

Universidade do Minho

Escola de Engenharia

Departamento de Informática

Grupo 10

Diogo Ribeiro, a84442

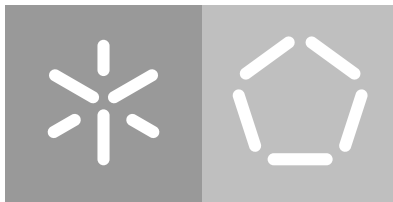
José Monteiro, a83638

Rui Mendes, a83712

Rui Reis, a84930

**Conhecimento e Raciocínio no Universo
da Contratação Pública**

Maio 2020



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Grupo 10

Diogo Ribeiro, a84442

José Monteiro, a83638

Rui Mendes, a83712

Rui Reis, a84930

Conhecimento e Raciocínio no Universo da Contratação Pública

Sistemas de Representação de Conhecimento e Raciocínio

Mestrado Integrado em Engenharia Informática

Maio 2020

CONTEÚDO

1	INTRODUÇÃO & CONTEXTUALIZAÇÃO	1
2	PRELIMINARES	2
2.1	Definições Chave	2
2.2	Universo de Atuação	2
2.2.1	Crime	3
2.2.2	Incapacidade	3
2.2.3	Administrador	4
2.2.4	Fiscal	4
2.2.5	Adjudicante & Adjudicatário	4
2.2.6	Sub-Empresa	4
2.2.7	Contrato	5
2.3	Implicações Práticas	6
2.3.1	Pessoas	7
2.3.2	Empresas	7
2.3.3	Crime	7
2.3.4	Incapacidade	8
2.3.5	Administrador	8
2.3.6	Fiscal	8
2.3.7	Fiscaliza	9
2.3.8	Adjudicante & Adjudicatário	9
2.3.9	Sub-Empresa	9
2.3.10	Contrato	9
3	IMPLEMENTAÇÃO EM PROLOG	11
3.1	Utensílios Base	11
3.2	Base das Entidades	12
3.3	Operações Básicas	12
3.4	Definição de Invariantes	12
3.4.1	Remoção de Pessoas	13
3.4.2	Validez Contratual	13
3.4.3	Histórico de Fiscais	14
3.4.4	Requisitos de Ajuste Direto	14
3.4.5	Regras dos 3 anos	15
3.5	Conhecimento Imperfeito	15

3.5.1	Conhecimento Incerto	16
3.5.2	Conhecimento Impreciso	16
3.5.3	Conhecimento Interdito	16
4	TESTES REALIZADOS	18
4.1	Inserção de Conteúdo	18
4.1.1	Regras do 3 anos	18
4.1.2	Unicidade das Entidades	19
4.1.3	Interação de Sub-Empresas	20
4.1.4	Pessoas em Contratos	21
4.1.5	Administração Empresarial	21
4.1.6	Relevância Criminal	22
4.1.7	Limitação dos Contratos	22
4.2	Remoção de Conteúdo	23
4.2.1	Referência de Pessoas	23
4.2.2	Referência de Contrato	23
4.2.3	Constituição da Administração	24
4.3	Conhecimento Imperfeito	25
4.3.1	Conhecimento Incerto	25
4.3.2	Conhecimento Impreciso	25
4.3.3	Conhecimento Interdito	26
5	CONCLUSÃO	27
A	SCRIPT DE POVOAMENTO	28

INTRODUÇÃO & CONTEXTUALIZAÇÃO

Em qualquer ramo do direito, o jurista, por natureza, tem de ser capaz de desenvolver uma análise e articulação em torno de vários diplomas legais. Isto porque, em termos da lei, é a todo o momento fundamental ter em consideração uma panóplia de diferentes mecanismos que tendem a se sobrepor entre si. Quer isto dizer que, ao aplicarmos duas leis distintas poderemos verificar uma sobreposição e restrição de domínios de atuação destas.

Desta forma, depreende-se que o bom saber da análise de sobreposição de diplomas é imprescindível, quer em modos de funcionamento jurídico ou na ótica individual do jurista.

Posto isto, é imediato que qualquer tipo de mecanismo que possa auxiliar o jurista nesta articulação é imperioso, tanto do ponto de visto económico como jurídico.

Como tal, o presente projecto visa desenvolver um sistema de representação de conhecimento e raciocínio capaz de raciocinar em torno de universo de regido pelo Código de Contratos Públicos (CCP) e todas as entidades em seu redor.

Através da implementação deste tipo de tecnologia, a um ramo do Direito Administrativo, pretende-se demonstrar as suas capacidades e vantagem perante um domínio labiríntico.

Ademais, inserimos camadas de complexidade adicionais, para além daquelas referidas no enunciado, permitindo assim conceber um sistema representativo da realidade.

A nossa abordagem foi feita de forma incremental, primeiro assentando na nossa base de conhecimento definições chave do universo de atuação considerado, seguido de uma especificação detalhada de todas as componentes envolvidas, terminando na sua implementação em PROLOG.

Por fim, realizamos testes adequados e cruciais ao confronto das capacidades do sistema nas suas aplicações reais. Através da criação de diferentes cenários e situações pontuais, somos capazes de verificar, em termos práticos, as capacidades aqui mencionadas.

PRELIMINARES

2.1 DEFINIÇÕES CHAVE

O presente projecto visa somente garantir a funcionalidade do universo da contratação pública de forma fechada. Isto é, só entram em vigor os segmentos mais atómicos presentes no CCP.

Distinguem-se essencialmente 3 tipos de entidades, que são consideradas nesta peça de software. Cujas definições são as seguintes.

- **Adjudicante:** Entidade pública que pretende ver um serviço garantido. Por exemplo, «A **Câmara de Braga**(CB) emite um concurso público para a prestação de transportes públicos na cidade», neste exemplo, a **CB** corresponde à entidade adjudicante.
- **Adjudicatário:** Entidade, pública ou privada, que responde à prestação de serviços emitida por uma entidade adjudicante. Por exemplo, «Os **Transportes Urbanos de Braga**(TUB) concorrem e vencem o respectivo concurso para o fornecimento de transportes públicos à cidade» , neste exemplo, a **TUB** corresponde à entidade adjudicatária, respondendo a um serviço requerido pela **CB**.
- **Contrato Público:** Entidade que permite a celebração de acordos entre adjudicantes e adjudicatários. Fornecendo assim um mecanismo de obtenção de bens ou serviços por parte da administração pública.

Tanto os adjudicantes como os adjudicatários são univocamente identificados pelo seu Número de Identificação Fiscal (NIF).

Assume-se, por simplicidade, que o tipo de adjudicante e adjudicatário é irrelevante. Sendo por isso ignorados artigos do CCP referentes à tipicidade destas entidades.

2.2 UNIVERSO DE ATUAÇÃO

Primeiramente, são consideradas como entidades tanto empresas como pessoas individuais. Desta forma, empresas e pessoas são os atores deste sistema, que comunicam por meio de

contratos, maioritariamente. Apesar de que, as pessoas individuais apenas podem estar envolvidas num máximo de 3 contratos anualmente.

Estes atores, partilham entre si todos os atributos. Nomeadamente, ambos possuem um número de identificação fiscal, nome e morada. E ambos são univocamente identificados pelo seu NIF, o que implica obrigatoriamente que, quer se trate de uma empresa ou pessoa, não deve existir mais do que um ator identificado pelo mesmo NIF.

