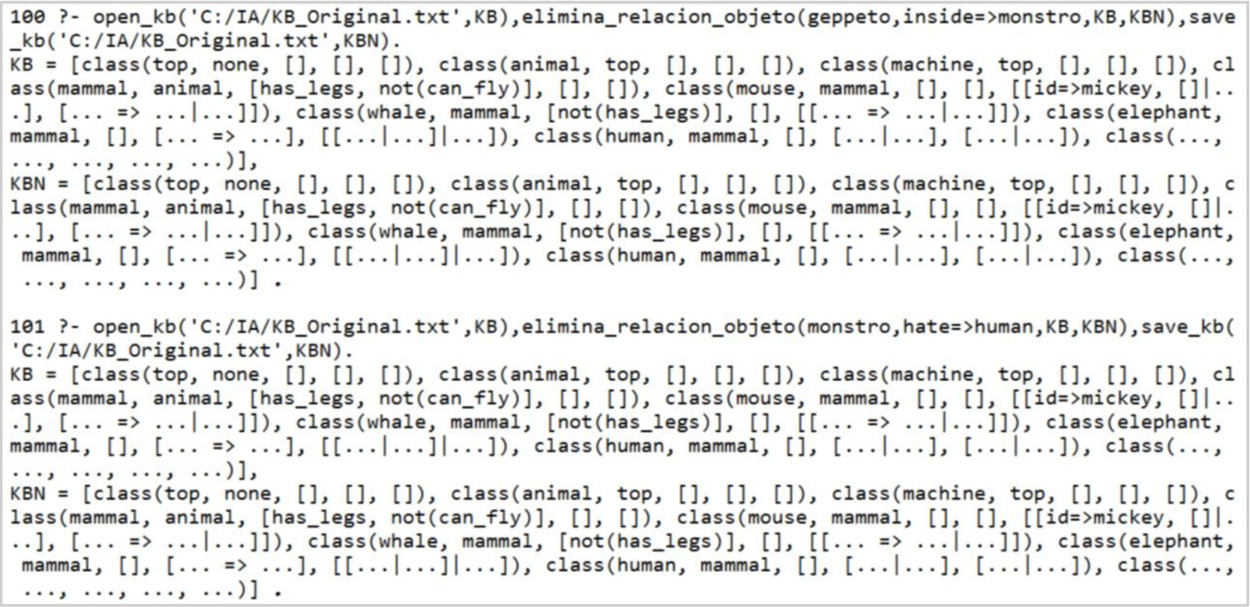
Descripción:
Elimina una relación de un objeto en formato Atributo => Valor o not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre del objeto.
Relacion: Relación a eliminar.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
elimina_relacion_objeto(NomObjeto,Relacion,KB_Original,KB_Nuevo) :-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Nuevo),
	verifica_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],Insts),
	reemplaza_elemento([id=>NomObjeto,PropsObjeto,RelsObjeto],[id=>NomObjeto,PropsObjeto
,RelsObjeto_New],Insts,Insts_New),
	elimina_elemento(Relacion,RelsObjeto,RelsObjeto_New).
```

Ejemplos de uso:



```
100 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_relacion_objeto(geppeto,inside=>monstro,KB,KBN),save
_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

101 ?- open_kb('C:/IA/KB_Original.txt',KB),elimina_relacion_objeto(monstro,hate=>human,KB,KBN),save_kb(
'C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

