



João Pedro Pereira Rico

Licenciado em Engenharia Informática

Sistema de aconselhamento de aplicação de produtos fitofarmacêuticos

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Carlos Viegas Damásio, Prof. Associado, Uni-
versidade Nova de Lisboa

Co-orientadora : Marta Maurício, Eng.^a Agrónoma, Borrego
Leonor & Irmão, S.A.

Júri:

Presidente: Nuno Manuel Robalo Correia

Arguente: Salvador Luís Bettencourt Pinto de Abreu

Vogal: Carlos Augusto Isaac Piló Viegas Damásio



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Fevereiro, 2013

Sistema de aconselhamento de aplicação de produtos fitofarmacêuticos

Copyright © João Pedro Pereira Rico, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

A agricultura moderna surge com a necessidade de aumentar a produção de bens alimentares de forma sustentável e, simultaneamente, suprindo a procura crescente. O processo de produção alimentar é realizado por agricultores e empresas agrícolas que recorrem a processos mecânicos que auxiliam a sementeira, colheita, prevenção e tratamento fitossanitário das culturas.

Estes tratamentos são realizados através da aplicação de produtos químicos, nomeadamente pesticidas, ou largada de organismos auxiliares conjugada com meios de luta cultural. No entanto, a sua utilização está sujeita a um conjunto de regras diferentes, e complexas, específicas em cada cultura, sendo que os agricultores necessitam de tomar conhecimento e obedecer a todas elas para evitar problemas na saúde humana ou animal, ou até mesmo no meio ambiente. Para isso, são gizados planos de tratamento que têm como objetivo evitar que os agricultores cometam erros nas aplicações, indicando em que situações é que os produtos devem ser usados e quais os inimigos que combatem.

Esta dissertação descreve um sistema que automatiza a instanciação de planos de tratamento de acordo com as regras de Proteção Integrada em vigor em Portugal. Este sistema foi desenvolvido numa linguagem de programação por conjuntos de resposta para a validação das regras dos tratamentos efetuados e para a geração de planos de tratamento. Além disso, o sistema efetua otimizações através de vários fatores disponíveis (e.g.: minimização de custos dos produtos, maximização da margem dos produtos, preferências de comprador/vendedor de produtos, etc) que podem ser indicados pelo utilizador para obter o melhor plano possível, consoante as suas necessidades. Para a interagir com este sistema, foi elaborada uma interface web para a inserção dos dados necessários e a visualização dos resultados obtidos.

Palavras-chave: Programação por conjuntos de resposta, Pesticidas, Plano de tratamento, Agricultura, Proteção Integrada

Abstract

The modern agriculture needs to increase food production in a sustainable manner and, simultaneously, supplying the growing demand. The food production process is done by farmers and agricultural companies which use mechanical processes that help to plant, harvest, prevention and phytosanitary treatment of the crops.

These treatments are performed by applying chemical products, namely pesticides, or shot of auxiliary bodies combined with cultural control methods. However, its use is subject to a set of different, and complex, rules specific for each crop, such that the farmers need to take notice and conform to all of them to avoid problems in the human or animal health, or even in the environment. To achieve this, treatment plans are generated that aim to prevent farmers from committing errors in the applications, indicating in which situations the products should be used and which enemies it treats.

This dissertation describes a system that automates the instantiation of treatment plans in accordance with the rules Integrated Protection in force in Portugal. This system was developed in answer set programming to validate the rules and for the generation of treatment plans. Furthermore, the system performs optimizations across multiple factors available (e.g.: minimization of product costs, maximization of product margins, buyer/seller product preferences, etc) that can be specified by the user to get the best plan possible, depending on their needs. To interact with this system, a web interface was created for entering the necessary data and to visualize the results.

Keywords: Answer Set Programming, Pesticides, Treatment plan, Agriculture, Integrated Production

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Descrição do problema	2
1.2.1	Tipos de regras e instruções	2
1.2.2	Planos de tratamento	4
1.2.3	Integração da Informação	6
1.3	Objetivos	7
1.4	Contribuições	7
1.5	Organização do documento	8
2	Estado de arte	9
2.1	iGreen	9
2.2	SAIFA	11
2.3	AgroVOC	13
2.4	AgroXML	14
2.5	AGRO EDI Europe	16
2.6	EPPO	17
2.7	Sanco Pesticides	18
2.8	Outras fontes de informação	19
2.8.1	Direção-Geral de Alimentação e Veterinária	19
2.8.2	Direção-Geral de Agricultura e Desenvolvimento Rural	20
2.8.3	Empresas de produtos fitofarmacêuticos	20
2.9	Conclusão	20
3	Tecnologias	23
3.1	Modelos de informação	23
3.1.1	Modelo relacional	23
3.1.2	RDF e RDF Schema	24
3.1.3	OWL	27

3.2	Programação por conjuntos de resposta	28
3.3	Java	31
3.4	Conclusão	32
4	Abordagem de Solução	33
4.1	Modelação da informação	33
4.2	Arquitetura do sistema	36
4.3	Funcionalidades do sistema	37
5	Implementação	39
5.1	Carregamento de dados	39
5.2	Geração de factos	42
5.3	Geração de tratamentos	45
5.3.1	Ciclo cultural	45
5.3.2	Tratamentos curativos	47
5.3.3	Tratamentos Preventivos	48
5.4	Regras de Proteção Integrada	50
5.4.1	Regras de aplicação	50
5.4.2	Número máximo de aplicações	51
5.4.3	Regras de restrições temporais	51
5.4.4	Limitações de uso	52
5.4.5	Regras de condições meteorológicas	53
5.5	Custos e margens dos produtos	54
5.6	Otimização	56
5.7	Aplicação	57
5.7.1	Gestão de <i>inputs</i>	57
5.7.2	Demonstração de um plano	60
5.8	Conclusão	62
6	Validação	63
6.1	Metodologias utilizadas	63
6.2	Testes e resultados	64
6.2.1	Cenário 1	64
6.2.2	Cenário 2	66
6.2.3	Cenário 3	68
6.2.4	Cenário 4	68
6.2.5	Cenário 5	70
6.2.6	Cenário 6	72
6.3	Conclusão	74
7	Conclusão e trabalhos futuros	77

<i>CONTEÚDO</i>	xi
A Anexos	83
A.1 Datas oficiais para comercialização e utilização dos produtos	83
A.2 Substâncias ativas	84
A.3 Níveis económicos de ataque	85

Lista de Figuras

1.1	Regras de aplicação do produto comercial Tor	4
1.2	Plano de tratamento para o tomate	5
2.1	Esquema AgroRDF das culturas [17]	10
2.2	Esquema AgroRDF da proteção de plantas [17]	11
2.3	Arquitetura do sistema SAIFA [29]	12
2.4	Exemplo de interface do sistema SAIFA [29]	13
2.5	Termos em AgroVOC	14
2.6	Estrutura de um documento AgroXML	15
2.7	Aplicações possíveis com AgroXML [19]	16
2.8	Representação do conceito “Videira” no EPPT	18
2.9	Informações acerca de uma substância ativa no Sanco Pesticides	19
3.1	Um grafo RDF [22]	25
3.2	Representação de um <i>statement</i>	25
3.3	Utilização de um <i>blank node</i> [22]	26
3.4	Estrutura da linguagem OWL	28
4.1	Diagrama de classes	34
4.2	Arquitetura do sistema	36
4.3	Diagrama de casos de uso	38
5.1	Resultados de uma consulta no <i>endpoint</i> do <i>AgroVOC</i>	40
5.2	Diagrama de classes para a escrita de factos lógicos	42
5.3	Página inicial da aplicação	57
5.4	Página de observações de inimigos	58
5.5	Formulário de adicionar observações	58
5.6	Remoção de uma observação de inimigo	59
5.7	Escolha de otimizações	60
5.8	Plano de tratamentos	61

Lista de Tabelas

6.1	Plano de tratamento do cenário 1	65
6.2	Custo/margem total do plano original do cenário 1	65
6.3	Tratamentos substituídos no cenário 1	66
6.4	Custo/margem total do novo plano do cenário 1	66
6.5	Plano de tratamento no cenário 2	67
6.6	Custo/margem total do plano original do cenário 2	67
6.7	Tratamentos substituídos no cenário 2	68
6.8	Custo/margem total do novo plano do cenário 2	68
6.9	Plano de tratamento do cenário 3	69
6.10	Plano de tratamento do cenário 4	69
6.11	Plano de tratamento do cenário 5	70
6.12	Custo/margem total do plano original do cenário 5	70
6.13	Tratamentos substituídos no cenário 5	71
6.14	Custo/margem total do novo plano do cenário 5	71
6.15	Plano de tratamentos do cenário 6	72
6.16	Custo/margem total do plano original do cenário 6	73
6.17	Tratamentos substituídos no cenário 6	73
6.18	Custo/margem total do novo plano do cenário 6	74

Listagens

3.1	Exemplo de um documento RDF [22]	25
3.2	Exemplo de query em SPARQL	27
3.3	Exemplo de programa em programação por conjuntos de resposta [2]	29
3.4	Programa lógico com otimização	30
3.5	Resultado do programa lógico	31
5.1	Exemplo de consulta SPARQL ao <i>endpoint</i> do <i>AgroVOC</i>	40
5.2	Inserção dos dados em <i>Java</i> com o <i>Apache Jena</i>	41
5.3	<i>Inputs</i> das observações do estados fenológicos	46
5.4	Geração do ciclo cultural	46
5.5	<i>Inputs</i> das observações de pragas	47
5.6	Gerador de tratamentos curativos	48
5.7	Gerador de produtos para os tratamentos curativos	48
5.8	<i>Inputs</i> para os tratamentos preventivos	49
5.9	Gerador de tratamentos preventivos	49
5.10	Gerador de produtos para tratamentos preventivos	50
5.11	Restrições de aplicação	51
5.12	Número máximo de aplicações de uma substância ativa	51
5.13	<i>Input</i> da previsão de colheita	51
5.14	Regras para o intervalo de segurança	52
5.15	Famílias químicas em programação por conjuntos de resposta	53
5.16	Regra de limitações de uso	53
5.17	<i>Inputs</i> de previsões meteorológicas	53
5.18	Regras relativas a condições meteorológicas	54
5.19	Regras para obter custos máximos e mínimos	54
5.20	Regras para atribuir custos e margens aos produtos utilizados	55
5.21	Otimização de planos de tratamento	56



Introdução

1.1 Motivação

A agricultura moderna surge, em meados do século XX, com o objetivo de aumentar a produção de bens alimentares em grande escala. O processo de produção alimentar é realizado por agricultores e empresas agrícolas que recorrem a equipamentos que ajudam a semear, colher e tratar grandes culturas, otimizando todo o processo de produção agrícola, e pela indústria alimentar que é responsável pela transformação e diversificação de produtos, assumindo um papel intermediário entre a produção agrícola e o consumidor, incorporando valor acrescentado, permitindo regularidade do consumo.

Quando a agricultura adquiriu características de atividade económica, a qual, por definição, tem como objetivo a obtenção de lucro, passaram a ocorrer condições para se intensificar o uso dos fertilizantes, em particular dos adubos minerais, e pesticidas.

Os adubos minerais apresentam nutrientes em formas de mais rápida absorção pelas plantas, o que se traduz em efeitos mais visíveis no aumento do crescimento vegetal e da quantidade das produções obtidas. Os pesticidas agrícolas são substâncias, ou misturas de substâncias, destinadas a prevenir e combater os inimigos das culturas e produtos agrícolas. Estes dividem-se em vários tipos, sendo os principais: fungicidas, herbicidas e inseticidas. Os inimigos das culturas podem ser agrupados em pragas, doenças e infestantes e os pesticidas, por sua vez, podem ser classificados em função da natureza do inimigo a combater.

A aplicação destes produtos fitofarmacêuticos está sujeita a diferentes regras que foram aprovadas pela União Europeia, e pelos seus estados membros, após terem sido sujeitos a testes científicos intensivos, pois a aplicação incorreta pode causar problemas na saúde humana ou animal, ou até mesmo no meio ambiente.

Nos últimos anos, surgiu uma tendência de usufruir do avanço da tecnologia para facilitar e melhorar o trabalho na produção agrícola. As empresas agrícolas têm tirado proveito da tecnologia para registrar todo o processo de produção, como por exemplo, datas de sementeira/plantação, aplicações de fertilizantes e pesticidas que foram realizadas, área e número de culturas cultivadas/colhidas, produção vendida, entre outros. Cada vez mais se aposta na agricultura e, hoje em dia, existem muitas empresas que se dedicam a implementar *software* e equipamentos para vender às empresas agrícolas e indústrias alimentares (e.g., ISAGRI¹ e Agrogestão²).

1.2 Descrição do problema

Os pesticidas contêm substâncias ativas que podem ser perigosas para a saúde humana ou animal e por isso todos os produtos possuem regras que os responsáveis pela produção agrícola necessitam de ter conhecimento e obedecer estritamente.

A Direção-Geral de Alimentação e Veterinária (DGAV)³ disponibiliza um conjunto de documentos relativos a cada cultura contendo as diversas regras para as substâncias ativas utilizáveis no tratamento da cultura, em função do problema. Adicionalmente, para cada substância ativa, os documentos indicam os nomes dos produtos comerciais em que esta contida, bem como a formulação, a concentração, o intervalo de segurança, número de aplicações, classificação toxicológica do produto formulado e observações [8]. O pesticida pode ser designado pelo nome da substância ativa. A par do nome vulgar, a nomenclatura abrange também o nome químico, a fórmula química da substância ativa e, ainda, o nome comercial. Estes podem ser classificados em função da natureza do inimigo a combaterem (e.g., inseticidas, fungicida, herbicida, etc), segundo as famílias químicas (e.g., anilino pirimidinas, ditiocarbamatos, piretroides, etc) e, também, pela sua toxicidade [20].

1.2.1 Tipos de regras e instruções

Relativamente às regras, os documentos possuem diversos tipos de regras e instruções em relação às substâncias ativas, tal como se pode observar no anexo A.2 com alguns exemplos de substâncias ativas contidas nos fungicidas [8]. Através de uma análise a estes documentos, estas podem classificar-se da seguinte forma, com alguns exemplos de regras aplicadas à cultura do tomate:

- **Regras de aplicação:** onde se restringe o modo e o momento em que um produto deve, ou não, ser aplicado.

Ex: - Os tratamentos são efetuados na altura da sementeira e/ou plantação.

- Tratar aquando da eclosão dos ovos/aparecimento das primeiras larvas.

¹<http://www.isagri.pt/>

²<http://www.agrogestao.com/>

³<http://www.dgv.min-agricultura.pt/>

- Não aplicar em tomateiro em estufa.

- **Número máximo de aplicações:** restrições do número de máximo de aplicações permitidas ao longo de todo o seu ciclo cultural até ao momento da colheita.

Ex: - Máximo duas aplicações.

- Realizar no máximo 3 aplicações por campanha.

- **Regras de restrições temporais:** regras que indicam o número de vezes que se pode aplicar, o intervalo que se deve aguardar entre as diversas aplicações e o intervalo de segurança para cada cultura.

Ex: - 3 dias em estufa e 7 dias ao ar livre. (Intervalo de segurança)

- Aplicar ao aparecimento da praga sobre as larvas (preferencialmente as mais jovens), repetindo, se necessário, com um intervalo de 7 a 14 dias. (Intervalo de aplicações)

- **Limitações de uso:** existem substâncias que possuem limitações ao seu uso para que os inimigos que combatem não criem resistências.

Ex: - Fungicida do grupo dos QoI, realizar no máximo 3 aplicações por ciclo cultural e no conjunto das doenças, com fungicidas deste grupo.

- Não efetuar mais do que um tratamento com este produto, nem recorrer a outro fungicida com o mesmo modo de ação (anilinopirimidina).

- **Regras de condições meteorológicas:** inimigos das culturas que devem ser tratados quando ocorrem determinadas condições meteorológicas e ter em atenção que a eficácia do produto depende das condições meteorológicas que se verificam durante e algumas horas depois da aplicação.

Ex: - Tratar quando o tempo decorra húmido ou chuvoso e as temperaturas mínimas sejam superiores a 10°C.

As empresas que se dedicam à comercialização dos produtos têm a obrigação de colocar as regras na descrição dos mesmos, consoante as que se encontram nos documentos disponibilizados pela DGAV, incluindo outras informações como compatibilidades com outros produtos da empresa, características, observações e informações complementares. A figura 1.1 ilustra algumas regras de aplicação do produto comercial *Tor*, um fungicida comercializado pela *SAPEC Agro*⁴ usado no combate ao míldio da videira, batateira e tomateiro. Como se pode ver, um produto pode ser aplicado em diversas culturas, mas apenas em certas situações. Para além disso, é indicado a concentração, a dose recomendada a aplicar e o intervalo de segurança (I.S.) para cada cultura. Existem outros produtos que possuem intervalo de concentração em função da severidade do ataque.

⁴<http://www.sapecagro.pt>

CULTURA	PROBLEMA	CONC g/hl	DOSE kg/ha	RECOMENDAÇÕES DE APLICAÇÃO	IS	PI
Batateira	Mildio	300	3	Os tratamentos deverão ser realizados quando o tempo decorra húmido ou chuvoso, a intervalos não superiores a 10 dias, reduzidos para 7 em condições de forte pressão da doença.	7 dias	✓*
Tomateiro (ar livre)				Iniciar os tratamentos no viveiro e após a transplantação em condições de humidade elevada. Para consumo em fresco: o intervalo entre tratamentos com este e outros fungicidas contendo ditiocarbamatos (Metirame, propinebe e manébe) após o início da floração, não deve ser inferior a 2 semanas. Alternar se necessário com fungicidas de outros grupos químicos.	3 dias tomate em fresco 28 dias tomate indústria	
Videira				Iniciar as aplicações quando a planta estiver susceptível e as condições climáticas forem favoráveis à ocorrência da doença, repetindo a intervalos de 8-10 dias.	28 dias uva de mesa 56 dias uva para vinificação	

* máximo 2 aplicações em Protecção Integrada da vinha

Figura 1.1: Regras de aplicação do produto comercial Tor

Estas, e muitas outras, regras são disponibilizadas através de várias empresas e direções nacionais, e internacionais, para além das que foram referidas até ao momento. A modelação e criação de padrões para efetuar a interação de toda esta informação trata-se de uma tarefa muito complexa devido às diversas formas de apresentação de cada uma. Acresce que as próprias regras descritas são, por vezes, conjugações de vários tipos de condições. Por exemplo, a regra relativa à videira na figura 1.1, numa só frase, indica que o produto pode ser aplicado num determinado estado da cultura e estado meteorológico.

Nesta dissertação considera-se apenas a cultura do tomate, pois possui bastantes substâncias ativas autorizadas e, conseqüentemente, uma grande variedade de regras.

1.2.2 Planos de tratamento

Devido à grande diversidade de regras dos vários produtos do mercado, os produtores agrícolas necessitam de recorrer a planos de tratamento para evitar cometer erros nas aplicações.

As empresas de agroquímicos disponibilizam planos apresentados em forma de tabela, indicando quais os produtos/substâncias ativas a aplicar para uma determinada doença, praga ou infestante em função do estado fenológico da cultura e severidade do ataque do inimigo. Todos os produtos contidos no plano são válidos e cumprem as suas regras de utilização. Um exemplo de um plano de tratamento do tomate, disponibilizado pela SAPEC Agro, encontra-se na figura 1.2. As cores indicam os tipos dos produtos, onde verde corresponde aos herbicidas, azul aos fungicidas, vermelho aos inseticidas e amarelo a outros. Alguns dos produtos que se encontram no plano são utilizados para efetuar tratamentos preventivos, com vários intervalos entre as aplicações que variam consoante as condições verificadas da cultura. Desta forma, os agricultores conseguem diminuir a probabilidade de emergência do inimigo correspondente ao produto utilizado. Os intervalos entre as aplicações podem ser reduzidos para lidar com condições meteorológicas

específicas que podem prejudicar a ação do tratamento. Ao contrário deste tipo de tratamentos, existem os tratamentos curativos que são aplicados quando a cultura se encontra infetada. Ambos os tipos de tratamento podem ser efetuados simultaneamente, caso os produtos utilizados na cultura assim o permitam.