Como não podia deixar de ser, estes atores interagem com o mundo à sua volta, originando predicados e condições de interesse do ponto de vista lógica. Quer isto dizer que, tanto empresas e pessoas podem assumir outros cargos ou estados ou longo da sua longevidade.

2.2.1 *Crime*

Considera-se, neste sistema, a existência de crimes que tanto podem ser cometidos por empresas, como por pessoas. A cada crime está associada um data de início, a data aquando do decreto da pena, e a respectiva data referente ao levantamento da pena.

Com este mecanismo, conseguimos sancionar todos os atores que não agirem com a conformidade da lei, o que é de todo o interesse na ótica dos juristas envolvidos na contratação pública.

É tido em conta que um criminoso pode ser acusado de vários crimes de uma só vez, ou seja, com janelas temporais sobrepostas.

2.2.2 *Incapacidade*

Naturalmente, existem pessoas que devido à sua condição estão inaptas a participarem em contratos, de qualquer tipo, e profissões de risco. Quer seja por elevada deficiência cognitiva ou por sanção regulamentar, é necessário precaver o sistema contra estes casos de impossibilidade da validade de um contrato.

Àqueles que são portadores de deficiência cognitiva são considerados inabilitados. Sendo que ao serem inabilitados estão barrados a compromissos contratuais para sempre.

Para, por exemplo, viciados em apostas, é emitida uma interdição o que torna a pessoa numa interdita. A interdição vem acompanhada de uma data de início e uma data de fim, sendo que durante este período o incorrente é isento de compromissos contratuais.

Porém, a interdição só pode ser emitida uma vez por intervalo de tempo. Ou seja, não existe sobreposição de interdições, ao contrário do que acontece com crimes. Quando uma pessoa é considerada interdita num dado espaço de tempo, só pode ser novamente interditada após esse período.

2.2.3 *Administrador*

Associado a cada empresa estão sempre figuras com mais responsabilidade dentro do órgão. Nomeadamente, a figura de administrador surge sempre que referenciamos qualquer empresa. Desta forma, os administradores são tidos em conta neste sistema. Sendo que o grau de operabilidade de uma empresa pode ser restringido pelo teor dos seus administradores.

Assume-se que cada administrador, no máximo, só pode administrar duas empresas simultaneamente. Considera-se também que um administrador não é inabilitado, apesar de poder estar, ou ter estado, interdito.

Devendo sempre haver a garantia de que pelo um administrador está inserido na base de conhecimento, associado a cada empresa.

2.2.4 *Fiscal*

Cada contrato que é produzido, especialmente no âmbito da contratação pública, deve ser escrutinado com muito rigor e desdém. Daí que surge, naturalmente, a posição de fiscal, que representa uma entidade com o objetivo de fazer a fiscalização de um contrato e de todas as suas componentes.

Um fiscal é considerado uma pessoa de elevada confiança, de tal modo que deve possuir um percurso limpo. Nomeadamente, nenhum fiscal pode ser inabilitado, ter sido alvo de uma interdição ou ter sido condenado por crime. Garantindo assim a máxima reputação desta entidade.

Um fiscal é associado a um contrato, numa dada data, que deve ser posterior ao estabelecimento do contrato. Ademais, por ser uma profissão de elevado rigor e profissionalismo, considera-se que um fiscal não pode fiscalizar mais de 2 contratos anualmente.

2.2.5 *Adjudicante & Adjudicatário*

Neste contexto, considera-se que qualquer ator, quer se trate de empresa ou pessoa, pode ser tanto um adjudicante como um adjudicatário de um qualquer contrato. Excetuando unicamente que nenhum ator pode fechar um contrato consigo próprio.

2.2.6 *Sub-Empresa*

É tido em conta que uma empresa pode ser detentora de várias outras empresas, e que pode também vir a ser posse de uma outra empresa. Este modelo é utilizado de forma a conseguir avaliar uma empresa, não só pelo seu valor e conteúdo direto, mas também por todos os comportamentos e atos de empresas da mesma família.

Considera-se que uma empresa apenas pode ser sub-empresa de uma única empresa, apesar de poder ter várias sub-empresas ela própria. Originando assim uma árvore de empresas, referenciada como a família da empresa.

2.2.7 *Contrato*

Sendo esta a entidade mais referenciada e com mais foco em todo o sistema, é também natural que possua um maior número de características e restrições.

A cada contrato encontramos associados um adjudicatário e um adjudicante, que concordam entre si sobre um acordo com uma determinada tipicidade com um certo valor. Este acordo é materializado na forma de um contrato, considerando também que um contrato específico pode vir a ser alterado, pelo que há um armazenamento de editais contratuais na forma de versões.

Em termos legais o estabelecimento de um contrato é alvo de um elevado escrutínio. Pelo que os termos de acesso e aceitação são muito restritos. Assume-se que nenhum ator pode fechar um contrato consigo próprio, o que seria muito provavelmente apenas aproveitamento.

Para além disso, os atores envolvidos não podem estar a cumprir pena criminal, ou, no caso de pessoas, estarem inabilitadas ou interditas, naquele período.

Como se trata de um procedimento rigoroso, toda a família da empresa, caso de uma empresa se trate, é verificada de forma a garantir que o contrato não pode ser formado se algum membro da família estiver a cumprir pena, evitando assim a propagação de metodologias fraudulentas. Tal como também só são aceites contratos vindos de empresas fiáveis, o que implica que contratos com uma administração criminosa não são aceites, independente de ter ou não acabado a pena.

Ademais, deve ser garantido que o fiscal atribuído a um contrato não está diretamente envolvido neste, nem pode ser administrador de nenhuma empresa das famílias de empresas envolvidas.

Deste modo, considera-se um contrato regido pelos seguintes termos. Especificamente, um contrato pode tomar qualquer um dos seguintes tipos:

- Aquisição de bens móveis.
- Locação de bens móveis.
- Aquisição de serviços.
- Empreitadas de obras públicas.

Sendo que cada um destes contratos deve ser tomados sobre um dos seguintes procedimentos, cada um com as suas restrições:

- **Ajuste direto:**

- Valor de contrato igual ou inferior a 5.000 euros.
- Só válido para contratos de aquisição ou locação de bens móveis ou aquisição de serviços.
- Prazo de vigência até 1 ano, inclusive, a contar da decisão de adjudicação.

- **Consulta prévia:**

- Valor de contrato igual ou inferior a 100.000 euros.
- Só válido a contratos de empreitadas de obras públicas.
- Só é permitido o envolvimento de empresas, pelo que pessoas não podem aceder diretamente.
- Prazo de vigência até 2 anos, inclusive, a contar da decisão de adjudicação.

- **Concurso público:**

- Valor de contrato igual ou inferior a 200.000 euros.
- Válido a todo o tipo de contratos.
- Prazo de vigência até 5 anos, inclusive, a contar da decisão de adjudicação.

A qualquer contrato aplica-se a **regra dos 3 anos** que implica que uma entidade adjudicante não pode convidar a mesma empresa para celebrar um contrato com prestações do mesmo tipo ou idênticas às de contratos que já lhe foram atribuídos, no ano económico em curso e nos dois anos económicos anteriores, sempre que o preço contratual acumulado dos contratos já celebrados (não incluindo o contrato que se pretende celebrar) seja igual ou superior a 75.000 euros.

Ademais, em resposta à lei de protecção de dados, surgiu uma variante de contratos que nos proporcionam um mecanismo de sigilo contratual. Nomeadamente, são considerados contratos confidenciais, sendo que estes contratos permitem a um adjudicante especificar um contrato que realizou com uma entidade adjudicatária desconhecida, e que nunca se poderá vir a saber. Desta forma, garantimos que cada entidade pode manter em sigilo todos os seus sócios de negócios. Porém, não pode ofuscar os valores contratuais.