4. Crear predicados para modificar:

a) El nombre de una clase u objeto
Nombre: *modifica_nombre_clase(…)*

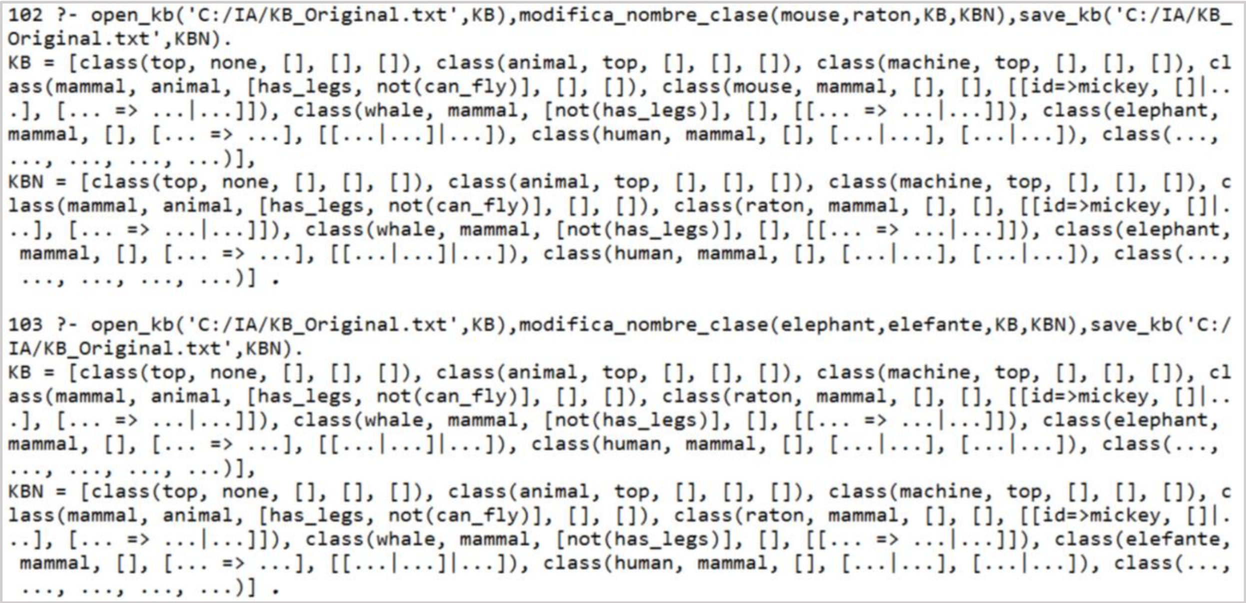
Descripción:
Modifica el nombre de una clase en la base de conocimiento.

Argumentos:
NomClase: Nombre actual de la clase.
NomClase_New: Nombre nuevo de la clase.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_nombre_clase(NomClase,NomClase_New,KB_Original,KB_Nuevo):-
	reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase_New,Madre,P
rops,Rels,Insts),KB_Original,KB_Aux),
	actualiza_clase_madre(NomClase,NomClase_New,KB_Aux,KB_Aux1),
	actualiza_toda_relacion(NomClase,NomClase_New,KB_Aux1,KB_Nuevo).
```

Ejemplos de uso:



```
102 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_nombre_clase(mouse,raton,KB,KBN),save_kb('C:/IA/KB_
Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

103 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_nombre_clase(elephant,elefante,KB,KBN),save_kb('C:/
IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

Nombre: *modifica_nombre_objeto(...)*

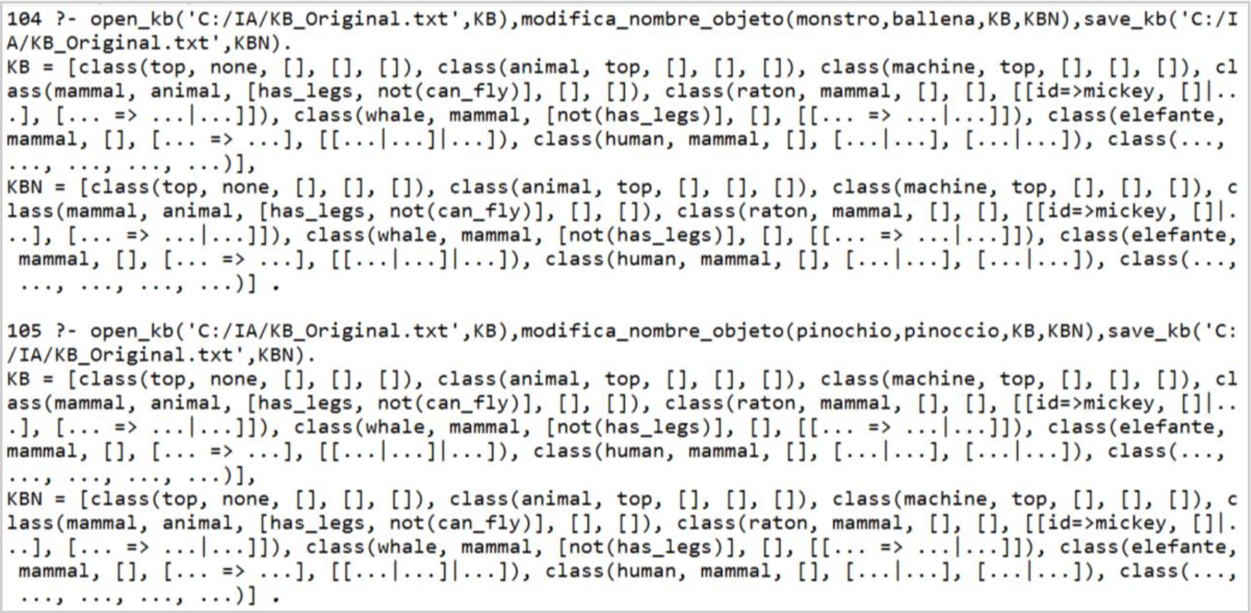
Descripción:
Modifica el nombre de un objeto en la base de conocimiento.