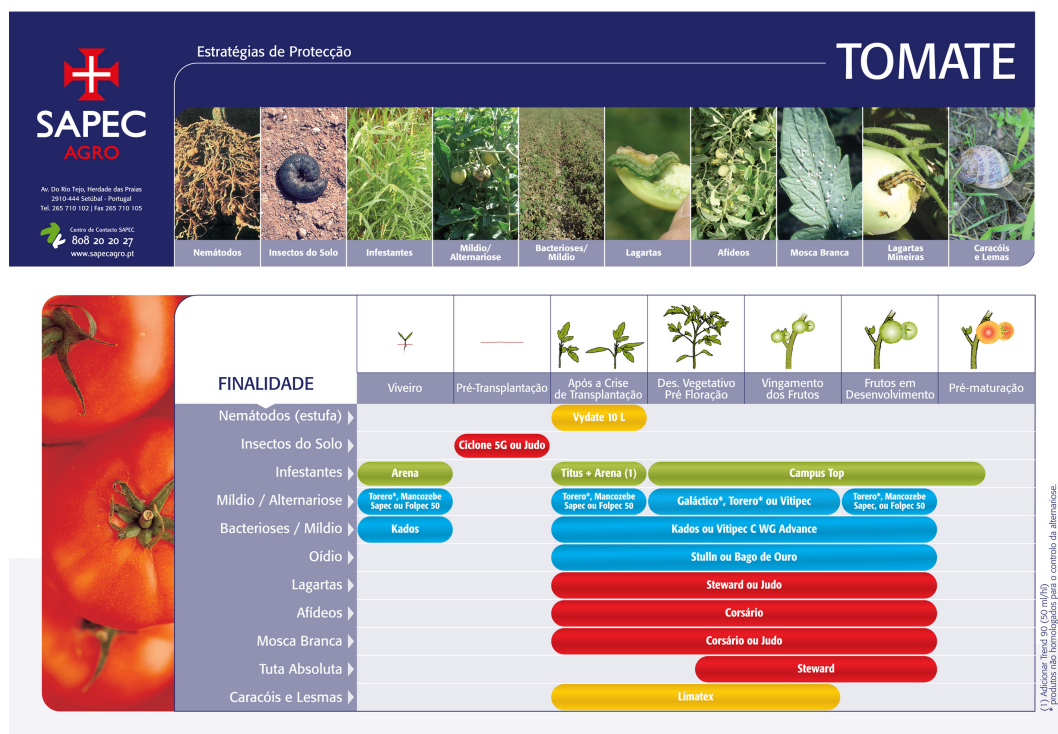


Figura 1.2: Plano de tratamento para o tomate

O problema dos planos de tratamentos é que apenas restringem as situações em que o produto deve ser aplicado e evita a utilização de produtos incorretos. Não disponibilizam as misturas incorretas, recomendações, indicações e observações para determinadas situações. Estas informações complementares encontram-se nos rótulos e fichas técnicas dos produtos. Muitas das empresas de agroquímicos disponibilizam estas instruções, juntamente com a tabela.

Cada uma destas empresas possui técnicos responsáveis pela construção dos planos de tratamento, o que não é uma tarefa imediata, pois estes técnicos necessitam do conhecimento de todas as regras dos diversos produtos, preocuparem-se com o custo dos mesmos e tentar construir um plano, consoante as suas necessidades, para os distribuidores e produtores agrícolas, tendo em conta os vários fatores: o potencial lucro, os custos com os produtos adquiridos para o próprio tratamento, as datas de autorização para utilização e os Níveis Económicos de Ataque (NEA) que estão disponibilizadas em documentos oficiais da Direção-Geral e Desenvolvimento Rural (DGADR)⁵.

Os NEA são definidos para cada cultura e para cada inimigo, como a intensidade de

⁵<http://www.dgadr.mamaot.pt/>

ataque a que se deve aplicar medidas limitativas, ou de combate, para impedir que a cultura sofra o risco de prejuízos superiores ao custo das medidas de luta a adotar, juntamente com os efeitos indesejáveis que estas últimas possam provocar [23]. Em Proteção Integrada privilegia-se, sempre, os métodos de luta cultural, biotécnica, biológica e só último caso se recorre à luta química. No anexo A.3 encontra-se um exemplo de uma tabela com os NEA para o tratamento de larvas mineiras que afetam a cultura de tomate, retirado de um documento oficial da DGADR [20]. Nesta dissertação são abordadas somente estratégias de luta química.

Relativamente à escolha dos produtos/substância ativas para um plano de tratamento, muitos dos técnicos ou empresários agrícolas podem ter mais confiança em certas marcas comerciais de produtos, dando-lhes prioridade em relação a outros na altura de instanciação do plano de tratamento, ou seja, nas situações de escolha dos produtos a serem incluídos num plano de tratamento, têm tendência a dar mais importância às soluções das marcas que mais conhecem e confiam relativamente a outras.

1.2.3 Integração da Informação

Para o cumprimento das regras relativas à Proteção Integrada, de modo a instanciar os planos de tratamento, é necessário ter em conta informação bastante variada, como é possível depreender nas secções 1.2.1 e 1.2.2. Para o modelo de dados, os conceitos necessários para a integração de toda a informação, possuem diversas propriedades que os descrevem.

Como informação biológica é necessário saber os atributos relativos às culturas e aos seus inimigos. As culturas são descritas pelo seu nome vulgar e nome científico. Todas estas passam por diversos estados fenológicos, desde a sementeira/plantação até à colheita da cultura, como foi visto anteriormente no plano de tratamento do tomate na figura 1.2. Tal como as culturas, os inimigos podem ser identificados pelos seus nomes científicos.

As substâncias ativas são a informação mais importante de toda esta integração, pois é necessário o conhecimento de todas elas para a criação de planos de tratamento. Estas são descritas tal como no anexo A.2, isto é, pelo seu tipo (e.g. fungicida, inseticida, herbicida, etc), formulação, classificação de risco, família química, incluindo os produtos comerciais em que esta contém determinada substância ativa, concentração, intervalos de segurança, inimigos e cultura em que se aplicam, número de aplicações possíveis e as regras de utilização.

É necessário ter em conta os custos e as doses recomendadas dos produtos comerciais relativos às substâncias ativas a utilizar, além do nome a que foram designados, para permitir a criação de planos de tratamentos económicos.

O estado meteorológico também é importante, pois algumas regras de utilização das substâncias ativas implicam a realização do tratamento quando ocorrem ou se preveem determinadas condições climáticas.

Por fim, é necessário ter o conhecimento das observações registadas pelos técnicos. Estes registos encontram-se nos cadernos de campo, específicos para cada cultura, onde estão reunidas as informações dos tratamentos efetuados, as substâncias ativas utilizadas, incluindo as datas da sua aplicação, os produtos comerciais utilizados, o inimigo alvo, o estado fenológico em que a cultura se encontra e data de colheita previsível.

1.3 Objetivos

O principal objetivo desta dissertação é o desenvolvimento de um sistema de aconselhamento na utilização de produtos fitofarmacêuticos que irá ajudar o utilizador a seguir as regras de Proteção Integrada, não cometendo erros na criação e implementação de planos de tratamento de uma cultura, visando a melhoria da sua produção e sustentabilidade. O sistema também irá estimar os custos dos produtos a aplicar e a determinar o plano mais económico para o empresário agrícola.

Os planos de tratamento a seguir serão instanciados a partir da informação integrada sobre as culturas, inimigos, produtos, substâncias ativas, estados fenológicos, estados meteorológicos e observações técnicas, tal como descrito na secção 1.2.3, que são validados de forma a cumprir as regras de Proteção Integrada e otimizados consoante determinados fatores (e.g.: minimização de custos dos produtos, maximização da margem dos produtos, preferências de comprador/vendedor de produtos, etc). Esta dissertação considera apenas a cultura do tomate.

Posteriormente, será dada a possibilidade ao utilizador de indicar os produtos que prefere, pois muitas das vezes o agricultor já tem uma certa confiança em certos produtos, e, dessa forma, obrigar a que o sistema os inclua no plano.

1.4 Contribuições

As principais contribuições desta dissertação são:

- Modelo de informação para aplicações agronómicas;
- A implementação de um sistema de validação de regras dos produtos fitofarmacêuticos;
- Integração das informações relativas à cultura do tomate;
- A construção de planos de tratamento válidos, sua avaliação económica e otimização;
- A disponibilização de dados de teste para a demonstração e validação dos resultados.

1.5 Organização do documento

O documento encontra-se dividido em 7 capítulos. O próximo capítulo trata do estado de arte onde se apresentam os mecanismos existentes que mais se assemelham a este sistema e que contribuem para o desenvolvimento do mesmo.

No capítulo 3 são listados os diferentes tipos de tecnologias e sistemas que se consideraram mais relevantes para o desenvolvimento desta dissertação.

O capítulo 4 será dedicado à abordagem de solução desta dissertação, indicando a forma como o sistema se encontra estruturado.

A explicação da forma como os componentes e operações deste sistema estão implementados é efetuada no capítulo 5.

Seguidamente, o capítulo 6 ilustra um conjunto de testes exercidos ao sistema para demonstrar o funcionamento, correção e desempenho do mesmo.

Por fim, o capítulo 7 inclui as conclusões acerca do trabalho realizado e algumas funcionalidades que possam ser adicionadas posteriormente.

Todos os nomes dos produtos que constam neste documento foram reduzidos de forma a censurar os nomes reais, não impedindo a demonstração do trabalho realizado.



Estado de arte

Neste capítulo é apresentado o estado de arte relevante para esta dissertação, ilustrando algumas técnicas, ferramentas e projetos existentes que contribuem para o sistema desenvolvido.

Inicialmente, apresentam-se os projetos iGreen e o SAIFA, nas secções 2.1 e 2.2 respetivamente, devido à sua semelhança com o sistema a desenvolver e ao uso de *standards* para troca de informação agrícola, como AgroVOC e AgroXML, nas secções 2.3 e 2.4 respetivamente. Nas secções 2.5, 2.6 e 2.7 são referidos o AgroEDI, EPPO e Sanco, respetivamente, que são organizações que disponibilizam dados e taxonomias relativos à agricultura e utilizados nesta dissertação. A secção 2.8 contém outras fontes de informação acerca dos tratamentos das culturas e dos pesticidas que são utilizados pelas empresas. Por fim, a secção 2.9 contém uma pequena conclusão acerca da relação dos vários sistemas apresentados com os objetivos desta dissertação.

2.1 iGreen

O *iGreen*¹ é um projeto alemão que visa desenvolver uma rede heterogénea de serviços baseados em localização e de gestão de conhecimento, através da integração de diversas fontes de informação públicas e privadas. O projeto já conta com 24 parceiros de diversas áreas de economia, ciência e do setor público.

O projeto faz a gestão de dados de diversas áreas a nível agrícola, como por exemplo, os processos de tratamentos de culturas efetuados por um agricultor. Neste caso, as fontes dos dados podem ser as informações dos sensores que determinam o fornecimento

¹ <http://www.igreen-projekt.de/iGreen/>

de nutrientes das plantas, bem como dados introduzidos pelo próprio agricultor num terminal e a transferência dos mesmo para outro terminal.

Tendo em conta que toda esta informação se encontra numa rede heterogénea, é necessário o uso de ferramentas e protocolos para tornar possível a transferência de informação entre os diversos dispositivos. O projeto *iGreen* faz o uso de diversos *standards* como o ISOBUS² para a comunicação com os sensores que se encontram nas máquinas agrícolas com a ajuda do padrão ISOXML para a troca dos dados, para além disso usa vocabulários como o *AgroVOC* (tesauro agrícola) e o *AgroXML* (interface padrão), que se encontram mencionados nas secções 2.3 e 2.4 respetivamente. Para as informações geográficas, a aplicação usa *Geography Markup Language* (GML)³. Este projeto também tira proveito das tecnologias da *web* semântica [15] para os diversos atributos e valores através do *AgroRDF*⁴.

Este projeto possui dois esquemas em *AgroRDF* para modelar o conhecimento necessário para o funcionamento do sistema. O primeiro contém informação acerca das culturas, incluindo as variedades de cada uma, com os respetivos identificadores as diversas datas de proteção e aprovação, e quais os produtores e técnicos responsáveis pelo seu tratamento, com as respetivas informações pessoais, como se pode observar na figura 2.1. O segundo esquema gera a informação relativa à proteção de plantas e as diferentes aplicações para o tratamento das mesmas. Este esquema encontra-se na figura 2.2, contém as informações das respetivas regras das aplicações, o inimigo e a cultura em que pode ser aplicado, o intervalo das aplicações, o custo máximo do tratamento, entre outros. Também são guardados os vários produtos utilizados para efetuar as aplicações, com a marca comercial, a área em que foi afetada, a data de utilização legal, a formulação e outras propriedades que se encontram no rótulo dos produtos.

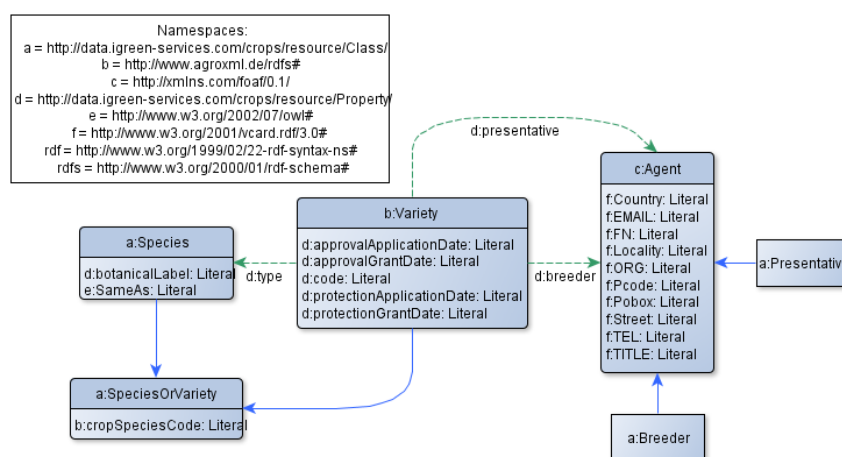


Figura 2.1: Esquema AgroRDF das culturas [17]

²<http://dictionary.isobus.net/isobus/>

³<http://www.opengeospatial.org/standards/gml>

⁴<http://data.igreen-services.com/agrordf>

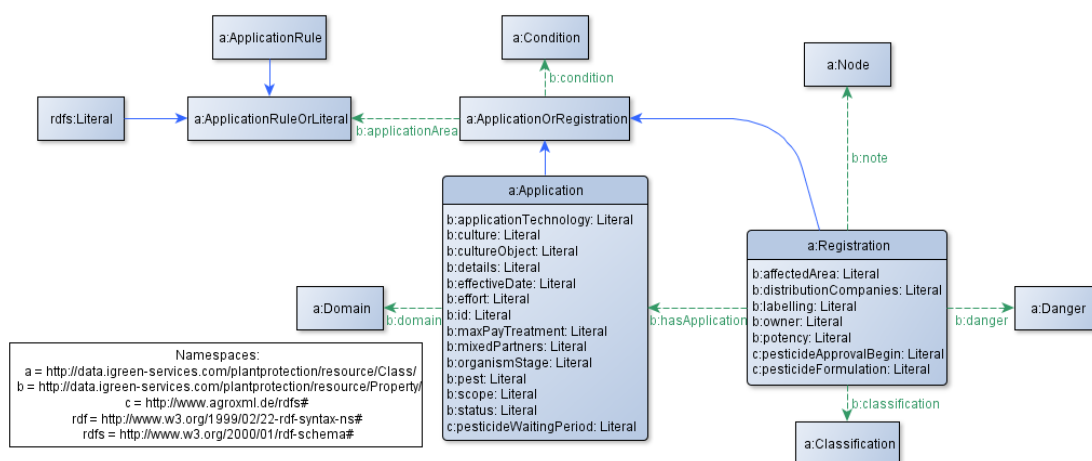


Figura 2.2: Esquema AgroRDF da proteção de plantas [17]

A forma como o *iGreen* se encontra implementado, facilita a partilha dos dados pela rede. Estes dados podem ser acedidos usando pedidos através dos protocolos habituais como HTTP, ou REST, e os resultados são devolvidos em formato RDF, que se encontra descrito na secção 3.1.2, ou XML.

2.2 SAIFA

O projeto SAIFA, designado de *Sistema de Alerta e Inforción Fitosanitaria Andaluz* é um projeto criado em Espanha, na Universidade de Almería, que resulta num sistema de informação *web* para a Produção Integrada do cultivo de azeitonas [29]. Este projeto ajuda os técnicos e coordenadores agrícolas a cumprirem com o protocolo RAIF⁵, que se trata de um sistema de alerta para a recolha de dados sobre o estado de uma cultura, monitorizando pragas e doenças.

Este sistema foi desenhado partindo de dois objetivos principais, que são:

1. Fornecer um mecanismo de decisões acerca dos tratamentos a serem efetuados nas culturas, bem como ajudá-los a cumprir com o protocolo RAIF;
2. Fornecer a capacidade de gestão aos coordenadores do RAIF e a geração de informação útil para melhor a qualidade de produção.

O SAIFA foi desenvolvido como uma aplicação *web* para que esta seja acessível em múltiplos locais, tenha a informação centralizada e garanta os diferentes tipos de necessidades para cada grupo de utilizadores.

Como o controlo dos inimigos das plantas é bastante importante para cumprir o objetivo desta aplicação, este possui um sistema de informação geográfica para disponibilizar informação georreferenciada.

⁵<http://www.juntadeandalucia.es/agriculturaypesca/raif/>

A sua arquitetura está organizada em quatro camadas diferentes, tal como se pode observar na figura 2.3.

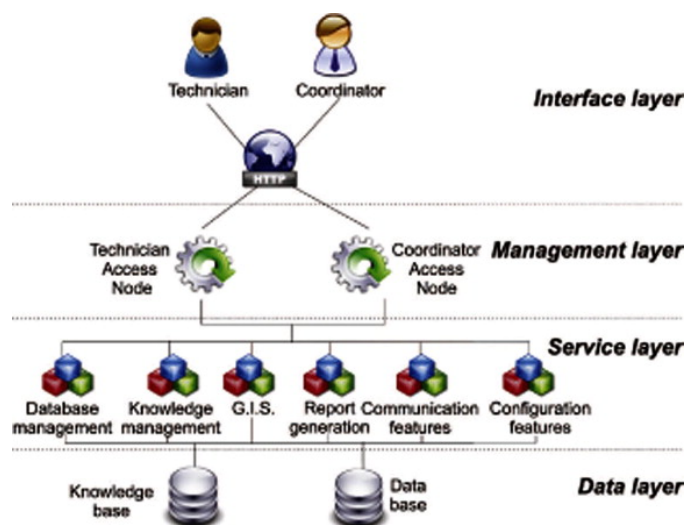


Figura 2.3: Arquitetura do sistema SAIFA [29]

A camada de interface é a responsável pela troca de informação entre o sistema e os seus utilizadores, entre os quais os técnicos e os coordenadores. Após o pedido efetuado pelo utilizador, a camada de gestão trata de resolver esse pedido, sabendo que serviços devem ser usados da camada de seguinte, a camada de serviços. A camada de serviços possui todas as características implementadas neste sistema. Por último, a camada de dados é o local onde ficam guardados todos os dados utilizados pelo sistema SAIFA, separando a informação do conhecimento.

Em termos de tecnologias, o SAIFA encontra-se implementado em *Java* através da *framework JavaServer Faces* e com gestores de bases de dados da *Oracle*. Para a execução deste sistema, é usado um servidor *Apache Tomcat* que atende os pedidos dos utilizadores e responde em HTML. A interação dos utilizadores é feita com funcionalidades *Ajax*.

Grande parte dos dados podem ser georreferenciado para que se torne mais fácil o controlo dos problemas das culturas. Consequentemente, o SAIFA foi implementado de forma a usar a API do *Google Maps* para representar esta informação.

Do ponto de vista do técnico, o sistema permite o registo de inspeções e a atribuição de ações a serem feitas às culturas. Dentro das ações a serem tomadas, o sistema permite receitar um tratamento através da seleção de produtos e a dose a ser usada, porém não os verifica/valida automaticamente. A informação dos produtos está contida na base de dados, mas também é possível encontrar informações externas acerca da substâncias ativas usadas nos mesmos. O sistema também dispõe de um conjunto de informações úteis que são disponibilizadas na interface, como se pode ver na figura 2.4.

Antes de ser designado o tratamento a ser realizado, é necessário saber se o estado da planta é tratável ou não, respeitando o protocolo RAIF. Para este caso, o sistema possui um serviço de suporte à decisão que determina se a cultura é para ser tratada ou não.



Figura 2.4: Exemplo de interface do sistema SAIFA [29]

Este serviço tira partido das redes *bayesianas* através da probabilidade de propagação que é calculada com as observações introduzidas pelo técnico. No fim, o serviço calcula o valor de probabilidade da cultura necessitar de tratamento.

Também permite ao técnico comunicar qualquer problema que encontra no sistema ou perguntar por indicações acerca do uso da aplicação.

Do ponto de vista do coordenador, o SAIFA gera, automaticamente, informação útil para facilitar a monitorização das culturas e a ter a certeza que as regras da Proteção Integrada estão a ser concretizadas. Para além disso, é possível o coordenador definir o controlo a ser efetuado pelos técnicos e quais as pragas e doenças a serem observadas, criando planos que devem ser seguidos numa determinada altura do ano. Além disso, existe a possibilidade de definir alertas para os produtores e técnicos.

2.3 AgroVOC

O *AgroVOC* é um tesouro constituído por mais de 40000 conceitos em diversas línguas, incluindo o português, relativos aos domínios de conhecimento alimentar, tais como a agricultura e pesca. Atualmente, está a ser usado como uma ferramenta de organização para o desenvolvimento de ontologias e funcionalidades de buscas multilingue.

A ferramenta é constituída por termos, de uma ou mais palavras, em que cada um representa um conceito. Estes termos encontram-se relacionados com outros através de

diferentes tipos de relações: BT (Termo genérico), NT (Termo específico), RT (Termo relacionado), UF (Utiliza-se para). A figura 2.5 ilustra a representação dos termos “Propriedade dos pesticidas” e “Tomate” no AgroVOC, com as respetivas relações. O termo “Tomate” possui algumas derivações, ao contrário do termo “Propriedades dos pesticidas”, e ambos possuem diversos termos relacionados, o que permite esclarecer o significado e o contexto em que se encontram.