2.3 IMPLICAÇÕES PRÁTICAS

De forma a fazer um tipo de organização estrutural, decidimos dividir a explicitação do universo de atuação das suas implicações em termos práticos.

Partindo desse ponto, a presente secção tem o objetivo de definir as regras que devem ser embutidas no sistema de forma a circunscrever todas as dinâmicas associadas às diferentes

entidades. Assim, ficamos com um resumo da explicitação do universo, mais apto a ser diretamente passado a PROLOG.

2.3.1 *Pessoas*

Uma das entidades base deste ecossistema, sobre a qual são aplicadas diversas mecânicas. Por esta razão é necessária uma atenção especial a esta entidade. Especificamente, conseguimos definir as seguintes regras, que devem ser garantidas para o funcionamento correto desta entidade.

1. Só pode existir um Número de Identificação Fiscal (NIF) associado a cada entidade, sendo que este é partilhado com as empresas.
2. Não se pode remover uma pessoa do sistema caso esta esteja envolvida com alguma outra componente do sistema. *i.e.* uma pessoa não pode ser removida caso seja administradora de alguma empresa.
3. Pessoas apenas podem estar envolvidas num máximo de 3 contratos anualmente.

2.3.2 *Empresas*

Também com tanta, ou mais, relevância que a entidade acima referida é necessário dotar esta entidades de determinadas regras para o seu bom funcionamento e posteriormente estabelecimento de predicados sobre este.

1. Só pode existir um NIF associado a cada entidade, sendo que este é partilhado com as pessoas.
2. Não se pode remover uma empresa do sistema caso esta esteja envolvida com alguma outra componente do sistema. *i.e.* uma empresa não pode ser removida caso seja sub-empresa de outra.

2.3.3 *Crime*

Componente transversal às entidades base do sistema, pelo que tanto uma empresa como uma pessoa podem ser sujeitas a cumprir uma pena.

1. A entidade base acusada de crime deve existir. *i.e.* impossível condenar empresas inexistentes.
2. A data de fim de pena deve ser superior à data de início da respectiva pena.

3. Entidades a cumprir pena, não podem estar envolvidas em contratos.

2.3.4 *Incapacidade*

O universo de atuação tem especial atenção pelas pessoas em situações mais frágeis. Nomeadamente, separa-se o conjunto de pessoas incapacitadas em pessoas interditas e pessoas inabilitadas. Cada uma com as suas respectivas condições.

Interdito

1. A ordem de interdição deve ser emitida para uma pessoa existente.
2. A data de fim de interdição deve ser superior à data de início da respectiva interdição.
3. Pessoas a cumprir interdição não podem estar envolvidas em contratos.
4. Não podem ser emitidas interdições com intervalos sobrepostos para a mesma pessoa.

Inabilitado

1. A pessoa inabilitada deve existir, e só pode ser executada uma ordem de inabilitação por pessoa.

2.3.5 *Administrador*

Os administradores são peças importantes do sistema, e representam os órgãos humanos que representa a empresa e os seus valores. Razão pela qual ficam sujeitas a condições especiais.

1. O administrador deve ser uma pessoa válida, e empresa que este administra deve também ser válida.
2. Um administrador só pode administrar no máximo 2 empresas simultaneamente.
3. Administrador não pode ser inabilitado.
4. Não podem ser removido um administrador caso este seja o único administrador da empresa.

2.3.6 *Fiscal*

O fiscal é tido como uma pessoa de elevada capacidade, com regras e condições de acesso, à posição, rigorosas.

1. O fiscal indicado deve ser uma pessoa válida.
2. Fiscal não poder ser inabilitado, ter cadastrado criminal ou alguma vez ter sido interdito.

2.3.7 *Fiscaliza*

Fiscaliza concretiza atuação física de um fiscal perante o ecossistema. Um fiscal fiscaliza contratos numa determinada data, pelo que este ato deve ser delimitado.

1. O fiscal envolvido na fiscalização deve ser válido.
2. O contrato envolvido deve ser válido, e a data de fiscalização deve ser seguinte à data em que o contrato foi estabelecido.
3. Fiscal pode estar envolvido na fiscalização de no máximo 2 contratos anualmente.
4. Só pode haver um fiscal responsável por cada contrato.
5. O fiscal atribuído a um contrato não pode estar diretamente envolvido neste, nem pode ser administrador das empresas envolvidas, ou respectivas famílias de empresas.

2.3.8 *Adjudicante & Adjudicatário*

Relativamente a adjudicantes e adjudicatários, não é verdadeiramente necessária a concretização destas entidades no sistema. Visto que, através das entidades pessoas e empresas, temos todas as necessárias ferramentas para trabalhar com as entidades envolvidas em contratos.

2.3.9 *Sub-Empresa*

Uma empresa pode naturalmente ser uma sub-empresa de uma outra empresa, unicamente se forem cumpridos critérios válidos sobre estas.

1. Empresa mãe e empresa filha devem existir, e não podem ser a mesma.
2. Empresa filha não pode ser mãe da empresa mãe.
3. Cada empresa só pode ter uma empresa mãe.

2.3.10 *Contrato*

Corresponde à componente de maior importância no sistema, pelo que necessariamente tem uma maior agregação de regras.

1. ID do contrato não pode ser repetido.
2. Adjudicantes e adjudicatários devem existir.
3. Tipo de contrato e tipo de procedimento deve ser válido.
4. Não pode ser removido um contrato se este estiver em fiscalização.
5. Não se pode adicionar informação confidencial sobre o adjudicatário de um contrato confidencial pré-existente.
6. Nenhum ator pode fechar um contrato consigo próprio.
7. Atores do contrato não podem estar a cumprir pena, interditos ou inabilitados à data do fecho do contrato.
8. Nenhum membro da família de empresas envolvidas pode estar a cumprir pena.
9. Empresas envolvidas com administrações com registo criminal não podem estar envolvidas em contratos.
10. As especificações de cada tipo de procedimento contratual devem ser sempre respeitadas.
11. Regras dos 3 anos, mencionada acima.

IMPLEMENTAÇÃO EM PROLOG

De forma a descentralizar o conteúdo visado no sistema, procedeu-se à separação do sistema em diversos ficheiros, sendo que cada ficheiro é relativo a cada tipo de funcionalidade. Designadamente, os ficheiros foram separados da seguinte forma.

- `utils.pl`: Contém predicados auxiliares, no domínio de extensão à linguagem de PROLOG.
- `base.pl`: Define dinamicamente os predicados dedicados a todas as entidades do sistema, bem como a sua respectiva extensão ao Pressuposto do Mundo Fechado.
- `operacoes.pl`: Explicita predicados, nomeadamente *getters*, *inserts* e *removes* associados a cada uma das entidades, utilizados maioritariamente com o intuito de *debugging*.
- `invariantes.pl`: Define todas as específicas relações entre todas as diversas entidades, explicitando, assim, tanto invariantes referências como estruturais.
- `povoamento.pl`: Ficheiro de povoamento inicial da base de conhecimento, incluindo conhecimento perfeito e imperfeito.
- `ex1.pl`: Ficheiro encarregue pela gestão e funcionamento dos outros ficheiros de código auxiliares, responsável pelo *include* adequado de cada um dos ficheiros.