Argumentos:
NomObjeto: Nombre actual del objeto.
NomObjeto_New: Nombre nuevo del objeto.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_nombre_objeto(NomObjeto,NomObjeto_New,KB_Original,KB_Nuevo) :-
       reemplaza_elemento(class(NomClase,Madre,Props,Rels,Insts),class(NomClase,Madre,Props
,Rels,Insts_New),KB_Original,KB_Aux),
       verifica_elemento([id=>NomObjeto|T],Insts),
       reemplaza_elemento([id=>NomObjeto|T],[id=>NomObjeto_New|T],Insts,Insts_New),
       actualiza_toda_relacion(NomObjeto,NomObjeto_New,KB_Aux,KB_Nuevo).
```

Ejemplos de uso:



```
104 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_nombre_objeto(monstro,ballena,KB,KBN),save_kb('C:/I
A/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

105 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_nombre_objeto(pinochio,pinoccio,KB,KBN),save_kb('C:
/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

b) El valor de una propiedad específica de una clase u objeto
Nombre: *modifica_propiedad_clase(...)*

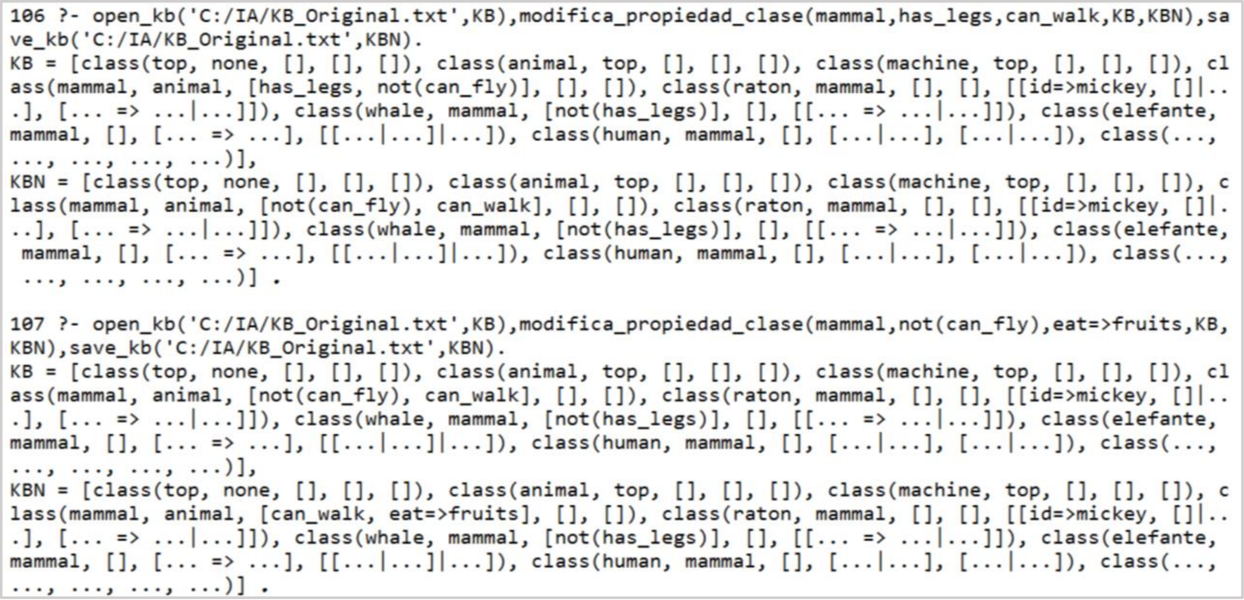
Descripción:
Modifica una propiedad específica de una clase utilizando el formato átomo, Atributo => Valor, not(átomo) o not (Atributo => Valor).