Figura 2.5: Termos em AgroVOC

O *AgroVOC* está a ser usado mundialmente por investigadores, bibliotecários, gestores de informação e outros especialistas responsáveis pela organização de dados agrícolas.

O tesouro encontra-se disponibilizado em diversos formatos: RDF/XML, *N-Triples*, OWL, *MySQL* e *Protege DB*. Também é disponibilizado o *VocBench* sendo uma ferramenta, baseada na *web*, que tem como característica a possibilidade de gerir os termos do *AgroVOC* de forma livre. Em alternativa ao *VocBench*, existe um *endpoint*⁶ disponível para efetuar consultas SPARQL e obter os resultados em tripos ontológicos: RDF/XML ou *N-Triples*.

2.4 AgroXML

O *AgroXML* [18] é uma interface padrão permitindo a troca e armazenamento de informação para aplicações agrícolas. Surgiu devido à necessidade dos agricultores terem de se preocupar com a documentação e verificação de práticas agrícolas. Os primeiros conceitos desta interface foram modelados pelos seus autores, ambos em 2005 e tem vindo a

⁶<http://202.45.142.113:10035/repositories/agrovoc>

evoluir ao longo dos anos. A primeira versão foi lançada em Maio de 2006 e atualmente encontra-se na versão 1.5.

A linguagem é descrita através de um *XML Schema*⁷ que modela processos reais do mundo da agricultura. Fornece a possibilidade de modelar a produção de plantas e está a dar os primeiros passos para produção dos animais, mas o *Schema* respetivo ainda não foi testado. A figura 2.4 representa a estrutura geral de um documento *AgroXML*.

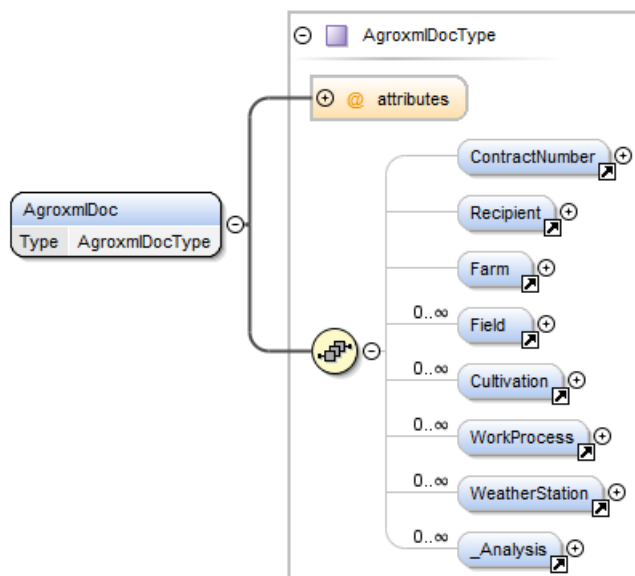


Figura 2.6: Estrutura de um documento AgroXML

Um documento *AgroXML* representa um perfil acerca de apenas uma exploração agrícola pertencente a uma empresa. As informações no documento incluem os vários conteúdos de uma exploração, as culturas que estão a ser produzidas e todos os registos que foram realizados para o tratamento das mesmas. Também dá a possibilidade de registar diversos tipos de análise químicas, mas de momento só disponibiliza campos para registar valores relativos à análise de solos. A meteorologia também é uma informação importante para os agricultores e os documentos *AgroXML* podem conter informações acerca de registos meteorológicos das várias estações.

Esta ferramenta possui uma grande diversidade de aplicações em que pode ser usada, tal como se encontra na figura 2.7. De facto, ela já é usufruída por diversas aplicações implementadas por várias empresas, como por exemplo *agriXchange* e a *iGreen*, indicada na secção 2.1.

⁷ <http://www.w3.org/standards/xml/schema>

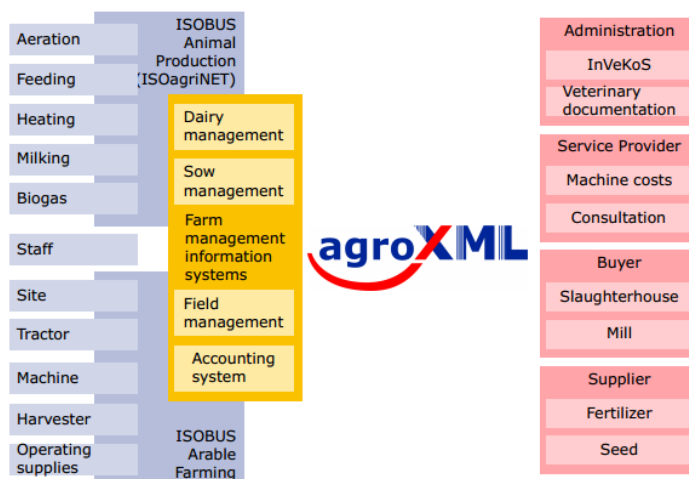


Figura 2.7: Aplicações possíveis com AgroXML [19]

2.5 AGRO EDI Europe

O *AGRO EDI Europe*⁸ (AEE) é uma associação francesa fundada em 1992 pela *Coop de France*, *InVivo* e *ANR*. Esta associação conta com mais de 240 membros em diferentes sectores da agricultura, tanto públicos, como privados. O seu objetivo é contribuir para o desenvolvimento de sistemas *Electronic Data Interchange* (EDI) para sectores de agricultura e alimentação.

O EDI é o processo de troca de informações entre empresas através do uso de computadores [10], com a criação de *software* e padrões que facilitem a comunicação. Para esse efeito, o AEE começou por definir documentos com formatos padrão para dados relacionados com as culturas.

Em 2001, o AEE definiu a *DAPLOS message*⁹ que se trata de um padrão para descrever a informação relacionada com uma cultura específica [10]. A partir da *DAPLOS message*, muitos programadores de *software* tentaram respeitar este padrão. Alguns anos depois, foi criada a UN/CEFACT, onde se começou por criar diversos projetos de troca de informação para os vários sectores, respeitando sempre o padrão do *DAPLOS message*.

Adicionalmente, em 2008, foi criado um formato *standard* com o nome de *Agronomical Observation Report (AgroObs)*¹⁰ com o intuito de melhorar a troca de informação relativa às observações efetuadas pelos técnicos agrícolas, para o controlo das pragas nas culturas, entre várias entidades de uma forma normalizada.

Para além destes *standards*, o AEE também é responsável por um repositório que contém diversos documentos com codificações para a área da agricultura. Nomeadamente, existem documentos, escritos por outras organizações, que listam a codificação,

⁸<http://www.agroedieurope.fr/>

⁹http://www.unece.org/trade/untdid/d05b/trmd/daplos_c.htm

¹⁰<http://www1.unece.org/cefact/platform/display/TBG/EDI+of+Agonomical+Observations+Report>

gerada pelo AEE, dos diversos estados fenológicos de cada cultura, designados por escala BBCH [24].

A evolução de cada cultura passa por diversas fases, que se designam de estados fenológicos que se podem manifestar de diversas maneiras. Através da escala BBCH é possível qualquer estado que seja identificado através de um simples código, o que facilita qualquer *software*, ou troca de informação entre duas empresas, a especificar o estado em que se encontra as plantas. O acesso a esta informação é limitada apenas aos membros, ao qual exige um pagamento de uma quota anual.

2.6 EPPO

A *European and Mediterranean Plant Protection Organization*¹¹ (EPPO) é uma organização intergovernamental fundada em 1951 por 15 países Europeus e conta já com 50 membros.

A organização tem o objetivo de proteger a saúde das plantas tanto na agricultura, como nas florestas, e desenvolver estratégias e métodos de controlo para o combate à intrusão e propagação de pragas das mesmas.

A EPPO é responsável por um conjunto de códigos designados de *EPPO Codes*, antigamente conhecidos por *Bayer Codes*. Estes códigos alfanuméricos permitem identificar as culturas e as respetivas pragas. A representação das culturas é feita por um código de 5 letras e os de 6 letras é utilizado para os restantes organismos.

Para além da manutenção dos *EPPO Codes*, a EPPO publica uma grande diversidade de documentação para a proteção de plantas. Também possui algumas bases de dados com informação relevante às pragas em quarentena, produtos para proteção de plantas e vocabulários dos diversos conceitos da agricultura.

O vocabulário desta organização encontra-se publicada numa ferramenta *online* com o nome *EPPO Plant Protection Thesaurus*¹² (EPPT), possui os conceitos acerca das culturas, pragas, inimigos naturais e organismos usados para estudos ecotoxicológicos. A ferramenta dá a possibilidade de efetuar pesquisas gratuitamente e para cada organismo é devolvido o nome científico principal e os outros, o *EPPO code*, nomes comuns para as diversas línguas, relações taxonómicas e outras classificações, tal como se pode depreender na figura 2.8.

As bases de dados, por sua vez, possuem informações mais detalhadas. No caso dos produtos, a base de dados sobre a proteção de plantas [27] é onde se encontra os detalhes acerca dos testes de eficácia dos produtos no campo e em estufa, incluindo a comparação com os mesmos. Em cada teste são dados as condições em que foram realizados, o modo como foram aplicados e algumas métricas, como por exemplo, dados meteorológicos. A base de dados das pragas em quarentena [26] contém toda a informação acerca das avaliações das pragas que foram recomendadas pela *EPPO Council* para serem avaliadas pelos restantes membros da EPPO. Grande parte das pragas já se encontram avaliadas,

¹¹ <http://www.eppo.int/>

¹² <http://eppt.eppo.org/index.php>

Figura 2.8: Representação do conceito “Videira” no EPPT

no entanto ainda algumas que apenas foram recomendadas pela *EPPO Council*. Estas bases de dados encontra-se disponíveis através de um pagamento simbólico.

2.7 Sanco Pesticides

O Sanco Pesticides¹³ é uma base de dados *online* pertencente à União Europeia que serve para consultar informação sobre o limite máximo de resíduos dos pesticidas e dos seus produtos, bem como as informações detalhadas acerca das substâncias ativas.

As informações relativas às substâncias ativas referem-se ao estado de aprovação em que se encontra a substância ativa, a que categoria pertence, classificação, países em que se encontra autorizada, informações toxicológicas, os relatórios que foram efetuados na revisão da mesma, entre outros detalhes. A figura 2.9 representa um exemplo de uma substância que não foi aprovada e, por isso, não se encontra autorizada em qualquer país.

Relativamente aos produtos e pesticidas, regista a evolução histórica do limite máximo de resíduos autorizados, nas diversas culturas, incluindo os documentos das legislações que foram produzidas pela União Europeia por cada alteração feita dos limites com a justificação dos mesmos.

As informações encontram-se em diversas línguas e é possível serem consultadas a partir da aplicação disponível *online* e também é possível descarregar em ficheiros XML ou Excel. Está prevista a disponibilização das informações em RDF, à qual se teve acesso prévio.

¹³http://ec.europa.eu/sanco_pesticides

Trifluralin

Status under Reg. (EC) No 1107/2009 (repealing Directive 91/414/EEC) Back

Status: Not Approved Current Legislation: 2010/355/EU

RMS: EL EFSA Risk Assessment:

Category: HB Review Report: Trifluralin 2010.pdf
 Trifluralin 2007.pdf

Remarks: Initially not included by Commission Decision 2007/629. Application resubmitted for inclusion (Reg 33/2008). Non inclusion voted May 2010.

Classification

Dir. 67/548/EEC

Xi; R36
R43
N; R50/53

Reg. 1272/2008

Skin Sens. 1 - H317 Carc. 2 - H351
Aquatic Acute 1 - H400 Aquatic Chronic 1 - H410

Authorisations at national level

No authorisation in place

Toxicological information

Trifluralin

ADI:	Source:	Remark:	ARfD:	Source:	Remark:	AOEL:	Source:	Remark:
0.015	EFSA 05		Not appl.	EFSA 05		0.026	EFSA 05	
Other:								

Where no units are shown, values are expressed in mg/kg bw/day

EU - Maximum Residue Levels (Reg. (EC) No 396/2005) (MRLs)

Legislation: Trifluralin
Reg. (EC) No 149/2008

Annexes: Trifluralin
Annex IIIA MRLs

Figura 2.9: Informações acerca de uma substância ativa no Sanco Pesticides

2.8 Outras fontes de informação

2.8.1 Direção-Geral de Alimentação e Veterinária

A Direção Geral de Alimentação e Veterinária (DGAV)¹⁴ é uma nova organização responsável pela disponibilização de documentos relativos às áreas de agricultura, alimentação e animal.

Relativamente à área da agricultura, a DGAV disponibiliza os diversos documentos relativos às substâncias ativas, contidas nos pesticidas, tal como foi visto na secção 1.2. Estes documentos possuem informações quanto ao nome das substâncias ativas, a sua formulação, concentração, intervalo de segurança entre aplicações, o nome comercial de todos os produtos em que estão inseridas, classificação química, o limite de número de aplicações autorizadas e as várias regras de utilização. Os fornecedores dos produtos fitofarmacêuticos têm a obrigação de colocar estas informações nos respetivos rótulos.

Esta organização está a atualizar outros tipos de documentos importantes e que foram anteriormente criados pela Direção-Geral de Agricultura e Desenvolvimento Rural, mencionada na secção seguinte.

¹⁴<http://www.dgv.min-agricultura.pt/>

2.8.2 Direção-Geral de Agricultura e Desenvolvimento Rural

A Direção-Geral de Agricultura e Desenvolvimento Rural (DGADR)¹⁵ é uma organização governamental para a agricultura e desenvolvimento rural em Portugal que é constituída por um conjunto de divisões para a gestão da agricultura e do desenvolvimento rural em Portugal. Esta organização ainda possui algumas regulamentações que qualquer de outra empresa, da mesma área, terá de cumprir.

Nomeadamente, para a área da agricultura, a DGADR fornece documentos com as datas de autorização para comercialização e utilização dos produtos. Estes limites irão começar a ser atualizados regularmente pela DGAV, para todos os produtos existentes no mercado, pois um produto só pode ser comercializado a partir do momento em que se encontra registado e aprovado. Um extrato de um documento que contém todas estas datas [9] encontra-se no anexo A.1.

Além disso, esta organização disponibiliza os cadernos de campo de todas as culturas, que são obrigatórios para a Proteção Integrada. Todas as empresas agrícolas necessitam de elaborar os seus próprios cadernos de campos, com base nos que são disponibilizados pela DGADR. Estes documentos servem para registar a atividade agrícola, contendo as ocorrências dos estados fenológicos das culturas, bem como os registos das aplicações dos produtos, as datas em que foram realizadas e as observações efetuadas pelos técnicos [20].

2.8.3 Empresas de produtos fitofarmacêuticos

As empresas agrícolas necessitam de recorrer aos fornecedores e distribuidores de produtos fitofarmacêuticos para obterem os produtos que necessitam para proceder ao tratamento das suas culturas.

Em Portugal existe mais de uma dezena de fornecedores entre os quais se destacam: Bayer CropScience Portugal, LusoSem, SAPEC Agro e Syngenta Portugal. Estes, e outros, fornecedores são responsáveis por colocar nos rótulos as descrições dos produtos que comercializam, tais como o nome, as características, as regras de utilização, as culturas e inimigos alvo, os limites máximos de resíduos, etc. Todas estas informações têm de estar de acordo com os documentos publicados pela DGAV, especificada na secção 2.8.1. Os limites máximos de resíduos são geridos pela Sanco Pesticides, mencionada na secção 2.7.

Além dos produtos, as empresas como a Bayer, Syngenta e a SAPEC Agro, fornecem planos de tratamento para todas as culturas com os produtos que estão a comercializar. Um exemplo de plano da SAPEC Agro foi ilustrado na secção 1.2.2.

2.9 Conclusão

Em suma, a utilização de formatos padronizados e informação em RDF é uma tendência detetada nas várias ferramentas e fontes de informação presentes neste capítulo.

¹⁵<http://www.dgadr.mamaot.pt/>

O sistema *iGreen*, apresentado na primeira secção, tem como objetivo fornecer informação a nível da agricultura através de diversas fontes de informação com o uso de normas para uniformizar o modo de transferência. Este sistema inspirou o modelo de dados desenvolvido para o suporte à aplicação construída nesta dissertação. Embora seja um pouco diferente do que foi implementado, não deixa de ter as suas semelhanças quanto ao uso de várias fontes de informação diferentes, para fornecer ao utilizador.

Na secção seguinte, é apresentado o sistema SAIFA que possui diversas características úteis para a Produção Integrada, nomeadamente a prescrição de tratamentos da cultura feitos pelos técnicos. Nesta dissertação a abordagem foi diferente, uma vez que são gerados tratamentos preventivos e curativos, com verificação automática das regras de Proteção Integrada.

Na secção 2.3, é feita uma descrição do sistema AgroVOC, que foi utilizado para a gestão dos termos agrícolas, as várias relações entre eles e informação ontológica.

O *AgroXML*, apresentado na secção 2.4, serviu de auxílio para organizar a forma como os *inputs* do utilizador são inseridos no sistema desenvolvido.

Por fim, as últimas secções descrevem várias fontes de dados utilizadas nesta dissertação. Na secção 2.6, apresenta-se o EPPO que contém os *EPPO codes* que identificam as várias culturas existentes, bem como as pragas que as afetam, que foram guardados no sistema. Através do Sanco Pesticides, descrito na secção 2.7, foi recolhida a informação acerca das várias substâncias ativas. Para os planos de tratamento é necessário saber todos os estados fenológicos das culturas que foram obtidos, incluindo os respetivos códigos BBCH através de documentos específicos e apresentados na secção 2.5. A secção 2.8 foram indicadas as diversas empresas, das quais foram recolhidas informações relativas às regras das substâncias ativas e os respetivos produtos fitofarmacêuticos.

3

Tecnologias

Neste capítulo encontram-se descritas as diversas tecnologias selecionadas para a realização desta dissertação. Começando com os modelos de informação, onde será feita uma análise das várias possibilidades para gerir e manter a informação necessária: Modelos relacionais, RDF e OWL. Na secção seguinte apresenta-se brevemente a programação por conjuntos de resposta e as possibilidades que oferece para o sistema de regras do trabalho a desenvolver. Por último, dá-se uma breve explicação acerca de linguagens imperativas, neste caso a linguagem Java. Após todas as tecnologias serem apresentadas, será feita uma pequena conclusão, justificando as razões para a seleção das tecnologias que foram utilizadas para esta dissertação.

3.1 Modelos de informação

3.1.1 Modelo relacional

O modelo relacional [5] é um modelo de dados, utilizado nos atuais Sistemas de Gestão de Base de Dados (SGBD), criado com o objetivo de esconder a forma como os dados estão organizados internamente numa infraestrutura. Este modelo baseia-se em tabelas, que se relacionam entre si, e permitem efetuar consultas, inserções, atualizações e remoções dos dados que as constituem. Um exemplo de um sistema deste tipo é o PostgreSQL.

O PostgreSQL¹ é um SGBD *open source* orientado a objetos com mais de 15 anos. Atualmente na versão 9.2.2, este SGBD foi implementado com base na norma ANSI-SQL:2008, possui quase todas as características da linguagem SQL, como por exemplo *queries* complexas, com suporte para *sub-queries*, *triggers*, *views*, entre outros.

¹<http://www.postgresql.org/>

Uma das grandes características do PostgreSQL é ser altamente customizável, dando a possibilidade de acrescentar novos tipos de dados, bem como funções, operadores, funções de agregação e métodos de indexação. As linguagens procedimentais no PostgreSQL permitem o utilizador definir novas funções através das linguagens convencionais como Java, Ruby, Python, etc. Para além disso, o código fonte encontra-se disponível sem qualquer tipo de custo e dá a liberdade ao utilizador poder consultá-lo e modificá-lo.

Esta tecnologia dispõe de diversos tipos diferentes de indexação que podem ser úteis em certas circunstâncias. A habitual indexação B-tree, que suporta os diversos operadores de comparação nas *queries*, e também o suporte de índices Hash, que são úteis para quando se efetua *queries* simples com comparações de igualdade. Para além destas duas indexações, foram implementados dois tipos, que não se encontram em qualquer outro SGBD, que são o GiST e o GIN. O GiST é um tipo de índice que funciona como uma estrutura de dados que pode ser usado para qualquer tipo de pesquisa em árvore como B-Tree e R-Tree, dando a vantagem de um programador desenvolver pesquisas com o uso de tipos de dados customizados. Por último, o GIN funciona como um índice invertido que foi implementado para os casos em que os dados a serem indexados são valores compostos, tendo a mesma vantagem que o índice GiST em que permite funcionar com tipos de dados customizados.

Outra característica importante deste sistema de gestão de base de dados é a capacidade de guardar informação geográfica, através do PostGIS². O PostGIS é um projeto desenvolvido pela *Refractions Research Inc.* com o intuito de estender o PostgreSQL, de forma a fornecer o suporte de armazenar objetos geográficos numa base de dados relacional. Desta forma, é possível armazenar a representação pontos, multi-pontos, linhas, multi-linhas, polígonos, multi-polígonos e coleções geométricas. A gestão desta informação é feita através de índices próprios de forma a que o espaço em memória seja reduzido e haja um aumento da performance.

