Integrantes: Tarcísio Monteiro, João Gabriel Andrade, Victoria Pantoja, Victor Barros

Primeira Tarefa:

$(\exists x)\{H(x) \land P(x) \land (\forall y)[(H(y) \land P(y)) \rightarrow y = x]\}$
$\land (\exists x)\{G(x) \land P(x) \land (\forall y)[(G(y) \land P(y)) \rightarrow y = x]\}$
$\land (\exists x)\{B(x) \land P(x) \land (\forall y)[(B(y) \land P(y)) \rightarrow y = x]\}$
$\land (\exists x)\{I(x) \land P(x) \land (\forall y)[(I(y) \land P(y)) \rightarrow y = x]\}$

$(\exists x)(H(x) \land (\forall y)(H(y) \rightarrow y = clark))$

$(\exists x)(I(x) \land (\exists y)(B(y) \land C(x, y)) \land (\forall z)(I(z) \rightarrow z = x))$

$(\exists x)(B(x) \land (\forall y)(B(y) \rightarrow (y = x \land M(y, clark))))$

$(\exists x)[B(x) \land (\forall y)\{B(y) \rightarrow [x = y \land E(y) \land (\exists z)(G(z) \land L(y, z))]\}]$

$(\exists x)[H(x) \land (\forall y)\{H(y) \rightarrow [y = x \land (\exists z)(G(z) \land L(y, z))]\}]$
$\land (\exists x)[I(x) \land (\forall y)\{I(y) \rightarrow [y = x \land (\exists z)(H(z) \land L(y, z))]\}]$

$(\forall x)[P(x) \rightarrow (S(x) \lor D(x) \lor E(x))]$

$E(lex) \land S(lana) \land S(chloe) \land D(clark)$

$(\forall x)[(L(x, lana) \land x \neq clark) \rightarrow M(x, clark)]$

$L(harry, lana) \land L(clark, lana) \land L(lex, lana)$

$L(chloe, clark)$

$D = \{chloe, clark, lex, harry, lana\}$

P(x é personagem de uma série)
L(x é apaixonado(a) por y)
C(x chantageia y)
E(x é empresário)
S(x é estudante)
D(x trabalha numa lanchonete)
M(x é um irmão perdido de y)
H(x é o mocinho)
G(x é a mocinha)
B(x é o bandido)
I(x é a invejosa)

$(\forall x)[(E(x) \wedge L(x, lana) \rightarrow C(chloe, x)]$

Segunda Tarefa:

```
%% Uma série de TV tem como personagens um Mocinho, um Bandido, uma Mocinha e uma Invejosa.
%% Todo personagem é ou estudante, ou trabalha numa lanchonete ou é empresário.
character(X) :- good_guy(X), student(X).
character(X) :- good_guy(X), businessman(X).
character(X) :- good_guy(X), staff(X).
character(X) :- bad_guy(X), student(X).
character(X) :- bad_guy(X), businessman(X).
character(X) :- bad_guy(X), staff(X).
character(X) :- the_girl(X), student(X).
character(X) :- the_girl(X), businessman(X).
character(X) :- the_girl(X), staff(X).
character(X) :- jealous_girl(X), student(X).
character(X) :- jealous_girl(X), businessman(X).
character(X) :- jealous_girl(X), staff(X).

%% O Mocinho sempre é Clark.
good_guy(clark).

%% O Bandido é o irmão perdido na infância do Mocinho.
%% O Bandido é sempre um empresário que é apaixonado pela Mocinha.
bad_guy(X) :- good_guy(Y), lost_brother(X, Y).
bad_guy(X) :- businessman(X), the_girl(Y), loves(X, Y).

%% O Mocinho é apaixonado pela Mocinha e a Invejosa é apaixonada pelo Mocinho.
the_girl(X) :- good_guy(Y), loves(Y, X).

%% A Invejosa chantagea o Bandido.
jealous_girl(X) :- good_guy(Y), loves(X, Y).
jealous_girl(X) :- bad_guy(Y), blackmails(X, Y).

%% Lex é empresário, Lana e Chloé são estudantes e Clark trabalha numa lanchonete.
businessman(lex).
student(lana).
student(chloe).
staff(clark).

%% Todos os apaixonados por Lana, com exceção de Clark, são irmãos perdidos de Clark.
lost_brother(X, clark) :- dif(X, clark), loves(X, lana).

%% Harry, Clark e Lex são apaixonados por Lana.
%% Chloé é apaixonada por Clark.
loves(harry, lana).
loves(clark, lana).
loves(lex, lana).
loves(chloe, clark).
```

```
%% Chloé chantagea todos os empresários que são apaixonados por Lana.
blackmails(chloe, X) :- businessman(X), loves(X, lana).
```