3.1 UTENSÍLIOS BASE

De forma a dotar o sistema com todas as ferramentas necessárias à aplicação de todos os predicados desejados, construímos o sistema sobre um conjunto de predicados relativos à inserção, remoção de conhecimento, e programação em lógica estendida. Nomeadamente, através dos seguintes predicados:

- `evolucao`, inserção de conhecimento.
- `involucao`, remoção de conhecimento.

- `demo`, para programação em lógica estendida.

Para além dos demais, foram implantados outros predicados auxiliares, todos especificados no ficheiro `utils.pl`.

3.2 BASE DAS ENTIDADES

Com o objetivo de definir uma base funcional para o contínuo desenvolvimento e aplicação dos diferentes predicados, foi criado um ficheiro base, que visa alocar dinamicamente todos os predicados, para posteriormente poder ser feita a evolução de conhecimento destes. Através da definição do seguinte, para cada entidade:

```
:- dynamic(empresa/3).
```

Ademais, é também extendido o Pressuposto do Mundo Fechado (PMF), a cada um dos predicados.

```
-empresa( N,No,M ) :-
    nao(empresa( N,No,M )),
    nao(excecao(empresa( N,No,M ))).
```

3.3 OPERAÇÕES BÁSICAS

Para mais facilmente conseguirmos trabalhar com a base de conhecimento, desenvolvemos operações básicas que achamos de interesse para proceder ao *debugging*, Nomeadamente:

- Operações de acesso especializado à base de conhecimento, i.e. *getters*.
- Operações para inserção de remoção de conhecimento. i.e. *wrappers* dos predicados `evolucao` e `involucao`.

Neste contexto, são também definidos predicado necessários à inserção e remoção de contratos confidenciais, temática do conhecimento imperfeito subjacente ao sistema em questão.

3.4 DEFINIÇÃO DE INVARIANTES

No respectivo ficheiro encontram-se definidos todos os invariantes necessários à completude do sistema, separados em invariantes referências e estruturais. Mencionando invariantes referências

como medidas que permitem a garantia da integridade do conhecimento e estruturais os invariantes que garantem o cumprimento das normas subjacentes ao sistema.

A título de exemplo, apresenta-se de seguida um invariante referencial seguido de um estrutural. De forma a verificar de que forma foi obtida esta completude.

3.4.1 *Remoção de Pessoas*

Pretende-se que a remoção de uma pessoa seja impossível caso esta esteja envolvida em algum contrato, crime, fiscalização, empresa, etc.. Garantido assim que não é referenciada nenhuma pessoa, sem que esta esteja inserida no sistema. Isto pode ser obtido com o seguinte invariante, representativo de conhecimento negativo. Por outras palavras, o invariante abaixo garante que uma pessoa só pode ser removida da base de conhecimento, caso o seu NIF não seja mencionado em nenhuma outra entidade.

```
-pessoa(NIF,_,_):(
  solucoes(NIF,
    (crime(NIF,_,_);
     interdito(NIF,_,_);
     inabilitado(NIF);
     administrador(NIF,_);
     contrato(_,_,NIF,_,_,_,_,_,_,_);
     contrato(_,_,_,NIF,_,_,_,_,_,_);
     fiscal(NIF)), R),
  comprimento(R,0)).
```

3.4.2 *Validade Contratual*

É também necessário garantir que todos os contratos possuem tipos de contratos e procedimentos válidos. O que pode ser dito, por outras palavras, pela garantia de que há 0 contratos cujos tipos não sejam aqueles predefinidos. Como se segue.

```
+contrato(_,_,_,_,_,_,_,_,_,_):(
  solucoes(ID,
    ( contrato(ID,_,_,_,TC,TP,_,_,_,_,_) ,
      (nao(membro(TC,
        ['aquisicao de bens moveis',
         'locacao de bens moveis',
         'aquisicao de servicos',
         'empreitadas de obras publicas']))) ;
      nao(membro(TP,
        ['ajuste direto',
```

```

        'consulta previa',
        'concurso publico'])) ) ), R),
    comprimento(R,0)).

```

3.4.3 *Histórico de Fiscais*

Em termos estruturais, como discutido anteriormente, todos os fiscais devem ser entidades com um histórico limpo, em todos os aspetos. Logo, esta garantia pode ser dada garantindo que o número de fiscais que possuem características indesejadas é 0.

```

+fiscal(_):(
    solucoes(NIF,
        (fiscal(NIF),
        (inabilitado(NIF);
        interdito(NIF,_,_) ;
        crime(NIF,_,_))), R),
    comprimento(R,0)).

```

3.4.4 *Requisitos de Ajuste Direto*

Ademais, deve também ser garantido que todos os contratos do tipos ajuste direto respeitam as indicações acima. O que se obtém garantido um número nulo de contratos que não cumprem esses requisitos, por meio de:

```

+contrato(_____,_):(
    solucoes(ID,
        ( contrato(ID,_____,Tc,'ajuste direto',_,Val,Pr,_,Dt),
        ((Val < 0 ; Val > 5000) ;
        (Tc == 'empreitadas de obras publicas') ;
        (Pr < 0 ; Pr > 365)) ), R),
    comprimento(R,0)).

```

3.4.5 Regras dos 3 anos

Uma das regras mais complexas de integrar no sistema foi a regra dos 3 anos, explicitada acima. Isto porque envolve, não só a condição do contrato atual, mas sim todo o histórico de contrato desenvolvido por àquelas entidades. Desta forma, concretizamos este invariantes como os demais implementados acima, obrigando o sistema a bloquear tentativas de inserções que violem esta regra.

```
sumAux([],0).
sumAux([(_,V)|T], R) :-
    sumAux(T,R1), R is R1 + V.

+contrato(_____,_):-(
    solucoes(ID,
        (contrato(ID,_,NIF1,NIF2,Tc,Tp,_,Vi,_,_,data(Y,_,_)),
        Yo is Y - 2,
        empresa(NIF2,_,_),
        solucoes((ID2,Val),
            (contrato(ID2,_,NIF1,NIF2,Tc,Tp,_,Val,_,_,Dt2) ,
            dataEm(Dt2,Yo,Y) ), R1),
        sumAux(R1,Vt),
        VF is Vt - Vi,
        VF >= 75000) , R2),
    comprimento(R2,0)).
```

3.5 CONHECIMENTO IMPERFEITO

Como qualquer outro sistema, é necessário que o conhecimento imperfeito, em todas as suas formas, seja concretamente confrontado.

Partindo disto, estabelecemos todos os diferentes tipos de conhecimento imperfeito. Nomeadamente,

- Conhecimento Incerto.
- Conhecimento Impreciso.
- Conhecimento Interdito.

De forma a dotar o sistema com mecanismos de conhecimento imperfeito, foram definidos, manualmente, casos precisos em que este tipo de conhecimento se verifica. De forma a conseguir dotar e comprovar a inserção deste tipo de funcionalidades.

Daí que se ilustram de seguida caso específicos em que foi aplicado o conhecimento imperfeito.

3.5.1 *Conhecimento Incerto*

O conhecimento incerto diz respeito a conhecimento sobre o qual está na dúvida, ou simplesmente não temos acesso.

Por exemplo, de forma a inserir uma pessoa, cuja morada é desconhecida, basta adicionarmos o seguinte excerto.

```

pessoa(015,'Alberto Dias',morada00).
execcao(pessoa(N,Na,Mo)) :-
    pessoa(N,Na,morada00).