3.1.2 RDF e RDF Schema

O *Resource Description Framework* (RDF) [22] é um modelo de informação que tem como objetivo facilitar a representação dos conteúdos da Web, nomeadamente a meta-informação sobre as diversas páginas. Como grande vantagem, torna possível a leitura da informação por outras aplicações, facilitando a troca de conhecimento entre as mesmas. Uma das serializações de sintaxe do RDF recorre à linguagem XML. O RDF é uma recomendação da W3C desde 1999, até esta ter sido revista e publicada a sua correção em 2004. Também existe o uso de JSON como alternativa ao XML, ao que ainda se encontra em fase experimental, mas já se encontra a ser avaliada para recomendação no W3C [7].

Este *framework* tem um modelo de dados, baseado num grafo orientado etiquetado [1], tal como se pode verificar na figura 3.1. Este grafo pode ser descrito através de um documento XML como o do exemplo 3.1.

²<http://postgis.refractions.net/>

Listing 3.1: Exemplo de um documento RDF [22]

```

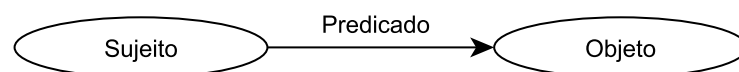
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
4   <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
5     <contact:fullName>Eric Miller</contact:fullName>
6     <contact:mailbox rdf:resource="mailto:em@w3.org"/>
7     <contact:personalTitle>Dr.</contact:personalTitle>
8   </contact:Person>
9 </rdf:RDF>

```



Figura 3.1: Um grafo RDF [22]

O grafo RDF é constituído por triplos, designados por *statements*, que contêm um sujeito, um predicado e um objeto, tal como se pode ver na figura 3.2.

Figura 3.2: Representação de um *statement*

Os predicados são propriedades que descrevem a relação entre o sujeito e o objeto, como por exemplo “criado por”, “contacto”, entre outros. Estes são os arcos de um grafo RDF que contêm uma etiqueta a identificar a propriedade que representa.

Tanto o sujeito, como o objeto são nós do grafo que representam os recursos. Cada recurso é denotado por um *Uniform Resource Identifier* (URI), em particular pode ser um URL, um literal ou um nó anónimo, designado de *blank node*. Um URL é um identificador de uma página *web*, enquanto que um literal é um valor que pode ser um texto ou um número. Para os literais podem ser usados os diversos tipos de dados fornecidos pelo

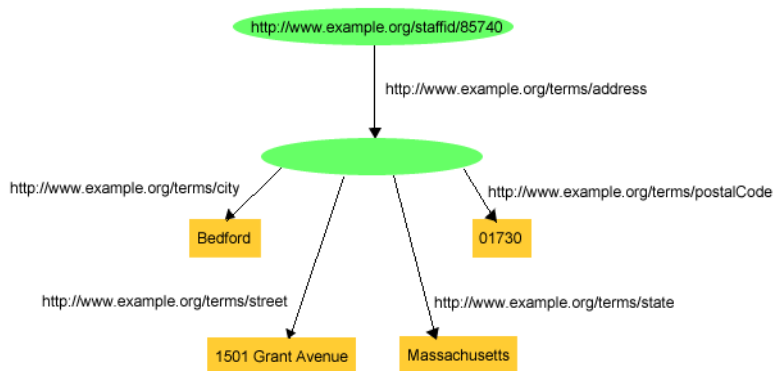


Figura 3.3: Utilização de um *blank node* [22]

XML Schema, como valores inteiros ou *strings*. Por fim, os *blank nodes* são bastante utilizados em situações que se quer agregar mais do que um objeto a um sujeito. A figura 3.3 apresenta um exemplo de uma morada, em RDF, através da utilização de um *blank node*, onde é possível representar o código postal, a cidade, a rua e o estado, separadamente, para o sujeito identificado por *http://www.example.org/staffid/85740*.

Uma desvantagem do RDF é não haver qualquer método para organizar a informação. Para isso existe o *RDF Schema* (RDFS), que fornece a capacidade ao RDF de estruturar os conteúdos *web*. Esta especificação é uma recomendação da W3C desde Fevereiro de 2004 [16], que tem o objetivo estender vocabulário ao RDF de forma a permitir a definição de classes e algumas propriedades novas. As classes são responsáveis por agrupar os vários recursos de um grafo RDF, de forma a facilitar a identificação do tipo de cada um. As propriedades implementadas no RDFS descrevem algumas relações entre as classes como domínios, tipos e sub-classes. Com isto, é possível construir uma hierarquia de classes para definir as diversas vinculações de toda a informação. Por exemplo, as figuras 2.1 e 2.2 relativas ao projeto iGreen, da secção 2.1, são estruturas do *AgroRDF* definidas através do RDFS.

Para o armazenamento de dados através desta tecnologia são usados *Triple Stores*. Os *Triple Stores* são ferramentas que servem como base de dados com coleções de triplos, tal como no RDF e outras linguagens ontológicas. Alguns exemplos relevantes de *Triple Stores* são o *Apache Jena*³ e o *Sesame*⁴.

O *Apache Jena* é uma *framework* Java dirigido a aplicações que utilizem tecnologias de Web Semântica, por exemplo o RDF e o OWL. Ao contrário do *Sesame*, que apenas suporta a linguagem RDF e possui uma API própria como interface, designada de *Sail*.

Dentro das ferramentas do *Apache Jena* está incluída uma API para a linguagem SPARQL. Semelhantemente ao SQL, o SPARQL é uma linguagem capaz de interrogar e manipular a informação numa base de dados/conhecimento, neste caso em RDF [31]. O exemplo 3.2 mostra um exemplo de uma consulta em SPARQL para a estrutura do

³<http://jena.apache.org/>

⁴<http://www.openrdf.org/>

projeto iGreen, na figura 2.2 da secção 2.1, onde se obtém o pagamento e valor de esforço de um tratamento para uma cultura. Através desta linguagem é possível extrair informação com consultas, especificando os padrões dos triplos do grafo RDF. Além disso, é possível a manipulação de informação através do SPARQL/Update, que se trata de uma extensão do próprio SPARQL dando a capacidade de inserir, alterar e remover dados de uma base de conhecimento (*Triple Store*) RDF. Esta tecnologia encontra-se na versão 1.1 e é uma recomendação da W3C de Março de 2013.

Listing 3.2: Exemplo de query em SPARQL

```

1 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2 PREFIX b: <http://data.igreen-services.com/plantprotection/resource/Property/>
3 PREFIX c: <http://www.agroxml.de/rdfs#>
4 PREFIX a: <http://data.igreen-services.com/plantprotection/resource/Class/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6
7 SELECT ?culture ?maxPayTreatment ?effort
8 WHERE
9   { ?application rdf:type a:Application .
10     ?application a:Application ?culture .
11     ?application a:Application ?maxPayTreatment .
12     ?application a:Application ?effort
13   }

```

3.1.3 OWL

A linguagem *Web Ontology Language* (OWL) [21] é outro exemplo de linguagens para representação de conhecimento em forma de ontologias. Esta linguagem tornou-se numa das recomendações do W3C em Fevereiro de 2004 e tem vindo a ser a nova aposta para expressar conhecimento na *web*, atualmente encontra-se na segunda versão, lançada em Dezembro de 2012. Tal como qualquer linguagem baseada em ontologias, a OWL surge com o objetivo de representar informação e conhecimento de forma a que seja processada automaticamente.

A figura 3.4 representa a estrutura da linguagem OWL na última versão. No topo do diagrama estão as várias formas possíveis de representar a linguagem OWL, dando a possibilidade de efetuar a troca de ontologias. O centro é onde é feito todo o mapeamento e processamento da informação e a parte inferior representa as duas especificações da semântica da linguagem.

Esta linguagem encontra-se implementada com base em algumas linguagens usadas para a Web Semântica e recomendadas pela W3C, que são: XML, *XML Schema*, RDF e *RDF Schema*.

A OWL veio acrescentar novo vocabulário ao RDF, como por exemplo relações entre classes e, mais geralmente, conceitos, cardinalidade e novas propriedades. Para além disso, acrescenta novas capacidades de inferência. Com isto, possui a capacidade de produzir novas informações a partir das que são fornecidas, através das relações entre os

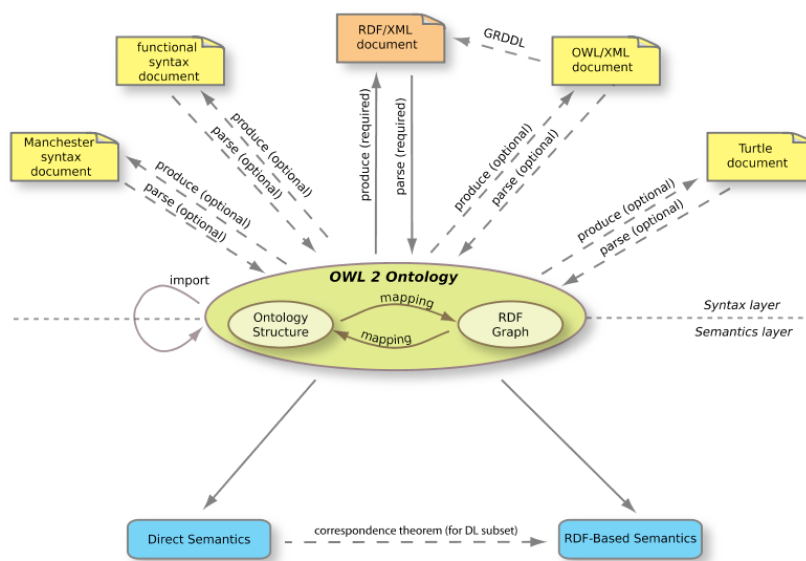


Figura 3.4: Estrutura da linguagem OWL

dados, como por exemplo relações de transitividade, simetria e igualdade. Esta linguagem é utilizada em diversas ferramentas, como por exemplo o *AgroVOC*, mencionado na secção 2.3, para a representação dos diversos termos e efetuar a referências nas várias relações entre eles.

O OWL 2 possui três perfis que constituem sub-linguagens do próprio OWL, estes são: OWL 2 EL, OWL 2 QL e OWL 2 RL [25]. O OWL 2 EL foi definido para a ser usado em ontologias com bastantes classes e propriedades definidas, envolvendo capacidades de inferência mais complexas do que nos outros perfis. O OWL 2 QL serve para aplicações com muitas instâncias com recurso a bases de dados relacionais. Por fim, o OWL 2 RL trata-se de um sub-conjunto do OWL 2, que tenta equilibrar as capacidades expressivas com escalabilidade. Este perfil está desenhado para poder ser implementado em sistemas de regras.

Esta linguagem pode ser benéfica para determinadas aplicações na recolha e mapeamento de informação e dar a possibilidade de ser disponibilizada para outras, tornando-as extensíveis.

3.2 Programação por conjuntos de resposta

Um outro tipo de tecnologia útil para esta dissertação é o uso de linguagens lógicas, nomeadamente o *Answer Set Programming* (Programação por conjuntos de resposta) [2]. A linguagem de programação por conjuntos de resposta é uma linguagem de paradigma declarativo para representação de conhecimento e raciocínio com o objetivo de resolver problemas de pesquisa complexos, baseando-se em modelos estáveis da programação lógica [14]. Esta linguagem tem sido utilizada numa grande diversidade de aplicações no que toca a bases de dados, representação de conhecimento e raciocínio, nomeadamente a

integração de regras, ontologias e planeamento.

Um programa desenvolvido nesta linguagem é constituído por um conjunto de regras, factos e restrições de integridade. As regras têm a seguinte sintaxe base:

$$A_0 :- A_1, \dots, A_n, \text{ not } A_{n+1}, \dots, \text{ not } A_m$$

Os elementos A_x são átomos lógicos que podem ter argumentos, normalmente constantes ou variáveis. O operador `not` efetua a negação por omissão de um átomo, sendo `not A_j` designado por literal. Assim, o literal `not A_j` é verdadeiro se não for possível derivar o átomo A_j . Intuitivamente, o significado da regra anterior é que é possível derivar A_0 se for possível derivar A_1, \dots, A_n e não for possível derivar A_{n+1}, \dots, A_m . O conjunto de átomos e literais à direita do símbolo `:-` constituem o corpo da regra, enquanto que o átomo à esquerda (A_0) é a cabeça da regra. A semântica de modelos estáveis [4] pode atribuir a cada programa 0, 1 ou mais modelos estáveis. Neste paradigma de programação, o programador apenas especifica o que necessita para resolver o problema, em vez de indicar como é que deve ser resolvido [2]. Desta forma, cada modelo resultante corresponde a uma solução diferente para o problema que se pretende resolver.

A ordem dos literais no corpo da regra e a ordem das regras num determinado programa não interessa nesta linguagem. Há extensões à linguagem que permitem a definição de literais condicionais, átomos de cardinalidade, regras de escolha e agregados que possibilitam a simplificação da escrita dos programas e tornar a linguagem mais declarativa. Os factos são regras sem corpo que, conseqüentemente, são sempre verdadeiros, ocorrendo em todos os modelos estáveis. Contrariamente, as restrições de integridade são regras sem a cabeça, que servem para prevenir que certos modelos estáveis sejam criados. O exemplo 3.3 ilustra um programa contendo os vários tipos de regras, onde a primeira linha é um facto, a segunda linha é uma regra de escolha e a última é uma restrição de integridade. Este programa modela o problema clássico de coloração de grafos. O conjunto de factos `c(1..n)` especifica que existem n cores. A regra 2 diz que cada vértice v tem uma, e uma só, cor. A última regra indica que vértices ligados por uma aresta adjacente não podem ter a mesma cor. A instância a resolver é fornecida através de um conjunto de factos $v/1$ (vértices) e $e/2$ (arestas).

Listing 3.3: Exemplo de programa em programação por conjuntos de resposta [2]

```

1 c(1..n).
2 1 {color(X,I) : c(I)} 1 :- v(X).
3 :- color(X,I), color(Y,I), e(X,Y), c(I).

```

A programação por conjuntos de resposta é uma das alternativas à linguagem Prolog, que embora seja uma linguagem bastante desenvolvida, possui algumas limitações [2]. Por exemplo, no caso da ordem dos literais numa regra em Prolog é importante, pois a execução é feita da esquerda para a direita, caso que não acontece na programação por conjuntos de resposta. Por outro lado, algumas extensões da programação por conjuntos de resposta permitem disjunções na cabeça das regras. A execução das regras em Prolog

é feita em modo *top-down*, fazendo com que possam ocorrer casos de programas que entrem em ciclos infinitos, enquanto que em programação por conjuntos de resposta, o processo é *bottom-up*, baseado na instanciação (*grounding*) de programas com um motor de inferência, gerador dos modelos (*solver*). A linguagem Prolog obtém as respostas a partir de um único modelo para o programa, enquanto que a programação por conjuntos de resposta gera vários modelos alternativos. Contudo, a linguagem Prolog permite a utilização de termos complexos nos argumentos, o que não é normalmente permitido em programação por conjuntos de resposta.

Atualmente, existem bastantes ferramentas que interpretam esta linguagem e obtêm os modelos estáveis de um programa, nomeadamente o *Pottasco*. O *Pottasco*⁵ é um projeto desenvolvido pela Universidade de *Postdam*, na Alemanha, que possui um conjunto de ferramentas para a resolução de problemas com programas em programação por conjuntos de resposta [11]. Duas ferramentas particularmente úteis para o sistema desenvolvido são o *gringo* [13] e o *clasp*.

O *gringo* permite a tradução de programas lógicos para programas equivalentes e sem variáveis. Posteriormente, os resultados gerados pelo *gringo* podem ser usados no *clasp* [12] para obter os modelos estáveis. Uma característica do *gringo* é a possibilidade de utilizar funções de otimização, dando prioridades e pesos aos literais para que se possa minimizar, ou maximizar, os modelos estáveis, tal como se pode ver no exemplo 3.4 que obtém o melhor entre 5 hotéis, definindo os respetivos valores de peso e prioridade quanto ao número de estrelas, ao custo e ao ruído, sendo que quanto maior for o valor, maior é a prioridade de otimização. Nas primeiras três linhas do programa começa-se por definir os vários hotéis, a quantidade de estrelas e o custo associados a cada um, com as funções *hotel/1*, *star/2* e *cost/2* respetivamente. A quarta define que o hotel 4 se encontra numa rua principal. De seguida, define-se o conceito de hotel com ruído, onde um hotel contém ruído quando se encontra numa rua principal (neste caso o hotel 4). Por fim, são feitas as várias otimizações, em que a prioridade mais baixa é maximizar os hotéis com mais estrelas. Como segunda prioridade, é definido a minimização, mediando do custo e da quantidade de estrelas dos hotéis. A prioridade mais alta é a minimização de hotéis com ruído.

Listing 3.4: Programa lógico com otimização

```

1 { hotel(1..5) } 1.
2 star(1,5). star(2,4). star(3,3). star(4,3). star(5,2).
3 cost(1,170). cost(2,140). cost(3,90). cost(4,75). cost(5,60).
4 main_street(4).
5 noisy :- hotel(X), main_street(X).
6 #maximize [ hotel(X) : star(X,Y) = Y @ 1 ].
7 #minimize [ hotel(X) : cost(X,Y) : star(X,Z) = Y/Z @ 2 ].
8 #minimize { noisy @ 3 }

```

Os resultados do programa estão apresentados no exemplo 3.5. Este programa resulta

⁵<http://potasco.sourceforge.net/>

de 2 modelos alternativos, onde o primeiro é escolhido o hotel 5 com custo de 60 e o segundo é escolhido o hotel 3 com custo de 90, que representa o modelo ótimo devido a ter mais estrelas que o hotel 5.

Listing 3.5: Resultado do programa lógico

```

1 Answer: 1
2 star(1,5) star(2,4) star(3,3) star(4,3) star(5,2) cost(1,170) cost(2,140)
3 cost(3,90) cost(4,75) cost(5,60) main_street(4) hotel(5)
4 Optimization: 0 30 15
5 Answer: 2
6 star(1,5) star(2,4) star(3,3) star(4,3) star(5,2) cost(1,170) cost(2,140)
7 cost(3,90) cost(4,75) cost(5,60) main_street(4) hotel(3)
8 Optimization: 0 30 14
9 OPTIMUM FOUND
10
11 Models      : 1
12   Enumerated: 2
13   Optimum   : yes
14 Optimization: 0 30 14

```

Outra funcionalidade do *gringo* é a possibilidade de utilizar *scripts Lua* nos programas lógicos. Desta forma, pode ser feita uma integração com uma base de dados para recolha de informação. É também atualmente o *solver* mais eficiente. Existem também outras alternativas, tal como o DLV⁶ ou Smodels⁷.

3.3 Java

Hoje em dia, as linguagens imperativas são utilizadas muitas aplicações. O exemplo mais conhecido é a linguagem Java, que foi lançado pela primeira vez em 1995 [28], sendo uma das linguagens imperativas mais desenvolvidas e utilizadas atualmente. É uma linguagem orientada a objetos, que possui uma sintaxe bastante simples e de fácil implementação.

Outras características fornecidas pela linguagem Java são, por exemplo, a facilidade de criação de programas concorrentes e possui um ambiente de execução seguro, em que o código é verificado através de uma máquina virtual própria [30]. Também é possível a interação com qualquer outro sistema de gestão de bases de dados e também com qualquer outro tipo de aplicações [3].

Ao contrário de outras linguagens imperativas, como por exemplo o C++. A linguagem Java possui um mecanismo de gestão de memória, designado de *garbage collector*, que permite a limpeza de memória, removendo objetos que já não estão a ser usados.

O Java também permite a execução de comandos na consola do sistema operativo e efetuar a leitura dos seus resultados. Esta funcionalidade permite a integração de várias

⁶<http://www.dlvsystem.com/>

⁷<http://www.tcs.hut.fi/Software/smodels/>

tecnologias diferentes, transferindo os resultados entre cada uma pela própria consola, em ficheiros de texto, etc.

3.4 Conclusão

Como se pode ver, existem diversas tecnologias que podiam ter sido utilizadas para a realização deste sistema de aconselhamento.

O PostgreSQL é um sistema de gestão de base de dados bastante simples de utilizar e é uma possível escolha para armazenamento de toda a informação necessária para o funcionamento do sistema. Aproveitando os vários índices que dispõe e a possibilidade de customização para o aumento de performance nas várias consultas.

O uso das ontologias foi a opção escolhida para apresentação do conhecimento e o armazenamento dos dados em *Triple Stores*. Desta forma, os dados podem estruturados de forma mais simples, uma vez que a integração da informação necessária é bastante complexo. De entre as várias possibilidades, foi utilizado o RDF(S), uma vez que é uma linguagem simples, mas suficiente para os nossos objetivos, ao contrário do OWL que é mais completo e complexo. Com esta tecnologia, recorreu-se ao PostgreSQL para guardar a informação e efetuou-se a manipulação da informação com consultas em SPARQL. Algumas as informações recolhidas foram feitas a partir de uma recomendação da W3C que converte uma base de dados relacional para RDF [6].

Devido ao modelo de informação do sistema a desenvolver ser bastante complexo, foi escolhido a programação por conjuntos de resposta para o desenvolvimento dos planos de tratamento para a elaboração das regras de Proteção Integrada. Desta forma, é possível gerar os vários planos alternativos, uma vez que os resultados são constituídos por vários modelos estáveis, evitando a implementação de algoritmos específicos para o efeito. Além disso, esta linguagem é mais expressiva que a álgebra relacional, que serve de base ao SQL.