Terceira Tarefa:

Quem são os personagens de uma série de TV?

```
?- character(X).
   Call: (7) character(_G2554) ? creep
   Call: (8) good_guy(_G2554) ? creep
   Exit: (8) good_guy(clark) ? creep
   Call: (8) student(clark) ? creep
   Fail: (8) student(clark) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) good_guy(_G2554) ? creep
   Exit: (8) good_guy(clark) ? creep
   Call: (8) businessman(clark) ? creep
   Fail: (8) businessman(clark) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) good_guy(_G2554) ? creep
   Exit: (8) good_guy(clark) ? creep
   Call: (8) staff(clark) ? creep
   Exit: (8) staff(clark) ? creep
   Exit: (7) character(clark) ? creep
X = clark ;
   Redo: (7) character(_G2554) ? creep
   Call: (8) bad_guy(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) lost_brother(_G2554, clark) ?
creep
   Call: (10) dif:dif(_G2554, clark) ? creep
   Exit: (10) dif:dif(_G2554{dif = ...},
clark) ? creep
   Call: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(harry, lana) ? creep
   Exit: (9) lost_brother(harry, clark) ?
creep
   Exit: (8) bad_guy(harry) ? creep
   Call: (8) student(harry) ? creep
   Fail: (8) student(harry) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Redo: (10) loves(clark, lana) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) lost_brother(lex, clark) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) student(lex) ? creep
   Fail: (8) student(lex) ? creep
   Redo: (8) bad_guy(_G2554) ? creep
   Call: (9) businessman(_G2554) ? creep
   Exit: (9) businessman(lex) ? creep
   Call: (9) the_girl(_G2628) ? creep
   Call: (10) good_guy(_G2628) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) loves(clark, _G2629) ? creep
   Exit: (10) loves(clark, lana) ? creep
   Exit: (9) the_girl(lana) ? creep
   Call: (9) loves(lex, lana) ? creep
   Exit: (9) loves(lex, lana) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) student(lex) ? creep
   Fail: (8) student(lex) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) bad_guy(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) lost_brother(_G2554, clark) ?
creep
   Call: (10) dif:dif(_G2554, clark) ? creep
   Exit: (10) dif:dif(_G2554{dif = ...},
clark) ? creep
   Call: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(harry, lana) ? creep
   Exit: (9) lost_brother(harry, clark) ?
creep
   Exit: (8) bad_guy(harry) ? creep
   Call: (8) businessman(harry) ? creep
   Fail: (8) businessman(harry) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Redo: (10) loves(clark, lana) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) lost_brother(lex, clark) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) businessman(lex) ? creep
   Exit: (8) businessman(lex) ? creep
   Exit: (7) character(lex) ? creep
X = lex ;
   Redo: (8) bad_guy(_G2554) ? creep
   Call: (9) businessman(_G2554) ? creep
   Exit: (9) businessman(lex) ? creep
   Call: (9) the_girl(_G2628) ? creep
   Call: (10) good_guy(_G2628) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) loves(clark, _G2629) ? creep
   Exit: (10) loves(clark, lana) ? creep
   Exit: (9) the_girl(lana) ? creep
   Call: (9) loves(lex, lana) ? creep
   Exit: (9) loves(lex, lana) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) businessman(lex) ? creep
   Exit: (8) businessman(lex) ? creep
   Exit: (7) character(lex) ? creep
X = lex ;
   Redo: (7) character(_G2554) ? creep
   Call: (8) bad_guy(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) lost_brother(_G2554, clark) ?
creep
   Call: (10) dif:dif(_G2554, clark) ? creep
   Exit: (10) dif:dif(_G2554{dif = ...},
clark) ? creep
   Call: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(harry, lana) ? creep
   Exit: (9) lost_brother(harry, clark) ?
creep
   Exit: (8) bad_guy(harry) ? creep
   Call: (8) staff(harry) ? creep
   Fail: (8) staff(harry) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Redo: (10) loves(clark, lana) ? creep
   Redo: (10) loves(_G2554{dif = ...}, lana)
? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) lost_brother(lex, clark) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) staff(lex) ? creep
   Fail: (8) staff(lex) ? creep
   Redo: (8) bad_guy(_G2554) ? creep
   Call: (9) businessman(_G2554) ? creep
   Exit: (9) businessman(lex) ? creep
   Call: (9) the_girl(_G2628) ? creep
   Call: (10) good_guy(_G2628) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) loves(clark, _G2629) ? creep
   Exit: (10) loves(clark, lana) ? creep
   Exit: (9) the_girl(lana) ? creep
   Call: (9) loves(lex, lana) ? creep
   Exit: (9) loves(lex, lana) ? creep
   Exit: (8) bad_guy(lex) ? creep
   Call: (8) staff(lex) ? creep
   Fail: (8) staff(lex) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) the_girl(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(clark, _G2554) ? creep
   Exit: (9) loves(clark, lana) ? creep
   Exit: (8) the_girl(lana) ? creep
   Call: (8) student(lana) ? creep
   Exit: (8) student(lana) ? creep
   Exit: (7) character(lana) ? creep
X = lana ;
Redo: (7) character(_G2554) ? creep
   Call: (8) the_girl(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(clark, _G2554) ? creep
   Exit: (9) loves(clark, lana) ? creep
   Exit: (8) the_girl(lana) ? creep
   Call: (8) businessman(lana) ? creep
   Fail: (8) businessman(lana) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) the_girl(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(clark, _G2554) ? creep
   Exit: (9) loves(clark, lana) ? creep
   Exit: (8) the_girl(lana) ? creep
   Call: (8) staff(lana) ? creep
   Fail: (8) staff(lana) ? creep
   Redo: (7) character(_G2554) ? creep
   Call: (8) jealous_girl(_G2554) ? creep
   Call: (9) good_guy(_G2628) ? creep
```

```
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(_G2554, clark) ? creep