```

Isto implica que, qualquer valor de morada, para aquela pessoa, é desconhecido. Pois simplesmente não temos acesso a este dado, seja por que razão for.

3.5.2 *Conhecimento Impreciso*

Ao nível de conhecimento impreciso, adicionamos no nosso sistema uma entidade, da qual não sabemos o nome, simplesmente sabemos que o nome dela se encontra algures, alfabeticamente, entre o nome 'Filipe' e 'Paulo'. O que pode ser facilmente modelado por:

```

execcao( pessoa(019,Nome01,'Barcelos') ) :-
    Nome01 @>= 'Filipe', Nome01 @=< 'Paulo'.

```

O que garante, por exemplo, 'Diogo' não é, de certeza, o nome da pessoa indicada.

3.5.3 *Conhecimento Interdito*

Num cenário caótico, imagina-se a possibilidade de cumprir penas perpétuas. Desta forma, a data de fim de uma pena é desconhecida e nunca poderá vir a ser conhecida, devido a se tratar de uma sentença sem prazo limite.

Por essa razão, através da seguinte condição, somos capazes de garantir que nunca poderá vir a ser adicionada uma data de fim a uma pena perpétua.

```

crime(123,data(2200,10,10),data(y1,m1,d1)).
execcao(crime(N,Di,Df)) :-
    crime(N,Di, data(y1,m1,d1)).

nulo(y1).
nulo(m1).
nulo(d1).

```

```
+crime(_,_,_):(
  solucoes((Y,M,D),
    (crime(123,_,data(Y,M,D)),
      nao(nulo(Y)),
      nao(nulo(M)),
      nao(nulo(D)) ), R),
  comprimento(R,0)).
```

Garantindo sempre, que uma data de inicio de pena nunca pode ser menor que a data de fim, segundo o conhecimento negativo.

```
-crime(_,data(Y1,_,_),data(Y2,_,_)) :-
  Y1 > Y2.
```

TESTES REALIZADOS

De forma a conseguir realizar testes adequados à *performance* do sistema como um todo é necessário primeiro popular a base de conhecimento com entidades suficientemente dispersas. Que consigam com alguma precisão identificar o panorama real do enorme universo que aqui se debate. Deste modo, para conduzir os testes, teve-se sempre em conta o *script* de povoamento fornecido pelo anexo A.

Devido ao vasto leque de diferentes regras que foram dotadas ao sistema, acima explicitadas, pretende-se unicamente, e a título exemplificativo, mostrar o funcionamento prático de regras que achamos cruciais ao sistema. Especialmente, pretende-se retratar as capacidades do sistema em cada um dos seguintes ramos de atuação.

- Inserção de Conteúdo.
- Remoção de Conteúdo.
- Conhecimento Imperfeito.

Olhando, cirurgicamente, para cada um destes ramos, com a finalidade de raciocinar sobre as implicações que estas regras impõem sobre o sistema, e debate sobre a sua necessidade para a manutenção da completude referencial e estrutural do sistema.

4.1 INSERÇÃO DE CONTEÚDO

De forma a testar o funcionamento da aplicação das regras no âmbito da inserção de conteúdo, foram desenvolvidos os seguintes testes exemplos.

4.1.1 Regras do 3 anos

Como explicitado acima, a regra dos 3 anos diz respeito ao historial de negócios de uma dada entidade com os constituintes do universo de atuação. Nomeadamente, uma entidade não poderá fechar contratos com uma empresa com quem tenha, nos últimos 3 anos, fechados

contratos com a mesma tipicidade com um valor superior a 75.000 €, exclusive o contrato atual.

Podemos com facilidade construir um cenário no qual podemos testar estas capacidades. Se pegarmos tentarmos repetir um contrato, entre duas mesmas empresas, com a mesma tipicidade contratual, ao longo do 3 anos com valores iguais de 30.000 €, então inevitavelmente verificaremos que a sua soma é superior a 75.000 €, pelo que a inserção de novos contratos do mesmo tipo, nesse intervalo de 3 anos, deverá ser impossível. Tal como se verifica de seguida. Assumindo X e Y como sendo tipos contratuais válidos.

```
?- addContrato(44, 1, 103, 106, X, Y, 30000, ..., data(2015,4,4)).
yes
?- addContrato(45, 1, 103, 106, X, Y, 30000, ..., data(2016,4,4)).
yes
?- addContrato(46, 1, 103, 106, X, Y, 30000, ..., data(2017,4,4)).
yes
```

Com sucesso, conseguimos inserir todos estes contratos, porém ao tentarmos inserir mais um contrato no ano de 2017, estaremos, nitidamente, a quebrar a regra dos 3 anos.

```
?- addContrato(47, 1, 103, 106, X, Y, 30000, ..., data(2017,4,4)).
no
```

Pelo que se verifica, na prática, a ocorrência da regra dos 3 anos. Porém, deverá ser possível, está mesma inserção, caso a entidade adjudicatária seja alterada.

```
?- addContrato(47, 1, 103, 107, X, Y, 30000, ..., data(2017,4,4)).
yes
```

Resultando no funcionamento pretendido desta regra basilar do sistema.

4.1.2 *Unicidade das Entidades*

A unicidade das entidades representa o seu direito a uma entidade única, de acordo com o seu Número de Identificação Fiscal. Deste modo, estes ambiente pode ser facilmente simulado tendo em conta o *script* de povoamento fornecido. Lá, tiremos partido da pessoa identificada com o NIF 001, de seguida, se tentarmos adicionar uma outra pessoa, com esse mesmo NIF, facilmente concluímos que é impossível. Por exemplo,

```
?- addPessoa(001,'Rodrigo Otavio','Rua Bonita n27').
no
```

De forma semelhante, se tentarmos criar uma empresa com este NIF, deverá ser também impossível.

```
?- addEmpresa(001,'Olaria Rodrigues','Rua Bonita n77').
no
```

Enquanto que, se tentarmos inserir esta mesma empresa com um NIF não atribuído, então isto será possível.

```
?- addEmpresa(999,'Olaria Rodrigues','Rua Bonita n77').
yes
```

Provando assim, na prática, a unicidade da identidade de cada uma das entidades base constituintes do ecossistema.

4.1.3 *Interação de Sub-Empresas*

Como explicitado anteriormente, uma empresa filha não pode ser mãe de uma das suas empresas mãe, inicialmente isto pode parecer um bocado confuso, mas vejamos o seguinte contexto. Assuma-se que temos a seguinte relação $C \leftarrow B \leftarrow A$, ou seja, C é sub-empresa de B e B é sub-empresa de A . Com este conhecimento, será intuitivo que qualquer tentativa de indicar que A é sub-empresa de C deverá ser impossível, isto porque A é uma das empresas mães de C .