Por fim, o Java foi também uma escolha para este sistema, pois é a tecnologia que efetua toda a integração das ferramentas escolhidas. É a partir desta linguagem que são feitas as leituras dos dados e são armazenados através da API do *Apache Jena*, referido na secção 3.1.2. Além disso, recebe os *inputs* do utilizador e executa os programas em programação por conjuntos de resposta para obtenção dos planos de tratamento e ilustra-os através de uma interface web.

4

Abordagem de Solução

Neste capítulo será apresentada a abordagem de solução do sistema desenvolvido. A seguinte secção apresenta o modelo de informação para os dados a serem armazenados, na secção 4.1. A secção 4.2 descreve a arquitetura do sistema construído, onde se explica a forma como este se encontra organizado. A secção 4.3 expressa as operações possíveis para interagir com o sistema desenvolvido.

4.1 Modelação da informação

A figura 4.1 representa o diagrama de classes com o modelo de informação utilizado para a integração da informação necessária para o funcionamento do sistema, tal como foi descrito na secção 1.2.3. Estes dados são exportados como factos lógicos, na linguagem de programação por conjuntos de resposta, para serem implementadas as restrições necessárias sobre os mesmos para a gestão dos planos de tratamentos.

Neste modelo é armazenada a informação das culturas disponíveis na classe *Culture*, embora apenas se considere a cultura do tomate nesta dissertação. Cada cultura possui um conjunto de estados fenológicos que são representados pela classe *PhenologicalState* onde se encontra o código respetivo, o nome e a duração média do respetivo estado fenológico, em dias.

Os diversos inimigos existentes registam-se como uma doença (*Disease*) ou uma praga (*Plague*), onde ambas herdam da classe *Enemy*. Ambos os inimigos e as culturas herdam da classe *Species*, cujo os seus elementos possuem os nomes vulgares, nomes alternativos e os códigos fornecidos pela EPPO. Algumas espécies possuem um determinado tipo *SpeciesType* (e.g.: Afídeos, Míldio, etc), principalmente os inimigos, que com o nome vulgar respetivo.

Os elementos da classe *CropProblem* servem para indicar os inimigos que afetam as culturas e quais as substâncias ativas que podem ser utilizadas para efetuar o seu tratamento.

As substâncias ativas são representadas através das classes: *ActiveSubstance*, *ActiveSubstanceContainer*, *Formulation*, *SubstanceConcentration* e *Substance*. Cada substância ativa existente nos documentos da DGAV está representada com a classe *ActiveSubstance*, separadas pelo inimigo alvo, isto é, podem existir substâncias repetidas, mas com identificadores diferentes para cada inimigo em que possa ser aplicada, pois em certos casos as regras diferem de um inimigo para outro. Esta classe indica o número máximo de aplicações permitidas por ciclo da cultura, se são consecutivas e o intervalo de segurança. Como as substâncias ativas podem ser combinações de substâncias, os elementos da classe *ActiveSubstanceContainer* têm o propósito de conter quais são as substâncias que pertencem a uma substância ativa, juntamente com o nome e sigla da sua formulação, na classe *Formulation*. As informações complementares de cada substância encontram-se em elementos da classe *SubstanceConcentration*, com as concentrações permitidas para uma substância pertencente a uma substância ativa, e as informações que designam a própria substância como nome e código único, através de elementos da classe *Substance*.

A maioria das substâncias têm um determinado modo de atuar na cultura. Essas informações estão especificadas em elementos da classe *ActionMode* (e.g.: Sistema nervoso, respiração, etc) com o nome designado e um identificador único.

Cada substância ativa pode ocorrer num conjunto de produtos. Cada produto é representado por um elemento da classe *Product* que possui o seu nome comercial, a sua classificação, número de autorização, a persistência, o código único e a sua utilização. Os produtos são classificados por um determinado tipo (e.g., Curativo, Preventivo, etc), especificado com elementos da classe *ProductType*, com o nome e identificador.

Para efeitos de otimização de custos de um plano de tratamento, cada produto possui um determinado preço unitário, através de elementos da classe *ProductPrice*, e as doses recomendadas de aplicação, por hectare, com objetos da classe *ProductDoses*. Para cada elemento *ProductPrice*, o preço unitário diz respeito a um pacote, de um determinado produto, com uma certa capacidade. Por outro lado, as doses de um determinado produto podem variar consoante o inimigo a tratar. O custo total é calculado pela dose necessária, juntamente com o preço do produto indicado no plano.

Ambos os produtos e as substâncias possuem categorias que são membros da classe *Category*, com um nome designado para o efeito (e.g., fungicida, herbicida, inseticida, etc).

A classe *Company* representa elementos que registam as diversas empresas fornecedoras de produtos fitofarmacêuticos. Cada uma é designada pelo seu nome, sigla e página *web*. Cada produto é comercializado por uma ou mais empresas existentes.

4.2 Arquitetura do sistema

A arquitetura proposta do sistema a desenvolver encontra-se na figura 4.2 que requer várias fontes de informação que serão integradas, como foi apresentado na secção 1.2.3. Para a integração foram recolhidas informações em vários formatos diferentes de cada fonte indicada na figura e colocada no modelo de dados indicado na secção anterior.

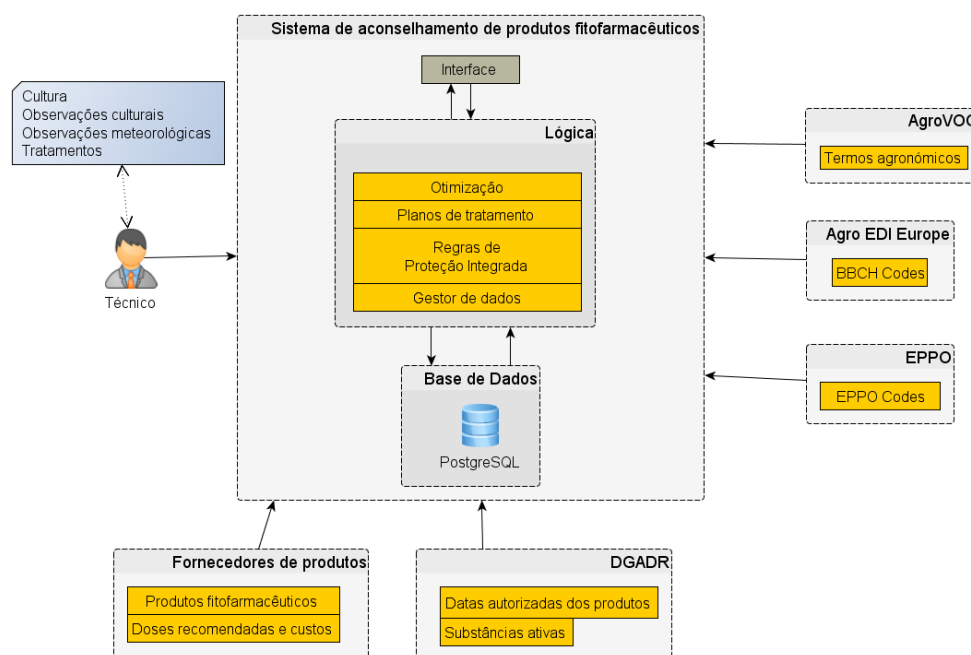


Figura 4.2: Arquitetura do sistema

Os fornecedores de produtos, mencionados na secção 2.8.3 disponibilizam dados relativos aos seus produtos fitofarmacêuticos com as respetivas descrições dos mesmos. Esta informação está contida textualmente nas próprias páginas *web* das empresas fornecedoras.

As informações recolhidas da DGADR incluem os inimigos existentes que afetam a cultura do tomate e as substâncias ativas com as respetivas propriedades e regras de utilização. Todos estes dados encontram-se em ficheiros PDF.

Os códigos BBCH, gerados pelo Agro EDI Europe, que foram obtidos em documentos específicos e indicados na secção 2.5, serviram para identificar os estados fenológicos das culturas em formato eletrónico. Muitos deles não foram colocados, uma vez que os estados não foram armazenados de forma tão detalhada como os que se encontram nestes documentos.

Os identificadores relativos aos organismos foram obtidos através do EPPO. Para cada organismo foi consultado o seu código, bem como o diversos nomes alternativos, na ferramenta EPPT, indicada na secção 2.6. Esta organização também disponibiliza a informação em XML aos utilizadores registados. Para o registo é necessário um pagamento anual.

De forma a obter a informação necessária, foram descarregados os termos agronómicos contidos no AgroVOC, nomeadamente os nomes dos vários inimigos para a cultura do tomate. Estes termos estão disponibilizados em diversos formatos, como visto na secção 2.3, entre os quais será utilizado o formato RDF/XML.

Juntamente com as informações obtidas, o utilizador terá de fornecer alguns *inputs* para o sistema gerar um plano de tratamento conforme as suas necessidades. É preciso indicar a cultura, o seu local, se está instalada em estufa ou ao ar livre, e o seu objetivo, se é para consumo ou para fins industriais. Posteriormente, terá de colocar as observações de pragas/doenças na cultura, os dias respetivos e o estado do tempo, para serem gerados tratamentos curativos respetivos ao inimigo a tratar. Por fim, pode indicar se quer começar a efetuar tratamentos preventivos para um determinado inimigo.

A aplicação desenvolvida está estruturada em três camadas: interface, lógica e base de dados. Na camada base de dados encontram-se os dados providenciados pelas fontes indicadas anteriormente. A tecnologia a ser utilizada para este efeito será o PostgreSQL, pelos motivos mencionados na secção 3.4. A informação contida nesta camada será enviada para a camada lógica, após ser requisitada.

A camada lógica possui quatro componentes: uma de gestão de dados, uma para as regras de Proteção Integrada, uma para os planos de tratamento e uma de otimização. O gestor de dados é o responsável pela manipulação da informação que se encontra na base de dados/conhecimento. É neste componente que foram inseridos os dados em triplos RDF, mencionados anteriormente, através da API do *Apache Jena*. Os dados armazenados são enviados, sob a forma de factos lógico, para as camadas seguintes. A componente RPI representa as regras de Proteção Integrada, estando estas regras capturadas na linguagem ASP. O componente de planos de tratamento irá recorrer às regras anteriores para instanciar os vários planos de tratamento válidos. Seguidamente, recorrendo à componente de otimização, estes planos serão restringidos aos planos de tratamentos mais económicos consoante os fatores mencionados na secção 1.2.2. As otimizações existentes exigem que o utilizador designe a importância de cada uma. Estas componentes foram desenvolvidas em programação por conjuntos de resposta, especificada na secção 3.2.

Por fim, a camada de interface contém todas as funcionalidades para a interação com o técnico agrícola que estão mencionadas na secção 4.3. É nesta camada que são apresentados os planos de tratamento, resultantes da camada anterior. A sua implementação foi realizada em Java, constituindo uma interface *web* simples para usufruir das funcionalidades deste sistema.

4.3 Funcionalidades do sistema

O sistema desenvolvido oferece um conjunto de funcionalidades simples para o utilizador, uma vez que a sua complexidade está na camada lógica, como foi indicado na secção 4.2, exigindo, maioritariamente, inserções de dados para que permita ao sistema efetuar a gestão de um plano de tratamento. A figura 4.3 trata-se de um diagrama de

casos de uso que ilustra as diversas operações existentes.

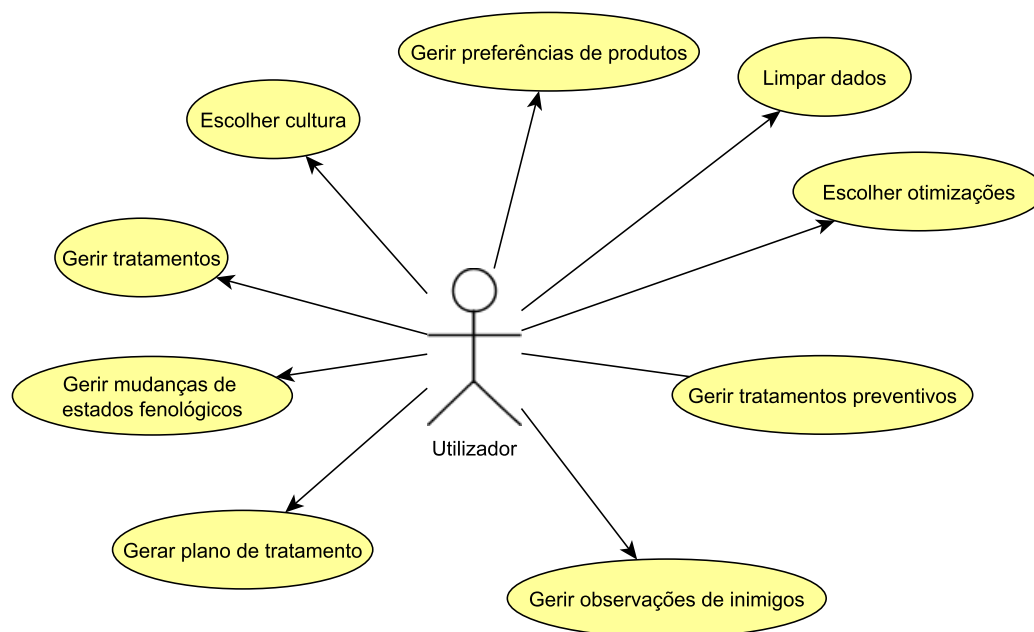


Figura 4.3: Diagrama de casos de uso

Inicialmente, o utilizador introduz um conjunto de *inputs* antes de indicar ao sistema para efetuar a gestão do plano. Desta forma, existe a funcionalidade `Escolher cultura` para indicar a cultura em questão, embora esta dissertação apenas se debruce sobre a cultura do tomate. Após a cultura ser escolhida, o utilizador pode começar por `Gerir observações de inimigos`, para indicar os inimigos que estão a afetar a cultura. Também pode `Gerir mudanças de estados fenológicos` para os casos em que a cultura tenha passado de um estado para outro de forma fora do tempo previsto. É necessário a indicação de otimizações, através da funcionalidade `Escolher otimizações`, para que o sistema saiba o que é necessário otimizar. Opcionalmente, o utilizador pode `Gerir preferências de produtos` para forçar o sistema a gerar um plano com os produtos indicados. Além disso, pode `Gerir tratamentos preventivos`, indicando ao sistema de que deseja começar um tratamento preventivo para um determinado inimigo.

Estes dados são suficientes para o utilizador indicar ao sistema para gerar um plano de tratamentos, através da funcionalidade `Gerar plano de tratamentos`. Aqui são enviados os diversos *inputs* para a gestão de um plano de tratamentos.

Após o plano inicial se encontrar criado, o utilizador pode `Gerir tratamentos`, tanto preventivos, como curativos, para que o sistema saiba que produtos foram utilizados nas aplicações que efetuou.

Se o utilizador desejar criar um novo plano de tratamento diferente do anterior, existe a funcionalidade de `Limpar dados` que limpa todos os dados inseridos de forma mais simples, isto faz com que o utilizador não tenha de apagar os dados individualmente.



Implementação

Neste capítulo serão descritos os métodos utilizados para o desenvolvimento de vários componentes que constituem o sistema proposto. Estes componentes foram listados na secção 4.2, juntamente com as tecnologias que são utilizadas para o funcionamento das mesmas.

A primeira secção explica a forma como os dados são inseridos no modelo de dados, demonstrado na secção 4.1. Seguidamente, na secção 5.2 são indicadas as consultas efetuadas aos dados a serem traduzidos para um ficheiro com factos lógicos que, posteriormente, são utilizados na geração dos planos de tratamento, explicado na secção 5.3. Na secção 5.4 resume-se o modo como são expressas as regras de proteção integrada para que sejam gerados planos de tratamento válidos. A otimização aos modelos obtidos é descrita na secção 5.6. A interface para o utilizador interagir com o sistema é apresentada na secção 5.7. Finalmente na secção 5.8, retiram-se conclusões acerca de todo o trabalho realizado.

5.1 Carregamento de dados

Uma das tarefas principais desta dissertação foi a recolha da informação necessária para o funcionamento do sistema implementado. São estes dados que constituem a base da arquitetura indicada na secção 4.2.

Inicialmente foram recolhidas informações dos organismos, tanto dos inimigos, como da própria cultura do tomate. Através de documentos da DGADR [32] foi obtida a lista dos inimigos que afetam a cultura do tomate, bem como os nomes vulgares, os nomes científicos e os códigos EPPO. Algumas informações adicionais, como referências a páginas *web* com explicação do organismo ou níveis taxonómicos, foram descarregadas do

AgroVOC, descrito na secção 2.3, através do *endpoint* com consultas SPARQL para obter os triplos respetivos. O exemplo 5.1 representa uma das consultas efetuadas ao *endpoint* para devolver os triplos relativos ao organismo *Gryllotalpa gryllotalpa*, que se trata de uma praga, designada de ralo, que afeta a cultura do tomate. A primeira condição do `WHERE` indica que o sujeito de cada triplo tem de ser `agrovoc:c_3404`, tratando-se do identificador do organismo que se pretende, enquanto que o filtro seguinte à condição serve para obter as propriedades desejadas, para reduzir os resultados. Por fim, os nomes são obtidos em português ou inglês com a função `lang`.

Listing 5.1: Exemplo de consulta SPARQL ao *endpoint* do *AgroVOC*

```

1 CONSTRUCT { agrovoc:c_3404 ?p ?o }
2 WHERE
3 {{
4   agrovoc:c_3404 ?p ?o.
5   }
6
7   FILTER((?p = skos:prefLabel && (lang(?o)="pt" || lang(?o)="en"))
8     || (?p = skos:altLabel && (lang(?o)="pt" || lang(?o)="en"))
9     || ?p=skos:exactMatch
10    || ?p=agront:hasTaxonomicLevel
11    || ?p=skos:closeMatch
12   )
13 }
```

A figura 5.1 ilustra os resultados que foram obtidos com a consulta referida anteriormente através do *endpoint* do *AgroVOC*. Estes resultados foram descarregados em formato N-Triples.

Subject	Predicate	Object
agrovoc:c_3404	skos:closeMatch	Gryllotalpa
agrovoc:c_3404	skos:exactMatch	44032
agrovoc:c_3404	skos:exactMatch	4379384-8
agrovoc:c_3404	skos:altLabel	"Ralo (insecto)"
agrovoc:c_3404	skos:altLabel	"Mole crickets"
agrovoc:c_3404	skos:prefLabel	"Gryllotalpa"
agrovoc:c_3404	skos:prefLabel	"Gryllotalpa"
agrovoc:c_3404	agront:hasTaxonomicLevel	agrovoc:c_11125
agrovoc:c_3404	skos:broader	agrovoc:c_31939

Figura 5.1: Resultados de uma consulta no *endpoint* do *AgroVOC*

Seguidamente, através de uma análise a documentos disponibilizados pela DGAV [8], efetuou-se, manualmente, uma listagem dos nomes de substâncias utilizadas na cultura do tomate, em português e inglês. A todas as substâncias adicionadas foram atribuídos os identificadores químicos únicos, designados de número CAS, que foram retirados de páginas *web* com informações acerca de substâncias químicas. Além da listagem das

substâncias, também se obteve, nos mesmos documentos, as informações das substâncias ativas e dos produtos necessários para o tratamento do tomate.

Algumas informações adicionais relativas aos produtos já se encontravam numa base de dados, em PostgreSQL, previamente construída. No entanto, informações complementares como persistências, preços e tipos de produtos (e.g., preventivo, curativo, etc) foram fornecidas mais tarde e introduzidas manualmente.

Toda a informação recolhida foi transformada para um único ficheiro, sob a forma de triplos ontológicos, em formato N3, organizada da mesma forma que o modelo de dados que se encontra na secção 4.1. Os dados que se encontram neste ficheiro são armazenados num *Triple Store*, criado no PostgreSQL, através do *Apache Jena*. Para usufruir das funcionalidades desta tecnologia, é utilizado a API *Java* que esta dispõe, tal como se pode ver no pseudo-código 5.2. Nesta listagem, as primeiras 5 linhas cria uma ligação ao PostgreSQL, onde o objeto `conn` é uma variável global que possui a função `getStore` para obter a ligação ao *Triple Store*. Caso a base de dados já se encontre criada e preenchida, então os dados existentes são limpos para serem atualizados na sua totalidade, pois no caso de alguns dados terem sido eliminados no ficheiro, estes não ficam armazenados. Na linha 12, após a criação das tabelas necessárias para o *Triple Store*, é feita uma leitura do ficheiro com os dados e armazenados, simultaneamente, tal como se pode ver desde a linha 15 à 19. Feitas as inserções, as duas últimas linhas fecham as ligações à base de dados.