   Exit: (9) loves(chloe, clark) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) student(chloe) ? creep
   Exit: (8) student(chloe) ? creep
   Exit: (7) character(chloe) ? creep
X = chloe ;
   Redo: (8) jealous_girl(_G2362) ? creep
   Call: (9) bad_guy(_G2436) ? creep
   Call: (10) good_guy(_G2436) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) lost_brother(_G2436, clark) ?
creep
   Call: (11) dif:dif(_G2436, clark) ? creep
   Exit: (11) dif:dif(_G2432{dif = ...},
clark) ? creep
   Call: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(harry, lana) ? creep
   Exit: (10) lost_brother(harry, clark) ?
creep
   Exit: (9) bad_guy(harry) ? creep
   Call: (9) blackmails(_G2362, harry) ?
creep
   Call: (10) businessman(harry) ? creep
   Fail: (10) businessman(harry) ? creep
   Fail: (9) blackmails(_G2362, harry) ?
creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Redo: (11) loves(clark, lana) ? creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(lex, lana) ? creep
   Exit: (10) lost_brother(lex, clark) ?
creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) blackmails(chloe, lex) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) student(chloe) ? creep
   Exit: (8) student(chloe) ? creep
   Exit: (7) character(chloe) ? creep
X = chloe ;
   Redo: (9) bad_guy(_G2436) ? creep
   Call: (10) businessman(_G2436) ?
creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) the_girl(_G2436) ? creep
   Call: (11) good_guy(_G2436) ? creep
   Exit: (11) good_guy(clark) ? creep
   Call: (11) loves(clark, _G2437) ? creep
   Exit: (11) loves(clark, lana) ? creep
   Exit: (10) the_girl(lana) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) blackmails(chloe, lex) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) student(chloe) ? creep
   Exit: (8) student(chloe) ? creep
   Exit: (7) character(chloe) ? creep
X = chloe ;
   Redo: (7) character(_G2362) ? creep
   Call: (8) jealous_girl(_G2362) ? creep
   Call: (9) good_guy(_G2436) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(_G2362, clark) ? creep
   Exit: (9) loves(chloe, clark) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) businessman(chloe) ? creep
   Fail: (8) businessman(chloe) ? creep
   Redo: (8) jealous_girl(_G2362) ? creep
   Call: (9) bad_guy(_G2436) ? creep
   Call: (10) good_guy(_G2436) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) lost_brother(_G2436, clark) ?
creep
   Call: (11) dif:dif(_G2436, clark) ? creep
   Exit: (11) dif:dif(_G2432{dif = ...},
clark) ? creep
   Call: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(harry, lana) ? creep
   Exit: (10) lost_brother(harry, clark) ?
creep
   Exit: (9) bad_guy(harry) ? creep
   Call: (9) blackmails(_G2362, harry) ?
creep
   Call: (10) businessman(harry) ? creep
   Fail: (10) businessman(harry) ? creep
   Fail: (9) blackmails(_G2362, harry) ?
creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Redo: (11) loves(clark, lana) ? creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(lex, lana) ? creep
   Exit: (10) lost_brother(lex, clark) ?
creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) blackmails(chloe, lex) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) businessman(chloe) ? creep
   Fail: (8) businessman(chloe) ? creep
   Redo: (9) bad_guy(_G2436) ? creep
   Call: (10) businessman(_G2436) ?
creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) the_girl(_G2436) ? creep
   Call: (11) good_guy(_G2436) ? creep
   Exit: (11) good_guy(clark) ? creep
   Call: (11) loves(clark, _G2437) ? creep
   Exit: (11) loves(clark, lana) ? creep
   Exit: (10) the_girl(lana) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) blackmails(chloe, lex) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) businessman(chloe) ? creep
   Fail: (8) businessman(chloe) ? creep
   Redo: (7) character(_G2362) ? creep
   Call: (8) jealous_girl(_G2362) ? creep
   Call: (9) good_guy(_G2436) ? creep
   Exit: (9) good_guy(clark) ? creep
   Call: (9) loves(_G2362, clark) ? creep
   Exit: (9) loves(chloe, clark) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) staff(chloe) ? creep
   Fail: (8) staff(chloe) ? creep
   Redo: (8) jealous_girl(_G2362) ? creep
   Call: (9) bad_guy(_G2436) ? creep
   Call: (10) good_guy(_G2436) ? creep
   Exit: (10) good_guy(clark) ? creep
   Call: (10) lost_brother(_G2436, clark) ?
creep
   Call: (11) dif:dif(_G2436, clark) ? creep
   Exit: (11) dif:dif(_G2432{dif = ...},
clark) ? creep
   Call: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(harry, lana) ? creep
   Exit: (10) lost_brother(harry, clark) ?
creep
   Exit: (9) bad_guy(harry) ? creep
   Call: (9) blackmails(_G2362, harry) ?
creep
   Call: (10) businessman(harry) ? creep
   Fail: (10) businessman(harry) ? creep
   Fail: (9) blackmails(_G2362, harry) ?
creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Redo: (11) loves(clark, lana) ? creep
   Redo: (11) loves(_G2432{dif = ...}, lana)
? creep
   Exit: (11) loves(lex, lana) ? creep
   Exit: (10) lost_brother(lex, clark) ?
creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) blackmails(chloe, lex) ? creep
   Exit: (8) jealous_girl(chloe) ? creep
   Call: (8) staff(chloe) ? creep
   Fail: (8) staff(chloe) ? creep
   Redo: (9) bad_guy(_G2436) ? creep
   Call: (10) businessman(_G2436) ?
creep
   Exit: (10) businessman(lex) ? creep
   Call: (10) the_girl(_G2436) ? creep
   Call: (11) good_guy(_G2436) ? creep
   Exit: (11) good_guy(clark) ? creep
   Call: (11) loves(clark, _G2437) ? creep
   Exit: (11) loves(clark, lana) ? creep
   Exit: (10) the_girl(lana) ? creep
   Call: (10) loves(lex, lana) ? creep
   Exit: (10) loves(lex, lana) ? creep
   Exit: (9) bad_guy(lex) ? creep
   Call: (9) blackmails(_G2362, lex) ?
creep
   Call: (10) businessman(lex) ? creep
   Exit: (10) businessman(lex) ? creep
```