Então, assumamos que A , B e C são NIFs de empresas já definidas no povoamento do sistema, então a seguinte tentativa terá um resultado nitidamente catastrófico.

```
?- addSubempresa(A, C).
no
```

Também se pretende que uma empresa possa unicamente ser filha de uma única empresa. Por isso, se consideramos o povoamento já inerente ao sistema, verificamos possuímos o seguinte predicado `subempresa(105,106)`. Daí, se tentarmos definir a empresa 105 como filha de qualquer outra, deverá ser nitidamente impossível.

```
?- addSubempresa(105, 107).
no
```


4.1.4 *Pessoas em Contratos*

Uma pessoa só pode estar envolvida num máximo de 3 contratos anualmente, para testar isto basta repetir várias vezes um mesmo contrato, alterado unicamente o respectivo ID, e à 3ª inserção, deve ser impossível proceder. Considere-se portanto o contrato com o ID 1 já inserido no povoamento. Como já há uma entrada deste mesmo contrato, onde a pessoa com o NIF 007 está envolvida, 2 mais inserções são o suficiente para impossibilitar inserções posteriores. Assume-se que todos os contratos inseridos envolvem a pessoa indicada, porém, por simplicidade, omite-se todos os parâmetros associados.

```
?- addContrato(20, ... ,data(2015,10,10)).
yes
?- addContrato(21, ... ,data(2015,10,10)).
yes
?- addContrato(22, ... ,data(2015,10,10)).
no
```

O último contrato não foi adicionado à base de conhecimento, isto porque a pessoa 007 já realizou 3 contratos naquele ano, o máximo imposto neste sistema. Daí, se tentarmos re-adicionar este contrato, mas agora numa nova data, seremos capazes de, com sucesso, inserir na base de conhecimento.

```
?- addContrato(22, ... ,data(2016,10,10)).
yes
```

4.1.5 *Administração Empresarial*

Uma administração, como qualquer outra, não deve estar sobrecarregada em termos de trabalho. Daí que nenhum administrador pode administrar mais do que 2 empresas simultaneamente. Para testar o sistema com este mecanismo, basta tentar inserir uma relação de administração mais do que 2 vezes, para um dado administrador. Daí, partindo do povoamento, verificamos que:

```
?- addAdministrador(001,100).
yes
?- addAdministrador(001,101).
yes
?- addAdministrador(001,102).
no
```

Como se verifica, a partir do momento em que inserimos mais de duas relações de administração, naturalmente, ocorre uma falha.

4.1.6 *Relevância Criminal*

Este sistema tem sempre em conta a relevância do registo criminal de tanto pessoas como empresas. Desta forma, uma empresa, ou pessoa, não pode estar envolvida em contrato no período em que esteja a cumprir pena. Pelo povoamento fornecido, verifica-se que a empresa 106 foi condenada a cumprir uma pena de 1990 a 2000. É também possível verificar que esta mesma empresa já está associada a um contrato que sucedeu em 2015. Porém, assim que tentamos inserir um contrato para a data em que este foi condenado, verificamos o seguinte.

```
?- addContrato(22, ..., data(1995,4,4)).
no
```

O que é nítido, pois, caso tivesse sucesso, implicaria que o contrato teria sido realizado numa altura em que a empresa em questão estava a cumprir pena, o que é impossível.

4.1.7 *Limitação dos Contratos*

Notoriamente, as especificações de cada um dos tipos contratuais devem ser respeitados. Será portanto impossível inserir qualquer contrato que quebre esta lógica associada. Consideremos portanto um contrato do tipo 'ajuste direto', então só será possível a sua inserção se, como enunciado acima, o seu prazo for menor que um ano. Então o seguinte, inevitavelmente, falhará.

```
?- addContrato(66, ..., 'ajuste direto', ..., 600, ...).
no
```

Isto porque, o prazo indicado é evidentemente superior a 1 ano. Porém, isto já funcionará, se este critério for cumprido. Como se segue,

```
?- addContrato(66, ..., 'ajuste direto', ..., 200, ...).
yes
```

Evidenciando assim, um tipo de critério no qual as especificações do contrato devem ser respeitadas.

4.2 REMOÇÃO DE CONTEÚDO

A remoção de conteúdo é uma quanto ou mais importante que a sua respectiva inserção, isto porque, tal como em qualquer sistema, deve ser sempre mantida a integridade referencial do sistema. Garantido que só são referenciadas, a qualquer momento, entidades existentes no sistema. Para além disso, invariantes para a remoção de conteúdo podem também servir outros propósitos.

Entidades que venham a ser inseridas, e posteriormente referenciadas, não podem vir a ser removidas antes que os predicados onde são referenciadas sejam removidos. Por absurdo, assumamos que possamos remover entidades, mesmo que estas sejam referenciadas noutros predicado. Então, poderia surgir o caso em que uma pessoa com o NIF ϕ fosse acusada de um crime. De seguida, essa mesma pessoa poderia ser removida da base de conhecimento, se, porventura, outra pessoa pudesse tomar posse desse mesmo NIF, então essa pessoa seria injustamente acusada de um crime que não cometeu.

Com o exemplo acima em conta, verifica-se a importância inquestionável de garantir a integridade referencial deste sistema.

4.2.1 *Referência de Pessoas*

Tenhamos como exemplo a pessoa identificada com o NIF 005, sobre esta pessoa foi imposta uma inabilitação, pelo que a tentativa de remover esta mesma pessoa deveria ser impossível.

```
?- remPessoa(005,'Dominic Torreto','Rua Furiosa, nº 10').
no
```

Enquanto que, se tentarmos remover uma pessoa que não está envolvida em nenhum predicado, então, inevitavelmente, a sua remoção será satisfeita. Peguemos na entidade com o NIF 011, que não está envolvida em nenhuma atividade extra, então verificamos que:

```
?- addPessoa(011,'Sr Amares','Rua do Padre').
yes
```

O que atesta a manutenção da integridade do sistema.

4.2.2 *Referência de Contrato*

De semelhante forma, a manutenção da integridade referencial dos contratos, deve também ser mantida. A única situação em que um contrato é referenciado é no predicado de fiscalização. Por isso, mais uma vez, tiremos partido do povoamento para tirar as nossas conclusões.

```
?- remContrato(1, ...).
no
```

O que é notório, visto que existe já uma fiscalização atribuída a este mesmo contrato. Porém, se removermos primeiro esta fiscalização da base de conhecimento, então já será possível remover o contrato indicado.

```
?- remFiscaliza(002, 1, data(2016,10,10)).
yes
?- remContrato(1, ...).
yes
```

Verificando assim que um contrato só pode ser removido assim que não seja referenciado em nenhum outro segmento do sistema.

4.2.3 *Constituição da Administração*

De forma a manter a integridade do tecido empresarial, cada uma das administrações deve ter sempre pelo menos um administrador. Pelo que, a remoção de um administrador, quando este é o único restante administrativo da empresa, deve ser expressamente proibido. Como se verifica, se tentarmos remover a pessoa 009, de forma a que esta deixa de ser administradora da empresa 107, vamos ser confrontados com um resultado fatídico.

```
?- remAdministrador(009, 107).
no
```

Porém, caso o quadro desta empresa venha a ser suplementado com um novo administrador, o administrador 009 poderia abandonar o seu cargo.

```
?- addAdministrador(006, 107).
yes
?- remAdministrador(009, 107).
yes
```

Como resultado, agora o administrador 006 fica vinculado à empresa, não podendo ser removido até que o quadro seja completado.

```
?- remAdministrador(006, 107).
no
```

4.3 CONHECIMENTO IMPERFEITO

A título exemplificativo, apresentam-se, de seguida, predicados onde fazemos uma apresentação e funcionalidade dos nossos tipos de conhecimento imperfeito, de forma a atestar a sua qualidade.