Argumentos:
NomClase: Nombre actual de la clase.
Propiedad: Nombre actual de la propiedad.
Propiedad_New: Nombre nuevo de la propiedad.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_propiedad_clase(NomClase,Propiedad,Propiedad_New,KB_Original,KB_Nuevo) :-
       elimina_propiedad_clase(NomClase,Propiedad,KB_Original,KB_Aux),
       agrega_propiedad_clase(NomClase,Propiedad_New,KB_Aux,KB_Nuevo).
```

Ejemplos de uso:



```
106 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_propiedad_clase(mammal,has_legs,can_walk,KB,KBN),sa
ve_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [not(can_fly), can_walk], [], []), class(raton, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

107 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_propiedad_clase(mammal,not(can_fly),eat=>fruits,KB,
KBN),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [not(can_fly), can_walk], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [can_walk, eat=>fruits], [], []), class(raton, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elefante,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)] .
```

Nombre: *modifica_propiedad_objeto(...)*

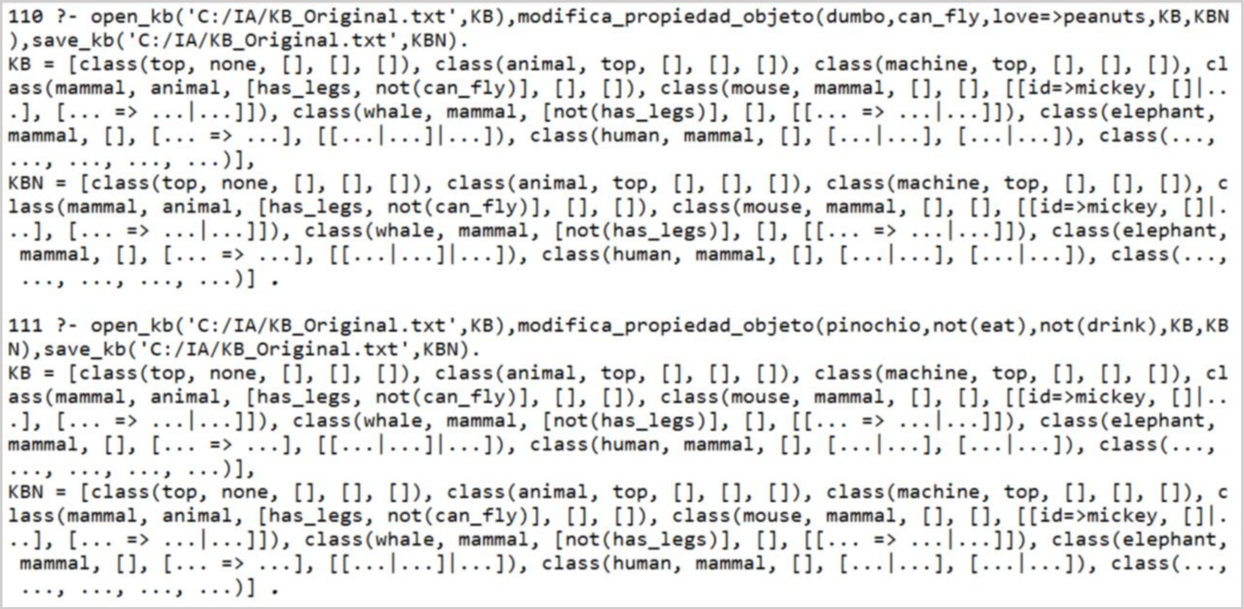
Descripción:
Modifica una propiedad específica de un objeto utilizando el formato átomo, Atributo => Valor, not(átomo) o not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre actual del objeto.
Propiedad: Nombre actual de la propiedad.
Propiedad_New: Nombre nueva de la propiedad.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_propiedad_objeto(NomObjeto,Propiedad,Propiedad_New,KB_Original,KB_Nuevo) :-
       elimina_propiedad_objeto(NomObjeto,Propiedad,KB_Original,KB_Aux),
       agrega_propiedad_objeto(NomObjeto,Propiedad_New,KB_Aux,KB_Nuevo).
```