Listing 5.2: Inserção dos dados em *Java* com o *Apache Jena*

```
1 // Ligações
2 Store store = conn.getStore();
3 Model model = SDBFactory.connectDefaultModel(store);
4
5 JDBC.loadDriver(JDBC_DRIVER);
6
7 // Criação das tabelas para a inserção
8 // caso ainda não estejam criadas/formatadas de forma correta
9 if (StoreUtils.isFormatted(store))
10     store.getTableFormatter().truncate();
11
12 store.getTableFormatter().create();
13
14 // Leitura do ficheiro e inserção dos dados
15 InputStream in = FileManager.get().open(filepath);
16 if (in == null)
17     throw new IllegalArgumentException("Unable_to_read_the_file.");
18
19 model.read(in, "http://www.saapf.pt/", fileformat);
20
21 // Fecho das ligações
22 model.close();
23 conn.closeConnection();
```

5.2 Geração de factos

Tal como foi referido na secção anterior, muitos dos dados que se encontram armazenados, no *Triple Store* criado, necessitam de ser transformados em factos lógicos, num ficheiro, para que estes sejam processados através da linguagem de programação por conjuntos de resposta, explicada na secção 3.2.

Devido às consultas SPARQL terem de ser feitas através da API Java do *Apache Jena*, utilizou-se o Java para obter a informação essencial e escrever para um ficheiro sob a forma de factos lógicos.

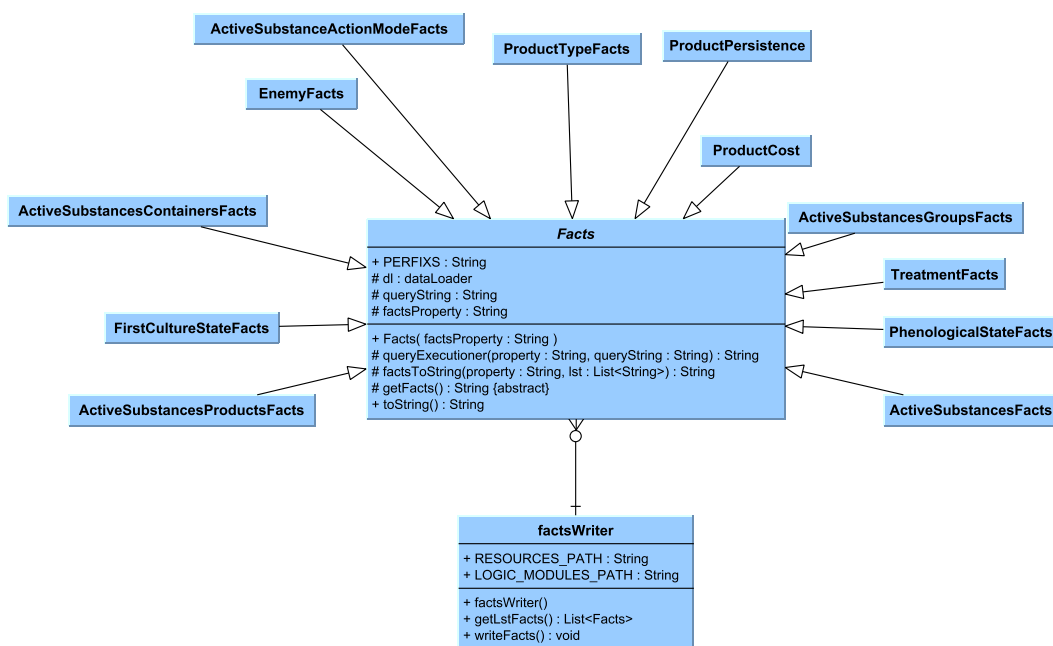


Figura 5.2: Diagrama de classes para a escrita de factos lógicos

Sendo necessário realizar muitas consultas ao *Triple Store*, criou-se um conjunto de classes Java que estão representadas no diagrama de classes da figura 5.2. A classe *Facts* é uma classe abstrata que contém os atributos e métodos a serem utilizados pelas classes que a herdam. Esta classe possui os atributos *PREFIXS* com todos os prefixos que constituem *URIs* a serem usados nas consultas, o *dl* que se trata de um objeto para efetuar as ligações ao *Triple Store*, a *queryString* com a consulta a efetuar à base de dados, podendo-se usar outra diferente, e o *factsProperty* que é o nome da propriedade dos factos resultantes de cada classe. O construtor desta classe apenas recebe o valor para *factsProperty*, uma vez que cada classe, que herda de *Facts*, define a sua *queryString* e o *dl* trata-se de um objeto único a todo o sistema. Além dos atributos, existem os métodos. O método *queryExecutioner* recebe duas *Strings*: a primeira com o nome do predicado de cada facto resultante e a segunda com a consulta a enviar para o *Triple Store*. Este método serve para efetuar a consulta ao *Triple Store* e devolver todos os resultados sob a forma de factos, numa só *String*, com o auxílio do *factsToString*

que recebe o nome do predicado para os factos e uma lista com o conjunto de átomos respetivos, devolvendo um facto escrito com a sintaxe correta (*clasp*). Todas as classes que herdam de `Facts` têm de implementar o método `getFacts`, que usufrui dos outros métodos anteriormente mencionados e devolve uma *String* que será escrita no ficheiro final. O método `toString` apenas chama o `getFacts`.

As classes que foram criadas, e que herdam da classe `Facts`, são as seguintes:

- **Nome:** `EnemyFacts`

Descrição: Gera todos os factos relativos aos inimigos das culturas. Cada facto possui o código EPPO, o nome e o tipo dos inimigos.

Exemplo: `enemy("PHYTIN", "Phytophthora infestans", "Míldio").`

- **Nome:** `TreatmentFacts`

Descrição: Obtém os factos com os tratamentos possíveis das culturas. Cada facto possui o código EPPO da cultura e do inimigo e o número da substância ativa que é possível ser utilizada para o tratamento.

Exemplo: `possible_treatment("LYPES", "PHYTIN", 67).`

- **Nome:** `PhenologicalStateFacts`

Descrição: Devolve os factos com todos os estados fenológicos das culturas. Cada facto possui um identificador do estado, o seu nome, a cultura respetiva e a duração em dias.

Exemplo: `phenological_state("E", "LYPES", "Floração", 14).`

- **Nome:** `FirstCultureStateFacts`

Descrição: Esta classe é um auxiliar que serve para indicar qual o estado inicial de cada cultura. Cada facto possui o identificador EPPO da cultura e o identificador do estado inicial da mesma.

Exemplo: `start_state("LYPES", "A").`

- **Nome:** `ProductPersistence`

Descrição: Devolve os dias que os produtos persistem nas culturas. Cada facto possui o nome do produto e o número de dias que perdura.

Exemplo: `prod_persistence("TOR", 10).`

- **Nome:** `ProductCost`

Descrição: Apresenta os custos dos produtos, juntamente com a margem. Cada resultado imprime dois factos diferentes: um com o custo unitário e outro com a margem relativa ao custo. Ambos os factos possuem o nome do produto respetivo.

Exemplos:

```
product_cost("TOR", 64, 2700) .  
product_margin("TOR", 64, 270) .
```

- **Nome:** `ProductTypeFacts`

Descrição: Obtém os tipos dos vários produtos, para se saber quais são os produtos que se podem usar para tratamentos preventivos e os que não se podem. Cada facto possui o identificador do produto, o respetivo nome e o seu tipo. No caso de vários tipos, resulta um facto para cada tipo.

Exemplo: `prod_type(327, "TOR", "Preventivo") .`

- **Nome:** `ActiveSubstancesFacts`

Descrição: Possui uma listagem das substâncias ativas que se encontram armazenadas. Cada facto possui o identificador da substância ativa e o seu nome.

Exemplo: `activesub(40, clortalonil) .`

- **Nome:** `ActiveSubstancesContainersFacts`

Descrição: Como cada substância ativa é um conjunto de uma ou mais substâncias, esta classe indica as substâncias que se encontram em cada substância ativa. Cada facto possui o identificador da substância ativa e o número CAS da substância. No caso de mais do que uma, escreve um facto para cada.

Exemplo: `activesub_container(40, "SC", "1897-45-6") .`

- **Nome:** `ActiveSubstancesGroupsFacts`

Descrição: Esta classe devolve a família química das substâncias ativas, onde apenas as que são necessárias se encontram armazenadas no *Triple Store*. Cada facto possui o identificador da substância ativa, o identificador e o nome da família química.

Exemplo: `activesub_group(44, 3, "Ditiocarbamatos") .`

- **Nome:** `ActiveSubstanceActionModeFacts`

Descrição: Obtém os modos de ação das substâncias ativas que se encontram armazenadas. Cada facto possui o identificador da substância ativa e o nome do modo de ação.

Exemplo: `activesub_actmode(39, 3, "Respiração") .`

- **Nome:** `ActiveSubstancesProductsFacts`

Descrição: Devolve os produtos de cada substância ativa. Cada facto possui o identificador da substância ativa e o nome do produto.

Exemplo: `activesub_product(40, "BRA 500") .`

Com todas as classes criadas e definidas, estas são armazenadas na classe `factsWriter` que contém uma lista com todas elas. Esta classe tem o objetivo de iterar a lista e escrever os factos resultantes de todas as classes no ficheiro final, através do método `toString`.

Este processo é feito cada vez que há uma atualização aos dados existentes no *Triple Store*, pois a informação que se encontra armazenada é estática e o ficheiro, para o qual são escritos os factos lógicos, pode ser sempre reaproveitado para a geração de novos planos de tratamento.

5.3 Geração de tratamentos

Após os dados essenciais estarem sob a forma de factos lógicos, pode-se passar à geração dos planos de tratamentos. Nesta secção serão apresentadas os dois tipos de tratamentos que podem ser construídos pela aplicação, juntamente com o modo como é criado o calendário com a duração do ciclo cultural e a forma como são escolhidos os produtos a serem utilizados nos tratamentos.

5.3.1 Ciclo cultural

Os tratamentos efetuados nas culturas não dependem só do inimigo que se pretender tratar. Também é preciso ter em conta o estado fenológico em que a cultura se encontra, juntamente com outros fatores que serão mencionados na secção 5.4.

Como cada estado fenológico possui uma duração estimada em dias, é possível estimar a totalidade do ciclo cultural através do somatório da duração de todos os estados de uma determinada cultura. Com isto, o calendário de tratamentos encontra-se organizado em dias, sendo que a cada dia é atribuído o estado em que se encontra. No entanto, as culturas não se desenvolvem de forma regular e, por vezes, é necessário a indicação, por parte do utilizador, de uma observação que especifique quando a cultura mudou de estado numa altura fora do esperado.

Estas indicações registadas pelo utilizado na interface da aplicação que se traduzem depois em factos lógicos, como se pode ver na listagem 5.3. Cada `state_observation` representa uma indicação de uma mudança de estado, fora do tempo pré-determinado, onde cada um possui o dia em que foi observado a mudança, a cultura que foi observada, identificada com o código EPPO, e o estado em que se encontra, representado com uma letra. A atribuição de letras aos estados fenológicos serve para que seja, mais facilmente, identificado a ordem dos estados desde início até ao fim do ciclo cultural, cujo a ordem se define através da ordem alfabética.

Listing 5.3: *Inputs* das observações do estados fenológicos

```

1 state_observation(1, "LYPES", "A") .
2 state_observation(61, "LYPES", "B") .
3 state_observation(75, "LYPES", "C") .

```

Estas observações são *inputs* opcionais que irão afetar o número total de dias do ciclo de desenvolvimento de uma determinada cultura, estimadas pelo sistema. Estes dias são gerados pelo conjunto de regras que se podem observar na listagem 5.4. As primeiras quatro regras são auxiliares para serem utilizadas pelas outras. A primeira auxiliar, na linha 1, obtém os estados de uma determinada cultura por projeção do predicado *phenological_state*. Seguidamente, a segunda, que se encontra na linha 4 do exemplo, indica que existe um *input* para um determinado estado por parte do utilizador. Na linha 7, encontra-se a terceira auxiliar que captura a relação de ordem consecutiva dos estados de uma cultura. A última auxiliar, na linha 13, indica se entre dois estados fenológicos existe um outro.

As regras *state_expected* servem para indicar os dias iniciais previsíveis de cada estado, tendo em conta os *inputs* mencionados anteriormente. O primeiro atribui o dia inicial de um estado, caso exista uma observação que assim o indique. Caso contrário, são utilizados os *state_expected* seguintes, onde um atribui o primeiro dia de uma cultura, caso não haja uma observação para tal, e o outro indica o seu dia final.

Por fim, todas as regras seguintes irão criar todos os dias do ciclo cultural desde o primeiro dia do primeiro estado fenológico até ao último do último estado fenológico. Assim, existirá uma propriedade *state* que indica, para um determinado dia, o estado em que a cultura se encontra, tanto estimado, como observado.

Listing 5.4: Geração do ciclo cultural

```

1 state(State,Culture) :-
2   phenological_state(State,Culture,_,_) .
3
4 state_observed(State,Culture) :-
5   state_observation(_,Culture,State) .
6
7 state_before(State1,State2,Culture) :-
8   state(State1,Culture) ,
9   state(State2,Culture) ,
10  State1 < State2 ,
11  not state_between(State1,State2,Culture) .
12
13 state_between(State1,State2,Culture) :-
14  state(State1,Culture) ,
15  state(State2,Culture) ,
16  state(State,Culture) ,
17  State1 < State , State < State2 .
18
19 state_expected(State,Culture,Day) :-
20  state_observation(Day,Culture,State) .

```

```

21 state_expected(State,Culture,1) :-
22     start_state(Culture,State),
23     not state_observed(State,Culture).
24
25 state_expected(State,Culture,Day+Duration) :-
26     state(State,Culture),
27     not state_observed(State,Culture),
28     state_before(Statel,State,Culture),
29     phenological_state(State,Culture,_,Duration),
30     state_expected(Statel,Culture,Day).
31
32 start_end_state(State,Culture,Start_day,Finish-1) :-
33     state_expected(State,Culture,Start_day),
34     state_expected(Statel,Culture,Finish),
35     Start_day < Finish,
36     not state_between(State,Statel,Culture).
37
38 has_start_end_state(State,Culture) :-
39     start_end_state(State,Culture,_,_).
40
41 state(Start_day..Finish,Culture,State) :-
42     start_end_state(State,Culture,Start_day,Finish).
43
44 state(Start_day..Finish,Culture,State) :-
45     state_expected(State,Culture,Start_day),
46     not has_start_end_state(State,Culture),
47     phenological_state(State,Culture,_,Duration),
48     Finish := Start_day + Duration.

```

5.3.2 Tratamentos curativos

Os tratamentos curativos são tratamentos a efetuar quando há uma observação de uma praga ou doença que está a afetar uma cultura. Para isso é necessário que o utilizador introduza esse tipo de observações para que sejam propostos os produtos fitofarmacêuticos possíveis de serem utilizados.

A lista de factos em 5.5 representa um conjunto de observações resultantes dos *inputs* do utilizador. Cada facto possui o dia da observação, os códigos EPPO da cultura e do organismo observado e o estado de desenvolvimento do problema, que pode ser *weak* ou *strong*. O estado *weak* significa que a praga ainda se encontra no estado inicial/fraco, enquanto que o estado *strong* representa a praga numa fase avançada.

Listing 5.5: *Inputs* das observações de pragas

```

1 enemy_observation(2,"LYPES","LAPHEG",weak).
2 enemy_observation(3,"LYPES","PLUSCH",weak).
3 enemy_observation(4,"LYPES","PLUSCH",strong).
4 enemy_observation(13,"LYPES","PLUSCH",strong).
5 enemy_observation(14,"LYPES","LAPHEG",strong).

```

Com estas observações, foi criada uma regra de escolha, como se pode observar no código 5.6, na linha 6, que atribui uma, e uma só, substância ativa a ser utilizada, dentro das que são possíveis para o inimigo alvo e que possuem produtos registados no *Triple Store*, pois algumas substâncias ativas podem não ter produtos correspondentes. A outra regra serve como auxiliar para indicar que existe uma determinada observação, para um certo dia, de um determinado tipo de inimigo (e.g.: Míldio, Tripes, etc).

Listing 5.6: Gerador de tratamentos curativos

```

1 infestation(Day,EType) :-
2   enemy_observation(Day,Culture,Organism,_),
3   enemy(Organism,_,EType),
4   culture(Culture,_,_,_).
5
6 1 {
7   treat_with_actsub(D,C,A) : possible_treatment(C,E,A) : activesub_has_prod(A)
8 } 1 :-
9   culture(C,_,_,_),
10  infestation(D,EType),
11  enemy(E,_,EType).

```

Com o código 5.6 será gerada uma substância ativa a ser utilizada para o tratamento do inimigo, representado pela propriedade `treat_with_actsub`, onde o primeiro argumento é o dia a ser aplicado, o segundo é a cultura alvo e o terceiro é o identificador da substância ativa.

Através deste predicado, terá de ser atribuído um, e um só, produto a ser utilizado na cultura alvo. Foi então criada uma outra regra de escolha para esse efeito, que se pode ver em 5.7, onde é gerado um facto com predicado `treat_with_product`. Este facto tem como primeiro argumento o dia a ser utilizado, o segundo é a cultura alvo, o terceiro é o nome do produto e o último é a substância ativa do produto correspondente.

Listing 5.7: Gerador de produtos para os tratamentos curativos

```

1 1 { treat_with_product(D,C,P,A) : activesub_product(A,P) } 1 :-
2   culture(C,_,_,_),
3   treat_with_actsub(D,C,A).

```

5.3.3 Tratamentos Preventivos

Os tratamentos preventivos, por outro lado, são tratamentos que servem para prevenir que as culturas sejam infetadas por um determinado tipo de inimigo.

O modo como se geram este tipo de tratamentos é diferente dos tratamentos curativos, mencionados na secção anterior, pois é necessário que o utilizador indique quando quer começar a pô-los em prática. Estes *inputs* serão traduzidos em factos com a propriedades `start_prevent_treatment`, tal como estão listados em 5.8, onde o primeiro argumento é o dia em que quer efetuar o tratamento, o segundo e o terceiro são a cultura

e o inimigo alvo, respetivamente, o quarto indica se as condições ambientais estão favoráveis/desfavoráveis ao desenvolvimento da cultura e o último serve para mencionar ao sistema se o utilizador deseja continuar com o tratamento preventivo, para que o sistema detete o próximo tratamento a partir do dia referido (com o valor 1), ou não (com o valor 0).

Após o tratamento ser aplicado, é necessário que o utilizador indique o produto que foi utilizado. Caso existam previsões de tratamentos, é necessário introduzir um novo `start_prevent_treatment` no dia em que foi previsto pelo sistema, senão é considerado um modelo inconsistente. Este *input* é traduzido para o facto `preventive_treat_prod`, cujo o primeiro argumento é o dia em que foi utilizado, que tem de ser o mesmo que o `start_prevent_treatment` respetivo, o segundo e o terceiro são a cultura e o inimigo alvo, respetivamente, e o último é o nome do produto utilizado.

Listing 5.8: *Inputs* para os tratamentos preventivos

```

1 start_prevent_treatment(20, "LYPES", "Mildio", "favoraveis", 1).
2 start_prevent_treatment(20, "LYPES", "Bacteriose", "favoraveis", 1).
3 start_prevent_treatment(42, "LYPES", "Bacteriose", "desfavoraveis", 1).
4
5 preventive_treat_prod(20, "LYPES", "Bacteriose", "CUP").

```

Tal como para os tratamentos curativos, existem regras próprias para a geração dos tratamentos preventivos, que se podem observar em 5.9. A primeira instrução é a geradora dos tratamentos a efetuar a partir dos `start_prevent_treatment` de uma dada cultura e inimigo alvo. As duas regras seguintes geram as previsões dos próximos tratamentos, caso o utilizador assim o deseje, através da persistência do produto utilizado. Se as condições forem declaradas como “favoráveis”, então será utilizada a persistência máxima do produto. Caso contrário, é utilizada a persistência mínima.

Listing 5.9: Gerador de tratamentos preventivos

```

1 1{
2 preventive_treatment (Day, Cult, EType, Actsub)
3 : possible_preventive_treatment (Cult, EType, Actsub)
4 : can_treat_preventive (EType)
5 } 1 :-
6   start_prevent_treatment (Day, Cult, EType, _, _)
7
8 1{
9 preventive_treatment (Day+Persistence_Days+1, Cult, EType, Actsub)
10 : possible_preventive_treatment (Cult, EType, Actsub) } 1 :-
11   start_prevent_treatment (Day, Cult, EType, "favoraveis", 1),
12   preventive_treat_prod (Day, Cult, EType, Prod),
13   day (Day+Persistence_Days+1, Cult),
14   max_persistence (Prod, Persistence_Days).
15
16

```

```

17 1{
18 preventive_treatment (Day+Persistence_Days+1,Cult,EType,Actsub)
19 : possible_preventive_treatment (Cult,EType,Actsub) } 1 :-
20   start_prevent_treatment (Day,Cult,EType,"desfavoraveis",1),
21   preventive_treat_prod (Day,Cult,EType,Prod),
22   day (Day+Persistence_Days+1,Cult),
23   min_persistence (Prod,Persistence_Days) .

```

Após os tratamentos estarem criados, é necessário aconselhar que produto se deve utilizar. Para isso as regras descritas em 5.10 atribuem um, e um só, produto para cada tratamento gerado. A primeira regra indica o produto a ser utilizado, enquanto que a segunda deduz factos com propriedade `treat_with_product` utilizados nos tratamentos curativos, para a integração de ambos os tipos de planos..