4.3.1 *Conhecimento Incerto*

Tomemos o exemplo 1.1) do povoamento, somos capazes de verificar a seguinte definição.

```
-pessoa(017,'Pedro','Ermesinde').
-pessoa(017,'Rodrigo','Ermesinde').
pessoa(017,nome00,'Ermesinde').
excecao(pessoa(N,Na,Mo)) :-
    pessoa(N,nome00,Mo).
```

Que garante que existe uma pessoa com o NIF 017 que vive em Ermesinde. Porém, o nome desta pessoa é desconhecido, apesar de termos a garantir que não se chama nem Pedro, nem Rodrigo. Daí que se tentarmos verificar no sistema se isto se aplica, ou não, obtemos que:

```
?- demo(pessoa(017,'Pedro','Ermesinde'), R).
R = falso ?
```

Obtemos o resultado esperado. Porém, se agora tentarmos indicar o nome dessa pessoa como sendo Rui, somos avisados que é desconhecido. Isto, porque a única garantia que há sobre a pessoa 017 é a de que, de certeza, não possui nenhum dos nomes indicados acima.

```
?- demo(pessoa(017,'Rui','Ermesinde'), R).
R = desconhecido ?
```

4.3.2 *Conhecimento Impreciso*

Tomando o exemplo 1.4), somos capazes de verificar que:

```
excecao( pessoa(018,'Rui','Madeira') ).
excecao( pessoa(018,'Rui','Lisboa') ).
```

Ou seja, não sabemos onde mora o Rui, mas sabemos que mora na Madeira ou em Lisboa, pelo que qualquer tentativa de indicar o contrário deve ser, nitidamente, falsa.

```
?- demo(pessoa(018,'Rui','Madeira'), R).  
R = desconhecido ?
```

```
?- demo(pessoa(018,'Rui','Ermesinde'), R).  
R = falso ?
```

4.3.3 *Conhecimento Interdito*

Considerando o exemplo de conhecimento interdito, apresentado no capítulo de preliminares, somos capazes de verificar que qualquer tentativa de especificar a data final da pena deverá ser impossível, assegurando o invariante de que a perpétua possui uma data de terminação desconhecida.

```
?- demo(crime(123,data(2200,10,10),data(2300,10,10)),R).  
R = desconhecido ?
```

CONCLUSÃO

Através de diversos testes conseguimos verificar que este sistema é capaz de contemplar um subconjunto do universo da contratação pública, permitindo um raciocínio, apesar de primitivo, acertado sobre os termos considerados no presente documento.

Ademais, pondera-se que este sistema possa facilmente ser estendido a uma maior parcela do sua área da atuação com facilidade. Tanto na especificação adicional de entidades, ou imposição de regras extra.

Concluimos desta forma que a linguagem PROLOG, e todas as extensões fornecidas para tal, permitem com rigor e sucesso descrever o mecanismo pretendido. Possibilitando uma especificidade extremamente descritiva de um universo com uma população detalhada com fortes dinâmicas entre entidades.



SCRIPT DE POVOAMENTO

```
% #####  
2 % ## SRCR - Dominio de Contratacao Publica. ##  
% ## Desenvolvido pelo grupo de trabalho n 10. ##  
4 % ## Ano letivo 2019/20 - MIEI. ##  
% ## ##  
6 % ## ----- ##  
% ## ##  
8 % ## Ficheiro: povoamento.pl ##  
% ## Descricao: O presente ficheiro tem o objetivo de ##  
10 % ## definir fazer um povoamento inicial da base de co- ##  
% ## nhecimento para esta, e respectivos predicado, po- ##  
12 % ## derem posteriormente ser testados. Fazendo a inser- ##  
% ## cao de conhecimento perfeito e imperfeito adequado ##  
14 % ## ao seu respetivo teste. ##  
% ## ##  
16 % ## NIFS 000 -> 099 : pessoas / 100 -> 199 : empresas ##  
% #####  
18  
% #####  
20 % Povoamento de conhecimento perfeito.  
  
22 % 1) Pessoa.  
  
24 ?- addPessoa(001,'Roberto Dias','Rua Calma, n 1').  
?- addPessoa(002,'Adolfo Monteiro','Rua Calma, n 2').  
26 ?- addPessoa(003,'Rudolfo Monteiro','Rua Calma, n 3').  
?- addPessoa(004,'Rudolfo Dias','Rua Calma, n 4').  
28 ?- addPessoa(005,'Dominic Torreto','Rua Furiosa, n 10').  
?- addPessoa(006,'Pastor Dias','Rua Curiosa, n 5').  
30 ?- addPessoa(007,'Luke Skywalker','Tatooine, n 301').  
?- addPessoa(008,'Chuck Silva','Rua do Chuck, n 000').  
32 ?- addPessoa(009,'Manuel Moreira','Travessa de Cima, n 101').  
?- addPessoa(010,'Ric Fazeres','Rua do Travesseiro Crocante, n 55').  
34 ?- addPessoa(011,'Sr Amares','Rua do Padre').
```



```

36 % 2) Empresa.

38 ?- addEmpresa(100,'Parafusos, lda','Rua Calma, n 1').
?- addEmpresa(101,'Pregos, lda','Rua Calma, n 2').
40 ?- addEmpresa(102,'Tabuas, lda','Rua Calma, n 3').
?- addEmpresa(103,'Areias, lda','Rua Calma, n 4').
42 ?- addEmpresa(104,'Rochas, lda','Rua Calma, n 5').
?- addEmpresa(105,'Zeca Panados', 'Bairro de Santa tecla, n6').
44 ?- addEmpresa(106,'PoBranco, lda', 'Avenida de Amarante, n7').
?- addEmpresa(107,'Maria Bolacha, lda', 'Zona Industrial Madeira, n34').
46 ?- addEmpresa(108,'Kebab Turco', 'Avenida Principal, n14').
?- addEmpresa(109,'Seguros, lda', 'Rua do Padre, n34').
48

% 3) Crime.
50

% ---- 1) Pessoas.
52 ?- addCrime(007, data(3000,10,10), data(5000,10,10)).
?- addCrime(010, data(2011,07,23), data(2021,08,02)).
54

% ---- 2) Empresas.
56 ?- addCrime(106, data(1990,12,03), data(2000,12,03)).
?- addCrime(109, data(2008,10,02), data(2015,11,25)).
58

% 4) Interdito.
60

?- addInterdito(001, data(1990,10,10), data(2010,10,10)).
62 ?- addInterdito(008, data(2015,08,15), data(2025,09,20)).
?- addInterdito(006, data(2020,09,20), data(2030,10,17)).
64

% 5) Inabilitado.
66

?- addInabilitado(005).
68

% 6) Administrador.
70

?- addAdministrador(010,104).
72 ?- addAdministrador(009,107).
?- addAdministrador(004,108).
74

% 7) Fiscal.
76

?- addFiscal(002).
78 ?- addFiscal(003).

80 % 8) Sub-empresa.

82 ?- addSubempresa(105, 106).

```

```

?- addSubempresa(103, 106).
84 ?- addSubempresa(100, 109).

86 % 9) Contrato.

88 ?- addContrato(1,1,007,106,'empreitadas de obras publicas',
      'concurso publico','luke comprou 100g de po',
90      1000, 200, 'Amarante', data(2015,10,10)).

92 ?- addContrato(2,1,001,105,'locacao de bens moveis',
      'ajuste direto','servicos basicos',
94      2000, 100, 'Lisboa', data(2011,10,10)).

96 % 7.1) Fiscaliza.

98 ?- addFiscaliza(002, 1, data(2016,10,10)).
?- addFiscaliza(003, 2, data(2016,10,10)).
100

102 % #####

104 % #####
% Povoamento de conhecimento imperfeito.
106

% 1) Pessoa.
108

% CONHECIMENTO INCERTO.
110

% ---- 1) O Alberto foi inserido no sistema, porem a sua
112 % morada e desconhecida.
      pessoa(015,'Alberto Dias',morada00).
114 excecacao(pessoa(N,Na,Mo)) :-
      pessoa(N,Na,morada00).
116

% ---- 2) O Rodolfo foi inserido no sistema, porem a sua
118 % morada e desconhecida.
      pessoa(016,'Rodolfo Dias',morada01).
120 excecacao(pessoa(N,Na,Mo)) :-
      pessoa(N,Na,morada01).
122

% ---- 3) Existe uma pessoa em Ermesinde que esta envolvida
124 % em muitas contratos, porem devido a um erro informatico
% o nome desta pessoa foi perdido, porem sabe-se que o nome
126 % dessa pessoa certamente nem e 'Pedro' ou 'Rodrigo'.
      -pessoa(017,'Pedro','Ermesinde').
128 -pessoa(017,'Rodrigo','Ermesinde').
      pessoa(017,nome00,'Ermesinde').