Ejemplos de uso:



```
110 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_propiedad_objeto(dumbo,can_fly,love=>peanuts,KB,KBN
),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

111 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_propiedad_objeto(pinochio,not(eat),not(drink),KB,KB
N),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

c) Con quien mantiene una relación específica una clase u objeto
Nombre: *modifica_relacion_clase(...)*

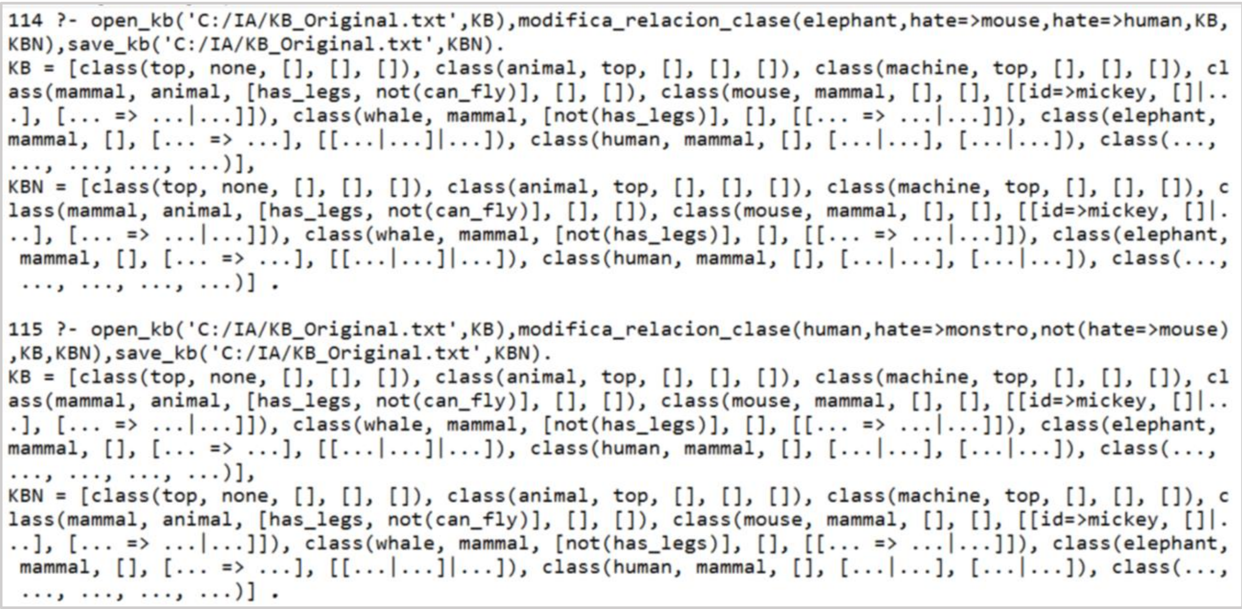
Descripción:
Modifica el valor de una relación específica de una clase utilizando el formato Atributo => Valor o not (Atributo => Valor).

Argumentos:
NomClase: Nombre actual de la clase.
Relacion: Nombre actual de la relación.
Relacion_New: Nombre nuevo de la relación.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_relacion_clase(NomClase,Relacion,Relacion_New,KB_Original,KB_Nuevo) :-
              elimina_relacion_clase(NomClase,Relacion,KB_Original,KB_Aux),
              agrega_relacion_clase(NomClase,Relacion_New,KB_Aux,KB_Nuevo).
```