Listing 5.10: Gerador de produtos para tratamentos preventivos

```

1 1{preventive_treat_prod (Day,Cult,EType,Prod)
2 : activesub_product (Actsub,Prod)
3 : preventive_prod (Prod) } 1 :-
4   preventive_treatment (Day,Cult,EType,Actsub) .
5
6 treat_with_product (Day,Cult,Prod,Actsub) :-
7   preventive_treat_prod (Day,Cult,_,Prod),
8   activesub_product (Actsub,Prod) .

```

5.4 Regras de Proteção Integrada

A geração de tratamentos, explicada na secção anterior, resulta em muitos modelos alternativos. No entanto, alguns deles não observam as regras de Proteção Integrada. É essencial a implementação destas regras para que os modelos inválidos sejam retirados dos resultados finais.

Nesta secção será explicado o modo como se representam os tipos de regras existentes, que foram indicados na secção 1.2.1. A cada tipo, são ilustrados pequenas partes do código, em programação por conjuntos de resposta, pois foram implementadas cerca de 200 regras lógicas para a cultura do tomate, juntamente com as auxiliares, e não é possível apresentá-las todas neste documento.

5.4.1 Regras de aplicação

As regras de aplicação são as que indicam o momento em que os tratamentos, com determinada substância ativa, devem ser aplicados, nomeadamente o estado fenológico em que a cultura se encontra.

O código 5.11 representa duas restrições de aplicação, onde a primeira impede que a substância ativa com identificador 76 (*folpete+iprovalicarbe*) seja aplicada após o estado “E” (Floração). Graças à atribuição de letras nos estados fenológicos, basta criar apenas

uma restrição com o operador de comparação `>`, pois a ordem dos estados é identificado pela ordem alfabética. A segunda regra impede que a substância ativa 34 (*clorpirifos*) possa ser aplicada num estado diferente ao "A" (Viveiro).

Listing 5.11: Restrições de aplicação

```
1 :- treat_with_actsub(Day, "LYPES", 76), state(Day, "LYPES", State), State > "E".
2 :- treat_with_actsub(Day, "LYPES", 34), not state(Day, "LYPES", "A").
```

5.4.2 Número máximo de aplicações

A maioria das substâncias ativas possuem um número limitado de aplicações possíveis de efetuar durante todo o ciclo cultural. Isto obriga a que o agricultor tenha de aplicar uma substância ativa diferente, caso já tenha utilizado várias vezes outras.

Para construir restrições deste tipo, é necessário criar um facto que indique o número de vezes que pode ser aplicada uma substância ativa, do mesmo modo que o exemplo 3.3 na secção 3.2. Tal como se pode ver nas regras em 5.12, o primeiro facto indica que a substância ativa 14 (*clorantraniliprol*), utilizado no combate às lagartas, só pode ser aplicado no mínimo 0 vezes e no máximo 2 vezes. O segundo facto serve para o caso em que a substância ativa tenha mais do que um identificador, tal como foi descrito na secção 4.1, pois cada identificador de uma substância ativa também serve para identificar o inimigo a combater, como é o caso do *indoxacarbe*.

Listing 5.12: Número máximo de aplicações de uma substância ativa

```
1 0{ treat_with_actsub(D, "LYPES", 14) : day(D, "LYPES") }2.
2
3 0 { treat_with_actsub(D, "LYPES", Actsub) : activesub(Actsub, indoxacarbe)
4   : day(D, "LYPES") } 4.
```

5.4.3 Regras de restrições temporais

Sendo a colheita o momento em que é a cultura é retirada do seu local onde se desenvolve e é enviada para o seu destino, é importante que não existam resíduos dos produtos que foram aplicados nos tratamentos da mesma. Para isso, a maioria das substâncias ativas possuem um intervalo de segurança, que indica o número de dias antes da colheita em que não pode ser aplicada determinada substância ativa.

Como não é possível saber a data exata em que será feita a colheita, é necessário que o utilizador indique, por meio de *input*, o momento previsto que a cultura será colhida. Este *input* é traduzido para um facto lógico que está descrito em 5.13 com a propriedade *harvest_observation*, onde o primeiro argumento indica o dia que está prevista a colheita e o segundo argumento representa a cultura respetiva com o código EPPO.

Listing 5.13: *Input* da previsão de colheita

```
1 harvest_observation(181, "LYPES").
```

Através deste *input* é possível a criação de regras para representar o intervalo de segurança. O código 5.14 ilustra uma regra auxiliar, na primeira linha, que indica, para um determinado dia e cultura, quantos dias restam até data de à colheita prevista. Esta regra é utilizada para a representação de intervalos de segurança, do mesmo modo que a segunda regra do exemplo. Nesta regra, é feita uma restrição de que não pode existir nenhum tratamento da substância ativa 41 (*difenoconazol*) num intervalo de 7 dias até à colheita prevista da cultura do tomate.

Listing 5.14: Regras para o intervalo de segurança

```
1 days_before_harvest (Day, Culture, Days_left) :-
2   harvest_observation (Harvest_day, Culture),
3   day (Day),
4   Day < Harvest_day,
5   Days_left := Harvest_day - Day.
6
7 :- treat_with_actsub (Day, "LYPES", 41),
8   days_before_harvest (Day, "LYPES", Days_left),
9   Days_left <= 7.
```

5.4.4 Limitações de uso

As limitações de uso são limitações às aplicações de grupos de substâncias ativas, que se designam de famílias químicas. As famílias químicas que foram identificadas nas regras relativas à cultura do tomate são:

- Anilinopirimidinas;
- Amidas do ácido carboxílico (CAA);
- Ditiocarbamatos;
- Fenilamidas;
- Neonicotinóides;
- Piretróides;
- Inibidores de quinona (QOI).

O modo utilizado para representar estes grupos de substâncias ativas em programação por conjuntos de respostas encontram-se listados em 5.15. Os factos `activesub_group` foram obtidos através da geração de factos, explicado na secção 5.2. Desta forma, o átomo do lado esquerdo de cada regra apenas contém o identificador da substância ativa com a respetiva família química.

Listing 5.15: Famílias químicas em programação por conjuntos de resposta

```

1 group_neonicotinoides(Actsub) :- activesub_group(Actsub,1,"Neonicotinoides").
2 group_piretroides(Actsub) :- activesub_group(Actsub,2,"Piretroides").
3 group_ditiocarbamatos(Actsub) :- activesub_group(Actsub,3,"Ditiocarbamatos").
4 group_fenilamidas(Actsub) :- activesub_group(Actsub,4,"Fenilamidas").
5 group_anilinopirimidinas(Actsub)
6   :- activesub_group(Actsub,5,"Anilinopirimidinas").
7 group_meti(Actsub) :- activesub_group(Actsub,6,"METI").
8 group_hidroxianilidas(Actsub) :- activesub_group(Actsub,7,"Hidroxianilidas").
9 group_CAA(Actsub) :- activesub_group(Actsub,8,"CAA").
10 group_QOI(Actsub) :- activesub_group(Actsub,9,"QOI").

```

Com esta informação, é possível representar este tipo de regras, que são todas do mesmo género que restrições, quanto ao número máximo de aplicações, onde para este tipo de substâncias ativas, o número máximo refere-se a todo o conjunto de substâncias ativas com a mesma família química. Uma delas encontra-se representada em 5.16 para a família química CAA, não se podendo efetuar mais de 3 aplicações ao longo de todo o ciclo cultural com substâncias ativas deste grupo.

Listing 5.16: Regra de limitações de uso

```

1 0{ treat_with_actsub(Day,Cult,Actsub) : group_CAA(Actsub) : day(Day,Cult) }3.

```

5.4.5 Regras de condições meteorológicas

As condições meteorológicas também se devem ter em conta no tratamento das culturas. Algumas regras indicam as situações meteorológicas em que as substâncias ativas podem, ou não, ser aplicadas.

Para que o sistema saiba das previsões meteorológicas em determinados dias, é necessário que o utilizador as introduza. O código 5.17 ilustra este tipo de *inputs* após serem traduzidos sob a forma de factos lógicos. A propriedade `weather_observation` indica o tipo de condição meteorológica prevista (e.g., chuvoso, húmido e sol) para um determinado dia, enquanto que a propriedade `temperature_observation` representa a temperatura prevista de um determinado dia, onde o segundo argumento é a temperatura mínima e o terceiro argumento é a temperatura máxima.

Listing 5.17: *Inputs* de previsões meteorológicas

```

1 weather_observation(3,rain).
2 temperature_observation(3,10,22).

```

As regras existentes deste tipo podem afetar o intervalo entre as aplicações. Tal como podem impedir que determinado tratamento seja efetuado.

O código 5.18 representa algumas regras quanto às condições meteorológicas. As primeiras duas referem à mesma regra, onde o intervalo entre as aplicações pode ser diferente caso o estado do tempo se encontra chuvoso ou não, isto é, a primeira regra indica que o intervalo entre 2 aplicações é no mínimo de 10 dias para a substância ativa

57 (*cimoxanil+cobre (oxicloreto)*), se nesses 10 dias não houver um único dia chuvoso, caso contrário, o intervalo é reduzido para 7 dias, o que é restringido pela segunda regra.

A última regra de todas indica que o tratamento com a substância ativa 66 (*cimoxanil+propinebe*) não pode ser aplicado, caso nesse dia a previsão meteorológica seja soalheiro.

Listing 5.18: Regras relativas a condições meteorológicas

```

1 :- treat_with_actsub(Day, "LYPES", 57),
2   treat_with_actsub(Next_Day, "LYPES", 57),
3   Next_Day > Day, Next_Day - Day < 10,
4   not treat_with_actsub_between_code("LYPES", Day, Next_Day, 57),
5   Count_rain = #count{rainy_day(Day..Next_Day)},
6   Count_rain < 1.
7
8 :- treat_with_actsub(Day, "LYPES", 57),
9   treat_with_actsub(Next_Day, "LYPES", 57),
10  Next_Day > Day, Next_Day - Day < 7,
11  not treat_with_actsub_between_code("LYPES", Day, Next_Day, 57),
12  Count_rain = #count{rainy_day(Day..Next_Day)},
13  Count_rain >= 1.
14
15 :- treat_with_actsub(Day, "LYPES", 66),
16   not rainy_day(Day),
17   not wet_day(Day).

```

5.5 Custos e margens dos produtos

Uma vez escolhidos os produtos a serem utilizados por cada tratamento gerado, o sistema também tem de considerar o custo por hectare e a margem por hectare de cada um, para possibilitar a otimização do plano resultante através destes fatores.

Estes valores são previamente calculados pelo Java na altura da geração de factos, explicado na secção 5.2, multiplicando as doses máximas e mínimas, por hectare, utilizadas nas diversas doenças de cada produto com o custo e margem unitária do mesmo. O resultado de cada valor é multiplicado por 100, pois o *clasp* não efetua cálculos com valores decimais.

Para saber os valores máximos e mínimos de cada fator, foram construídas as regras que se encontram em 5.19, onde a primeira regra obtém custos mínimos e a segunda regra abrange os custos máximos. A substância ativa *Actsub* encontra-se na regra devido às doses poderem ser diferentes consoante o inimigo.

Listing 5.19: Regras para obter custos máximos e mínimos

```

1 min_product_cost(Prod, Actsub, Min_Cost) :-
2   product_cost(Prod, Actsub, _),
3   Min_Cost = #min[product_cost(Prod, Actsub, Cost) = Cost].
4

```

```

5 max_product_cost (Prod, Actsub, Max_Cost) :-
6   product_cost (Prod, Actsub, _),
7   Max_Cost = #max[product_cost (Prod, Actsub, Cost) = Cost].

```

O código 5.20 lista as regras que atribuem os custos aos vários produtos atribuídos no plano. Neste exemplo definem-se as propriedades `max_dose_prod` e `min_dose_prod` servindo para os produtos que, obrigatoriamente, devem utilizar as doses máximas/mínimas em determinadas situações, especialmente quando a cultura se encontra no estado de máximo desenvolvimento vegetativo. Quanto às regras, a primeira regra atribui um, e um só, custo máximo/mínimo quando não existem restrições ao produto que está a ser utilizado num determinado dia. Para esses casos existem as duas regras seguintes à anterior, onde a primeira atribui o custo máximo do produto e a segunda atribui o custo mínimo do mesmo. As duas últimas regras servem para atribuir a margem do produto, sendo que a primeira regra atribui uma margem máxima, caso tenha sido atribuído um custo máximo ao produto, no mesmo dia, enquanto que a segunda regra efetua o inverso da primeira.

Listing 5.20: Regras para atribuir custos e margens aos produtos utilizados

```

1 l{ used_prod_cost (Day, Prod, Cost) : product_cost (Prod, Actsub, Cost) }1 :-
2   treat_with_product (Day, Cult, Prod, Actsub),
3   not max_dose_prod (Day, Cult, Prod),
4   not min_dose_prod (Day, Cult, Prod).
5
6 used_prod_cost (Day, Prod, Cost) :-
7   treat_with_product (Day, Cult, Prod, Actsub),
8   max_product_cost (Prod, Actsub, Cost),
9   max_dose_prod (Day, Cult, Prod).
10
11 used_prod_cost (Day, Prod, Cost) :-
12   treat_with_product (Day, Cult, Prod, Actsub),
13   min_product_cost (Prod, Actsub, Cost),
14   min_dose_prod (Day, Cult, Prod).
15
16 used_prod_margin (Day, Prod, Max_Margin) :-
17   treat_with_product (Day, _, Prod, Actsub),
18   used_prod_cost (Day, Prod, Cost),
19   max_product_cost (Prod, Actsub, Cost),
20   Max_Margin = #max[product_margin (Prod, Actsub, Margin) = Margin].
21
22 used_prod_margin (Day, Prod, Min_Margin) :-
23   treat_with_product (Day, _, Prod, Actsub),
24   used_prod_cost (Day, Prod, Cost),
25   min_product_cost (Prod, Actsub, Cost),
26   Min_Margin = #min[product_margin (Prod, Actsub, Margin) = Margin].

```

5.6 Otimização

São bastantes os planos de tratamentos gerados pelas regras e factos mencionados nas secções anteriores, pois existem várias substâncias ativas possíveis para efetuar os tratamentos das culturas, assim como existe mais do que um produto possível para cada substância ativa. No entanto, entre os vários planos resultantes, alguns são melhores que outros em termos económicos, ou em termos de preferências do próprio utilizador.

A otimização trata de escolher o melhor plano para o utilizador, dependendo dos fatores que são mais importantes, e apresentar os resultados ao próprio. Os fatores disponíveis são:

- Minimizar custo total, por hectare, dos produtos utilizados no plano;
- Maximizar margem total, por hectare, dos produtos apresentados no plano;
- Maximizar preferências de produtos do comprador/vendedor;
- Minimizar o número de substâncias ativas utilizadas;
- Minimizar o número de produtos gerados;
- Minimizar o número de tratamentos.

Para tornar possível esta otimização, estes fatores estão implementados na própria linguagem de programação por conjuntos de resposta, para que seja o próprio motor do *clingo* a efetuá-la, tal como se pode observar em 5.21, que codifica os fatores indicados na listagem anterior.

A cada fator é atribuído uma prioridade, da mesma forma que foi explicada na secção 3.2. Estas prioridades são expressas pelo próprio utilizador por meio de *input* e pode, também, retirar um dos fatores da otimização.

Listing 5.21: Otimização de planos de tratamento

```
1 #minimize[ treat_with_actsub(Day,_,Actsub) @ 1 ] .
2 #minimize[ used_active_subs(Actsub) @ 2 ] .
3 #minimize[ used_product(Prod) @ 3 ] .
4 #maximize[ used_prod_margin(Day,Prod,Margin) = Margin @ 4 ] .
5 #minimize[ used_prod_cost(Day,Prod,Cost) = Cost @ 5 ] .
6 #maximize[ buyer_pref_inc(Day,Prod) @ 6 ] .
7 #maximize[ seller_pref_inc(Day,Prod) @ 7 ] .
```

No momento de geração dos planos de tratamento, a motor da linguagem de programação por conjuntos de resposta irá tentar encontrar o melhor plano. O plano que for declarado como ótimo, apresenta-se com a indicação de “*OPTIMUM FOUND*” nos resultados. Na leitura dos resultados finais, são interpretados os diversos planos até encontrar o plano com essa indicação e são apresentados no ecrã.

5.7 Aplicação

A aplicação construída oferece uma interface simples, apenas para o utilizador indicar os dados necessários para enviar para o sistema e receber os planos gerados. A figura 5.3 ilustra a página inicial da aplicação.

Sistema de aconselhamento de aplicações fitofarmacêuticas Dissertação de mestrado de João Rico

Cultura

- Observações de inimigos
- Mudanças de estado
- Preferências de produtos
- Tratamentos preventivos
- Produtos preventivos usados
- Tratamentos curativos
- Otimizações
- Plano gerado

Cultura Selecionada

Cultura: Tomate

Local: Ar Livre

Objetivo: Consumo

Colheita: 181

Guardar

Gerar plano ✓

Limpar dados ✖

Figura 5.3: Página inicial da aplicação

No topo da página encontra-se uma barra a indicar o nome do sistema e o autor do mesmo. Do lado esquerdo está o menu que dá acesso a todas as funcionalidades da interface que foram indicadas na secção 4.3. Por fim, o lado direito da página apresenta o conteúdo necessário a cada funcionalidade.

5.7.1 Gestão de *inputs*

Todas as opções da barra lateral acedem a páginas para introdução de dados, exceto a opção de ver o Plano gerado e os botões Gerar plano e Limpar dados. Na figura 5.3, mencionada anteriormente, o utilizador encontra-se na página de introdução da cultura alvo. Sem a indicação da cultura, as outras opções irão avisar o utilizador de que a cultura não foi escolhida. Sendo assim, o utilizador tem de indicar a cultura, o local da plantação, o destino objetivo e o dia para o qual está prevista a colheita da mesma.

Após a cultura se encontrar selecionada, tem-se o acesso às outras páginas de introdução de dados. A opção Observações de inimigos regista a informação sobre os inimigos que estão a afetar a cultura. Um exemplo desta página encontra-se na figura 5.4, que possui uma tabela com os dados que já foram previamente introduzidos pelo utilizador.

Nesta caso já se encontram introduzidas 4 observações de inimigos, podendo adicionar novas através do botão Adicionar Observação. Ao pressionar este botão, surge uma janela com o formulário para adicionar uma nova observação, tal como se pode ver na figura 5.5. Os dados que se podem introduzir são os mesmos que se encontravam na tabela que lista as observações, isto é, o dia da observação, o tempo meteorológico do

Observações de inimigos

Dia	Cultura	Tipo de inimigo	Inimigo	Estado	Temperatura	Tempo
3	Tomate	Lagarta	Chrysodeixis chalcites	Fraco	10°C~22°C	Chuvoso
4	Tomate	Lagarta	Chrysodeixis chalcites	Forte	10°C~22°C	Sol
13	Tomate	Lagarta	Chrysodeixis chalcites	Forte	10°C~22°C	Sol
14	Tomate	Nóctuas	Spodoptera exigua	Forte	10°C~22°C	Sol

Adicionar Observação

Figura 5.4: Página de observações de inimigos

dia, o tipo de inimigo, o organismo e o estado de severidade do problema. Após os dados serem indicados, basta carregar no botão Adicionar e a nova observação será listada na tabela respetiva.

Adicionar nova observação

Dia: Dia da observação

Tempo: Sol Chuvoso Húmido

Temp. Min.: Temp. Máx.:

Tipo de inimigo: Escolher um tipo de inimigo...

Organismo: Escolher um organismo...

Estado: Fraco

Cancelar Adicionar

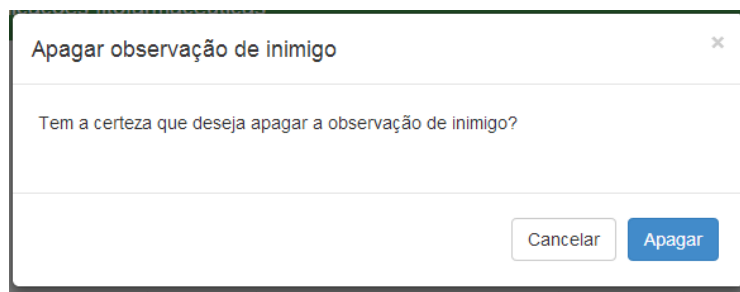
Figura 5.5: Formulário de adicionar observações

Também é possível eliminar uma determinada observação. Ao sobrepor o cursor numa determinada observação da tabela correspondente, irá surgir um botão do lado direito que permite proceder à eliminação da respetiva observação. Posteriormente, irá aparecer uma janela de confirmação da própria remoção para impedir o utilizador de se enganar, como demonstra a figura 5.6.