```

```

130 excecacao(pessoa(N,Na,Mo)) :-
      pessoa(N,nome00,Mo).
132
133 % CONHECIMENTO IMPRECISO.
134
135 % ---- 4) Sabe-se que o Rui possui o NIF 18, porem nao se
136 % sabe se ele mora na madeira ou em lisboa.
      excecacao( pessoa(018,'Rui','Madeira') ).
138 excecacao( pessoa(018,'Rui','Lisboa') ).
139
140 % ---- 5) Sabe-se que existe uma pessoa com o NIF 19 que mora
141 % em Barcelos, nao se sabe o nome ao certo. Porem, sabe-se que
142 % o nome dessa pessoa esta, alfabeticamente, entre 'Filipe' e
143 % 'Paulo'.
144 excecacao( pessoa(019,Nome01,'Barcelos') ) :-
      Nome01 @>= 'Filipe', Nome01 @=< 'Paulo'.      %>
146
147 % ---- 6) Sabe-se que o NIF 20 chama-se 'Pedro' e mora em Lisboa.
148 % Porem, suspeita-se que na realidade o NIF 20 tenha indicado
149 % uma identidade falsa, e que na verdade o verdadeiro nome comeca
150 % com R.
      pessoa(020,'Pedro','Lisboa').
152 excecacao( pessoa(020,N,'Lisboa') ) :-
      N @>= 'R', N @< 'S'.
154
155 % CONHECIMENTO INTERDITO.
156
157 % ---- 7) Sabe-se que existe um individuo com o NIF 21, porem
158 % esta pessoa usufrui de protecao diplomatica, pelo que o
159 % seu nome e morada sao totalmente desconhidos, e nunca podem
160 % ser conhecidos.
      pessoa(021,nome02,morada02).
162 excecacao(pessoa(N,Na,Mo)) :-
      pessoa(N,nome02,morada02).
164
165 nulo(nome02).
166 nulo(morada02).
167
168 +pessoa(_,_,_):(
      solucoes((Nome,Morada),
170      (pessoa(021,Nome,Morada),
      nao(nulo(Nome)),
172      nao(nulo(Morada))), R),
      comprimento(R,0)).
174
175 % 2) Empresa.
176

```

```

178 % CONHECIMENTO INCERTO.
179
180 % ---- 1) A empresa 'Kebabz' com NIF 122 esta inserida
181 % num local desconhido, pelo que nao se sabe a sua morada.
182 % Porem, sabe-se que nao e em Lisboa.
183 -empresa(122,'Kebabz','Lisboa').
184 empresa(122,'Kebabz',morada10).
185 excecacao(empresa(N,Na,Mo)) :-
186     empresa(N,Na,morada10).
187
188 % CONHECIMENTO IMPRECISO.
189
190 % ---- 2) Existe uma empresa no Porto com um nome desconhecido
191 % porem, sabe-se que este nome certamente nao comeca com 'A' ou
192 % 'B'.
193 -empresa(123,Nome10,'Porto') :-
194     Nome10 @>= 'A', Nome10 @< 'C'.
195 empresa(123,nome10,'Porto').
196 excecacao( empresa(N,Na,Mo) ) :-
197     empresa(N,nome10,Mo).
198
199 % CONHECIMENTO INTERDITO.
200
201 % ---- 3) Existe uma empresa sediada em Malta, porem, como esta
202 % sediada num paraíso fiscal, o seu nome e desconhecido. E
203 % nunca podera vir a ser conhecido.
204
205 empresa(124,nome11,'Malta').
206 excecacao(empresa(N,Na,Mo)) :-
207     empresa(N,nome11,Mo).
208
209 nulo(nome11).
210
211 +empresa(_,_,_):(
212     solucoes((Nome,Morada),
213         (empresa(124,Nome,'Malta'),
214             nao(nulo(Nome))), R),
215     comprimento(R,0)).
216
217 % 3) Crime.
218
219 % ---- 1) Sabe-se que a data de fim de uma pena, certamente
220 % nao e superior a sua data de inicio.
221 -crime(_,data(Y1,_,_),data(Y2,_,_)) :-
222     Y1 > Y2.

```

```

224 % CONHECIMENTO INCERTO.

226 % ---- 2) Sabe-se que a pessoa com o NIF 15 foi condenado em
227 % 2100-10-10, porem por erro juridico, nao se sabe qual sera
228 % a data final da pena.
      crime(015, data(2100,10,10), data(y0,m0,d0)).
230 execucao(crime(N,Di,Df)) :-
      crime(N,Di, data(y0,m0,d0)).
232
233 % CONHECIMENTO IMPRECISO.
234
235 % ---- 3) Sabe-se que a empresa com nif 124 foi condenada
236 % numa qualquer data entre o ano de 2100 e 2120.
      execucao( crime(124,data(Y1,_,_),data(Y2,_,_)) ) :-
238     Y1 >= 2100, Y2 <= 2120.          % >

240 % CONHECIMENTO INTERDITO.

242 % ---- 4) Sabe-se que a empresa com nif 123 foi condenada
243 % a perpetua na data de 2200, pelo que devera impossivel indicar
244 % a data de fim de pena, pois nao existe.

246 crime(123,data(2200,10,10),data(y1,m1,d1)).
      execucao(crime(N,Di,Df)) :-
248     crime(N,Di, data(y1,m1,d1)).

250 nulo(y1).
      nulo(m1).
252 nulo(d1).

254 +crime(_,_,_)::(
      solucoes((Y,M,D),
256     (crime(123,_,data(Y,M,D)),
      nao(nulo(Y)),
258     nao(nulo(M)),
      nao(nulo(D)) ), R),
260 comprimento(R,0)).

262 % 4) Interdito.

264 % NENHUM

266 % 5) Inabilitado.

268 % NENHUM

270 % 6) Administrador.

```

```

272 % NENHUM

274 % 7) Fiscal.

276 % NENHUM

278 % 7.1) Fiscaliza.

280 % NENHUM

282 % 8) Sub-empresa.

284 % NENHUM

286 % 9) Contrato.

288 % ---- 1) Contrato confidencial declarado pelo adjudicante
      % 006.
290 contrato(3,1,006,confidencial,'confidencialidade','sigilo','Pastor Dias
      garante nao divulgar identidade do investidor da sua empresa',10000,18250,
      'Rua Santa Catarina, Porto',data(2020,04,28)).
      excecacao(contrato(IDC,V,_,IDA,_,_,_,_,_,_):-
292      contrato(IDC,V,_,confidencial,_,_,_,_,_,_).
      % #####

```

Listing A.1: Script de povoamento utilizado.