Call: (10) loves(lex, lana) ? creep
Exit: (10) loves(lex, lana) ? creep
Exit: (9) blackmails(chloe, lex) ? creep
Exit: (8) jealous_girl(chloe) ? creep
Call: (8) staff(chloe) ? creep
Fail: (8) staff(chloe) ? creep
Fail: (7) character(_G2362) ? creep
false.

Quem são os irmãos perdidos de Clark?

```
?- lost_brother(X, clark).
   Call: (7) lost_brother(_G2974, clark) ? creep
   Call: (8) dif:dif(_G2974, clark) ? creep
   Exit: (8) dif:dif(_G2974{dif = ...}, clark) ? creep
   Call: (8) loves(_G2974{dif = ...}, lana) ? creep
   Exit: (8) loves(harry, lana) ? creep
   Exit: (7) lost_brother(harry, clark) ? creep
X = harry ;
   Redo: (8) loves(_G2974{dif = ...}, lana) ? creep
   Redo: (8) loves(clark, lana) ? creep
   Redo: (8) loves(_G2974{dif = ...}, lana) ? creep
   Exit: (8) loves(lex, lana) ? creep
   Exit: (7) lost_brother(lex, clark) ? creep
X = lex.
```

Dificuldades:

O programa em prolog foi modelado para satisfazer as perguntas do terceiro quesito. Uma tradução literal da lógica de primeira ordem para a linguagem Prolog parece inviável para as afirmações de 1 a 6, seguindo nossas tentativas de representar o ∃!. Além disso, precisamos tomar liberdades para definir a mocinha e a invejosa para garantir que as regras the_girl e jealous_girl sejam definidas.

Somente após tomar essas liberdades o programa foi capaz de identificar como cada objeto se relaciona aos predicados H(x é o mocinho), G(x é a mocinha), B(x é o bandido), I(x é a invejosa) de forma que respondesse o terceiro quesito.