Ejemplos de uso:



```
114 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_relacion_clase(elephant,hate=>mouse,hate=>human,KB,
KBN),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

115 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_relacion_clase(human,hate=>monstro,not(hate=>mouse)
,KB,KBN),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```

Nombre: *modifica_relacion_objeto(...)*

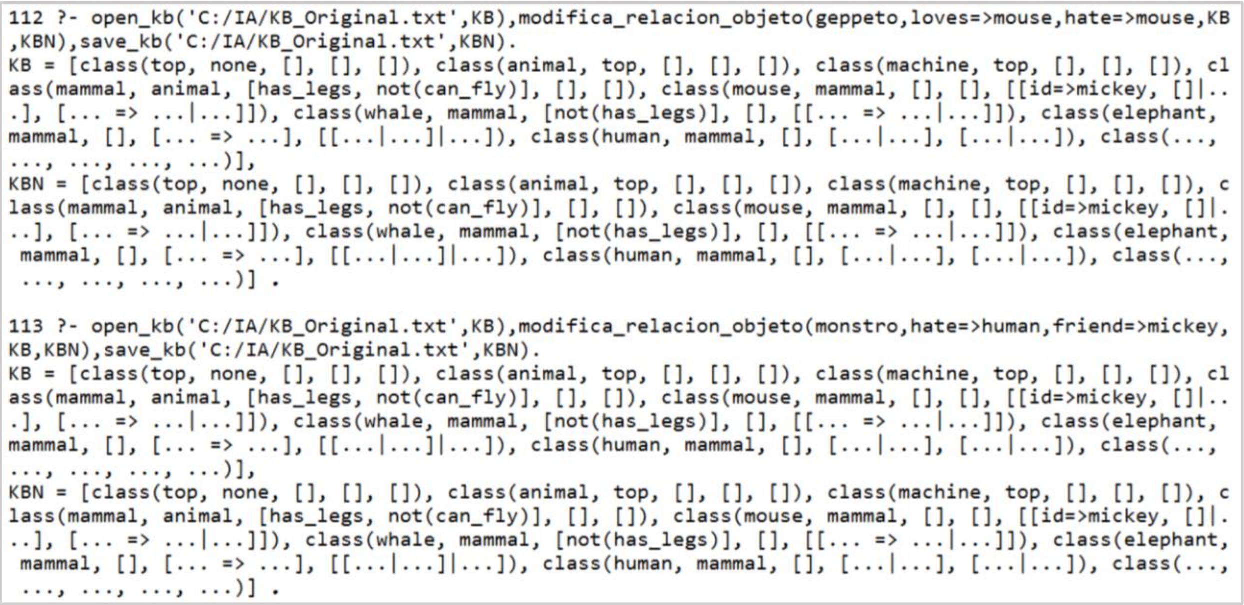
Descripción:
Modifica el valor de una relación específica de un objeto utilizando el formato Atributo => Valor o not (Atributo => Valor).

Argumentos:
NomObjeto: Nombre actual del objeto.
Relacion: Nombre actual de la relación.
Relacion_New: Nombre nuevo de la relación.
KB_Original: Base de conocimiento (Input) en formato lista.
KB_Nuevo: Base de conocimiento modificada (Output) en formato lista.

Código:

```
modifica_relacion_objeto(NomObjeto,Relacion,Relacion_New,KB_Original,KB_Nuevo) :-
       elimina_relacion_objeto(NomObjeto,Relacion,KB_Original,KB_Aux),
       agrega_relacion_objeto(NomObjeto,Relacion_New,KB_Aux,KB_Nuevo).
```

Ejemplos de uso:



```
112 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_relacion_objeto(geppeto,loves=>mouse,hate=>mouse,KB
,KBN),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .

113 ?- open_kb('C:/IA/KB_Original.txt',KB),modifica_relacion_objeto(monstro,hate=>human,friend=>mickey,
KB,KBN),save_kb('C:/IA/KB_Original.txt',KBN).
KB = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), cl
ass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|..
.], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
..., ..., ..., ...)],
KBN = [class(top, none, [], [], []), class(animal, top, [], [], []), class(machine, top, [], [], []), c
lass(mammal, animal, [has_legs, not(can_fly)], [], []), class(mouse, mammal, [], [], [[id=>mickey, []|.
..], [... => ...|...]]), class(whale, mammal, [not(has_legs)], [], [[... => ...|...]]), class(elephant,
 mammal, [], [... => ...], [[...|...]|...]), class(human, mammal, [], [...|...], [...|...]), class(...,
 ..., ..., ..., ...)] .
```