Todas as páginas de introdução de dados seguintes comportam-se da mesma forma que a de observações de inimigos, apenas diferem nos dados que são apresentados na tabela e nos campos do formulário de inserção de novos dados. Na mudança de estados é necessário a introdução do dia em que foi observada a mudança e o estado fenológico respetivo. Na página de preferências de produtos basta indicar o produto em questão e o

Dia	Cultura	Tipo de inimigo	Inimigo	Estado	Temperatura	Tempo
3	Tomate	Lagarta	Chrysodeixis chalcites	Fraco	10°C~22°C	Chuvoso
4	Tomate	Lagarta	Chrysodeixis chalcites	Forte	10°C~22°C	Sol

(a) Botão de remoção



(b) Confirmação

Figura 5.6: Remoção de uma observação de inimigo

tipo de preferência (comprador ou vendedor). Nos tratamentos preventivos indicam-se quais os dias em que se quer começar a efetuar aplicações preventivas na cultura. Caso o utilizador dê a indicação ao sistema de prever o próximo dia, só através da geração de novos planos é que se pode indicar o próximo dia onde foi previsto o próximo tratamento. Nestes casos, ao adicionar um novo tratamento preventivo, o sistema introduz o dia automaticamente, desativando o campo correspondente, para evitar que exista factos com propriedade `start_prevent_treatment` e `preventive_treatment` com dias diferentes. Além do dia, é necessário mencionar o inimigo que se quer tratar, o tipo de condições que se encontram no dia do tratamento e se deseja efetuar uma nova aplicação posterior ao que será adicionado. A introdução dos produtos utilizados nas aplicações é feita na página de `Produtos preventivos usados`, para os tratamentos preventivos, e na página de `Tratamentos curativos`, para os tratamentos curativos. No caso dos produtos preventivos, o utilizador só pode adicionar tratamentos aos dias marcados para tratamento preventivo. Quanto às previsões futuras, só é possível adicionar um produto utilizado, caso o produto do dia anterior seja igual ao que foi aconselhado pelo sistema, devido à persistência dos produtos serem diferentes. Se for indicado um produto diferente ao aconselhado, o sistema terá de gerar novamente o plano para saber o dia previsto para o próximo tratamento através do novo produto. A remoção de um tratamento preventivo implica a eliminação de todos os tratamentos seguintes ao dia removido, pois o sistema terá de efetuar novas previsões para obter os dias corretos, se assim o utilizador indique ao sistema para efetuar tais previsões. Por fim, nos produtos utilizados para os tratamentos curativos só é possível adicionar nos tratamentos que são indicados no plano resultante, para que o sistema indique a substância ativa que foi aconselhada a utilizar para cada observação de inimigo que foi listada pelo utilizador.

O último input necessário é a escolha de otimizações que o utilizador deseja que

o sistema faça. Esta página já se comporta de maneira diferente das outras, pois existe apenas um número limitado de otimizações, não havendo possibilidade de inserir novas, como se pode ver na figura 5.7 que ilustra a página de *Otimizações* que serve para gerir a escolha mencionada.

Otimizações

Otimizações a efetuar	Otimizações a não efetuar
<input type="checkbox"/> Min. Produtos	<input type="checkbox"/> Máx. Preferências Comprador
<input type="checkbox"/> Min. Substâncias ativas	<input type="checkbox"/> Máx. Preferências Vendedor
<input type="checkbox"/> Min. Tratamentos	
<input type="checkbox"/> Máx. Margem Total	
<input type="checkbox"/> Min. Custo Total	

Figura 5.7: Escolha de otimizações

Esta página possui duas listas: otimizações a efetuar e otimizações a não efetuar. A passagem de uma otimização de uma lista para a outra é feita através de um arraste com o cursor do rato. Após a escolha de otimizações desejadas, basta carregar no botão *Guardar* para finalizar. Caso o utilizador peça ao sistema para gerar um plano com a lista de otimizações a efetuar vazia, por defeito será efetuada uma otimização de minimização de tratamentos.

Todos estes dados ficam em *local storage* do *browser*, para que o utilizador possa continuar com o seu trabalho mais tarde, desde que não limpe a *cache* do mesmo. Caso deseje efetuar uma limpeza apenas desta página, existe o botão *Limpar dados* na barra lateral para esse efeito. Consequentemente, o utilizador terá de introduzir novos dados como se tivesse a interagir com a aplicação pela primeira vez.

5.7.2 Demonstração de um plano

Após os *inputs* indicados na secção anterior, o utilizador pode enviá-los para o sistema através do botão *Gerar plano* na barra lateral, estes serão transmitidos num ficheiro JSON. Após o envio dos dados, o sistema irá efetuar a leitura dos mesmos e traduzi-los para um ficheiro com factos lógicos da mesma forma que a geração de factos da secção 5.2. Posteriormente, será executado o *clingo* num processo extra com todos os ficheiros necessários para obtenção dos planos resultantes e o sistema irá aguardar a sua execução.

Ao terminar o processo, os planos resultantes são escritos num ficheiro de texto, onde o sistema efetua uma leitura dos mesmos e transforma-os em formato JSON para devolver ao utilizador. Os planos são apresentados através de uma tabela como na figura 5.8

Plano gerado

Plano 17 (Plano ótimo)

Dia	Tipo de tratamento	Substância ativa	Produto	Custo (ha)	Margem (ha)
74	Curativo	metribuzina	SEN WG	44.62€	4.46€
93	Curativo	tiametoxame	ACT 25 WG	38.50€	3.85€
93	Preventivo	cobre (hidróxido)	KAD	37.50€	3.75€
93	Preventivo	cimoxanil+mancozebe	MIC	62.49€	6.25€
93	Preventivo	enxofre	COS WDG	5.20€	0.52€
105	Curativo	folpete+iprovalcarbe	MEL	48.50€	4.85€
105	Curativo	cipermetrina	CYT 10 EC	37.00€	3.70€
115	Curativo	cimoxanil+folpete	VIT	40.00€	4.00€
115	Curativo	alfa-cipermetrina	FAS	16.30€	1.63€
115	Curativo	indoxacarbe	STE	48.05€	4.81€
128	Curativo	cimoxanil+folpete	VIT	40.00€	4.00€
128	Curativo	clorantraniliprol	ALT	74.75€	7.48€
143	Curativo	clorpirifos	PYR 48 EC	24.75€	2.48€
143	Preventivo	ciazofamida	RAN	35.13€	3.51€
155	Preventivo	enxofre	BAG	7.60€	0.76€
168	Curativo	cimoxanil+folpete	VIT	40.00€	4.00€
168	Curativo	clorantraniliprol	COR	70.04€	7.00€
Total:				670.43€	67.05€

Figura 5.8: Plano de tratamentos

com os dias em que o agricultor deve efetuar tratamentos, tipo de tratamentos, a substância ativa, os produtos, custos e margens de cada um, bem como o total do plano. Através do menu situado por cima do plano que se está a analisar, o utilizador tem a possibilidade de analisar outros os planos resultantes para que possa verificar as diversos planos de tratamentos válidos e escolher aquele que deseja por em prática. O plano de tratamento que o sistema declarou como ótimo possui uma indicação do mesmo na lista.

5.8 Conclusão

Neste capítulo foi apresentado o modo como foram desenvolvidas as diversas componentes e funcionalidades do sistema pretendido nesta dissertação, indicadas no capítulo anterior.

Em suma, foi apresentado o modo como são recolhidos os dados, bem como a transformação e carregamento dos mesmos para a base de dados, implementada e organizada com base no modelo apresentado no capítulo 4.

Demonstrou-se a possibilidade de criação de um sistema de geração e otimização de planos válidos, sem a necessidade de algoritmos específicos, graças à utilização da tecnologia de programação por conjuntos de resposta. A sua eficiência será analisada no próximo capítulo.

Criou-se uma interface simples para inserção dos dados necessários para a geração de novos planos de tratamento, onde posteriormente o utilizador deve inserir os tratamentos efetuados à cultura que indicou. O sistema gera diversos planos de tratamentos e apresenta-os ao utilizador para que possa escolher o que desejar por em prática.

6

Validação

Neste capítulo valida-se e avalia-se o sistema desenvolvido, através da análise do comportamento do sistema em vários cenários realistas diferentes e obter os seus resultados.

Na secção 6.1 deste capítulo são expressos os diferentes métodos utilizados para validar o sistema através dos cenários indicados na secção 6.2. Nesta secção são apresentados os dados utilizados, bem como os resultados obtidos através do sistema. Por fim, na secção 6.3 são extraídas conclusões acerca do desempenho do sistema.

6.1 Metodologias utilizadas

A avaliação do sistema desenvolvido foi realizada através da execução de dois tipos de testes distintos: validação de planos pelas regras implementadas e eficiência do sistema na geração de planos de tratamento.

Na validação do sistema quanto às regras implementadas, são inseridas as observações indicadas nos diversos planos de tratamento utilizados, relativamente à cultura do tomate, bem como os produtos aplicados, que se podem encontrar na secção 6.2. Após a inserção dos mesmos, a execução do sistema demonstra se o cenário correspondente é satisfazível ou não. No caso de um cenário ser satisfazível, o resultado obtido é o plano original, com a atribuição dos custos/margens por hectare. Caso contrário, o sistema apenas indica que não é válido e, manualmente, é obtido e justificado a razão do plano ser inválido. Para estes casos, retiram-se os produtos que tornam plano inválido e são inseridos produtos alternativos, indicados pelo próprio sistema, para o tratamento das observações inseridas, caso existam alternativas.

Para testar o desempenho do sistema na geração de planos de tratamento são utilizados os mesmos cenários do teste anterior, no entanto apenas são retirados alguns

produtos utilizados, deixando todas as observações para o sistema gerar vários planos de tratamentos diferentes e declarar o plano ótimo, consoante os fatores utilizados. No fim, são recolhidos dados quanto ao tempo de execução e ao número de planos gerados.

Para ambos os testes são realizados cálculos manuais para confirmar o custo e margem total, por hectare, do plano resultante que varia quanto à dose utilizada em cada aplicação de um produto. Para simplificar, assume-se que a margem é sempre de 10% sobre o custo unitário do produto. Os preços dos produtos são fictícios.

6.2 Testes e resultados

Nesta secção são apresentados os diversos planos de tratamentos utilizados em diferentes cenários. Todos eles possuem a mesma duração dos estados fenológicos e a previsão de colheita para o dia 181. Além disso, todos os cenários foram executados com os seguintes critérios de otimização e ordem:

1. Maximização de preferências do vendedor;
2. Maximização de preferências do comprador;
3. Maximização da margem total por hectare dos planos de tratamento;
4. Minimização do custo total por hectare dos planos de tratamento.
5. Minimização do número de produtos utilizados;
6. Minimização do número de substâncias ativas utilizadas;
7. Minimização do número de tratamentos;

Esta ordem de otimizações irá gerar novos planos de tratamento mais caros que o plano original utilizado para o primeiro teste de cada cenário, bem como a inclusão de produtos declarados como preferência de vendedor ou comprador, sendo mais benéfico para a entidade que está a vender os produtos.

Para cada cenário são indicadas as preferências de produtos declaradas, os dados utilizados para ambos os testes, incluindo os seus resultados, e é feita uma comparação, quanto ao custo total e produtos equivalentes entre o plano utilizado no teste à validação das regras e o plano ótimo gerado no teste ao desempenho.

6.2.1 Cenário 1

O primeiro cenário corresponde a uma cultura do tomate, instalada ao ar livre e destinada ao consumo. A tabela 6.1 indica o plano de tratamento correspondente ao primeiro cenário, contendo as observações e os produtos utilizados.

Antes da execução do sistema com este cenário, o produto SEN WG foi declarado como preferência do vendedor.

Tabela 6.1: Plano de tratamento do cenário 1

Dia	Tipo	Inimigo	Nível	Substância ativa	Produto
74	Curativo	Infestantes	Forte	metribuzina	SE WG
82	Curativo	Afídeos	Fraco	imidaclopridem	CORS
82	Preventivo	Míldio		cimoxanil+mancozebe	TOR
93	Curativo	Mosca branca	Fraco	imidaclopride	CON O-TEQ
93	Preventivo	Míldio		folpete	FOL 50
93	Preventivo	Oídio		enxofre	STU
103	Curativo	Míldio	Fraco	folpete+iprovalicarbe	MEL
116	Curativo	Lagarta	Fraco	Bacillus thuringiensis	BEL
116	Preventivo	Oídio		azoxistrobina	ORT
116	Preventivo	Míldio		azoxistrobina	ORT
136	Curativo	Míldio	Forte	cimoxanil+folpete	VIT
136	Curativo	Lagarta	Forte	clorantraniliprol	CORA
136	Curativo	Tripes	Fraco	acrinatrina	RUF
152	Curativo	Míldio	Fraco	mandipropamida	REV
152	Curativo	Lagarta	Forte	clorantraniliprol	CORA
152	Curativo	Alternariose	Fraco	difenoconazol	SCO 250 EC
165	Curativo	Míldio	Forte	cimoxanil+famoxadona	GAL
165	Curativo	Ácaros	Fraco	hexitiazox	VIR

Na sua execução, o sistema que indicou que o plano encontra-se válido quanto às regras de Proteção Integrada, devolvendo os custos/margens por hectare que se encontram na tabela 6.2.

Tabela 6.2: Custo/margem total do plano original do cenário 1

Dia	Produto	Custo/ha	Margem/ha
74	SEN WG	148,72€	14,87€
82	CORS	39,00€	3,90€
82	TOR	27,00€	2,70€
93	CON O-TEQ	39,65€	3,97€
93	FOL 50	28,00€	2,80€
93	STU	10,00€	1,00€
103	MEL	48,50€	4,85€
116	BEL	13,74€	1,37€
116	ORT	65,07€	6,51€
136	CORA	80,04€	8,00€
136	RUF	141,60€	14,16€
136	VIT	40,00€	4,00€
152	CORA	80,04€	8,00€
152	REV	45,18€	4,52€
152	SCO 250 EC	47,58€	4,76€
165	GAL	35,00€	3,50€
165	VIR	46,00€	4,60€
Totais:		935,12€	93,51€

A tabela 6.3 indica as observações às quais foram retirados os tratamentos no plano original para testar o sistema quanto ao desempenho, assim como os tratamentos posteriormente sugeridos pelo sistema, mantendo os restantes tratamentos. Os produtos que se encontram nesta tabela, são os que resultaram do plano gerado e declarado como ótimo.

Como se pode ver, nenhum produto utilizado previamente se manteve no novo plano de tratamento. No entanto, alguns são também utilizados noutros tratamentos, graças à

Tabela 6.3: Tratamentos substituídos no cenário 1

Dia	Tipo	Inimigo	Substância ativa	Produto
116	Curativo	Lagarta	spinosade	SPI
116	Preventivo	Oídio	enxofre	BAG
152	Curativo	Lagarta	spinosade	SPI
152	Curativo	Alternariose	azoxistrobina	ORT
165	Curativo	Míldio	cobre (sulfato de cobre e cálcio)	CAL
165	Curativo	Ácaros	hexitiazox	NISS

minimização do número de produtos, como é o caso do produto SPI e do ORT.

Para gerar este novo plano, o sistema gerou 68 modelos diferentes e demorou, aproximadamente, 2 minutos a declarar um dos modelos como ótimo. Além disso, sofreram-se alterações quanto aos custos e margens totais, que estão expressas na tabela 6.4.

Tabela 6.4: Custo/margem total do novo plano do cenário 1

Dia	Produto	Custo/ha	Margem/ha
74	SEN WG	148,72€	14,87€
82	CORS	39,00€	3,90€
82	TOR	27,00€	2,70€
93	CON O-TEQ	39,65€	3,97€
93	FOL 50	28,00€	2,80€
93	STU	10,00€	1,00€
103	MEL	48,50€	4,85€
116	SPI	38,00€	3,80€
116	BAG	65,07€	6,51€
116	ORT	185,00€	18,50€
136	VIT	80,04€	8,00€
136	CORA	141,60€	14,16€
136	RUF	40,00€	4,00€
152	REV	65,07€	6,51€
152	SPI	45,18€	4,52€
152	ORT	185,00€	18,50€
165	CAL	182,50€	18,25€
165	NISS	77,70€	7,77€
Totais:		1446,03€	144,61€

Graças à maximização da margem total do plano ser a prioridade mais alta, o sistema escolheu doses e custos mais caros para cada tratamento retirado, fazendo com que o novo plano de tratamentos seja mais caro do que o original.

6.2.2 Cenário 2

O segundo cenário possui as mesmas características quando à cultura utilizada no cenário anterior. No entanto, não existem quaisquer preferências declaradas. O plano correspondente a este cenário encontra-se na tabela 6.5.

O plano indicado não foi considerado válido quanto às regras implementadas, pois a substância ativa do produto MAT (*lufenurão*) só se pode utilizar em culturas instaladas em estufa, o que não é o caso. Para evitar a inconsistência do plano, substituiu-se o produto MAT, pelo produto SPI, com substância ativa *spinosade*. Os custos devolvidos pelo plano encontram-se na tabela 6.6. Nesta tabela, observa-se que o plano é mais barato que o do

Tabela 6.5: Plano de tratamento no cenário 2

Dia	Tipo	Inimigo	Nível	Substância ativa	Produto
74	Curativo	Infestantes	Forte	rinsulfurão	TITU
87	Curativo	Afídeos	Fraco	acetamiprida	EPI SG
87	Preventivo	Míldio		mancozebe	MAN
87	Preventivo	Oídio		enxofre	MIC
101	Curativo	Míldio	Fraco	cimoxanil+famoxadona	EQU
101	Curativo	Mosca Branca	Fraco	lambda-cialotrina	JUD
111	Preventivo	Oídio		enxofre	BAG
116	Curativo	Míldio	Forte	cimoxanil+mancozebe	CIM
116	Curativo	Lagarta	Fraco	lufenurão	MAT
128	Curativo	Míldio	Forte	dimetomorfe+piraclostrobina	CAB
128	Curativo	Alternariose	Fraco	dimetomorfe+piraclostrobina	CAB
128	Curativo	Lagarta	Fraco	spinosade	SPI
143	Curativo	Lagarta	Fraco	Bacillus thuringiensis	DIP
143	Preventivo	Míldio		folpete	ORT
163	Preventivo	Ácaros		enxofre	BAG
163	Curativo	Lagarta	Forte	Bacillus thuringiensis	DIP
163	Curativo	Míldio	Forte	cimoxanil+cobre (oxicloreto)	VIT C WG ADV

cenário anterior, uma vez que é menos 1 tratamento efetuado e existem alguns produtos mais baratos, à exceção do produto SPI que é dos mais caros.

Tabela 6.6: Custo/margem total do plano original do cenário 2

Dia	Produto	Custo/ha	Margem/ha
74	TITU	78,00€	7,80€
87	EPI SG	55,60€	5,56€
87	MAN	20,72€	2,07€
87	MIC	10,28€	1,03€
101	EQU PRO	35,43€	3,54€
101	JUD	26,80€	2,68€
111	BAG	38,00€	3,80€
116	CIM	25,29€	2,53€
116	SPI	185,00€	18,50€
128	CAB	61,80€	6,18€
128	SPI	185,00€	18,50€
143	DIP	15,66€	1,57€
143	ORT	19,20€	1,92€
163	BAG	38,00€	3,80€
163	DIP	15,66€	1,57€
163	VIT C WG ADV	38,49€	3,85€
Totais:		848,93€	84,90€

Para testar o desempenho do sistema neste cenário, foram retirados alguns tratamentos de certas observações para o sistema gerar um novo plano. Os tratamentos substituídos no cenário 2 encontram-se na tabela 6.7.

Neste novo plano, o produto SPI foi utilizado novamente e apenas o BAG se manteve em relação ao plano original. Para chegar a este resultado, o sistema gerou 94 planos de tratamento diferentes em, aproximadamente, 28 minutos. A razão do grande aumento de tempo de execução, em relação ao 1º cenário, deve-se ao facto de a maioria dos inimigos, cujo os tratamentos foram retirados, terem muitos produtos possíveis para os respetivos tratamentos.

Naturalmente, foram sofridas alterações quanto aos valores totais do custo e margem por hectare com o novo plano. Tais valores podem ser analisados através da tabela 6.8.

Tabela 6.7: Tratamentos substituídos no cenário 2

Dia	Tipo	Inimigo	Substância ativa	Produto
87	Curativo	Afídeos	imidaclopride	CON
128	Curativo	Alternariose	azoxistrobina	ORT
128	Curativo	Lagarta	clorpirifos	RIS 48 EC
143	Curativo	Lagarta	clorpirifos	RIS 48 EC
163	Preventivo	Ácaros	enxofre	BAG
163	Curativo	Míldio	cobre (sulfato de cobre e cálcio)	CAL
163	Curativo	Lagarta	spinosade	SPI

Novamente foi gerado um plano mais caro que o plano original devido à utilização dos mesmos fatores de otimização.

Tabela 6.8: Custo/margem total do novo plano do cenário 2

Dia	Produto	Custo/ha	Margem/ha
74	TITU	78,00€	7,80€
87	CON	61,50€	6,15€
87	MAN	20,72€	2,07€
87	MIC	10,28€	1,03€
101	EQU PRO	35,43€	3,54€
101	JUD	26,80€	2,68€
111	BAG	38,00€	3,80€
116	CIM	25,29€	2,53€
116	SPIR	185,00€	18,50€
128	CAB	61,80€	6,18€
128	ORT	65,07€	6,51€
128	RIS 48 EC	131,42€	13,14€
143	ORT	19,20€	1,92€
143	RIS 48 EC	131,42€	13,14€
163	BAG	38,00€	3,80€
163	CAL	182,50€	18,25€
163	SPI	185,00€	18,50€
Totais:		1295,43€	129,54€

6.2.3 Cenário 3

O plano de tratamento deste cenário encontra-se na tabela 6.9. A cultura à qual se efetuou os tratamentos indicados é igual ao dos cenários anteriores. Tal como o cenário 2, não possui preferências declaradas pelo utilizador.

Para este plano, o sistema indica que não se encontra satisfável. A razão de se tratar de um plano inválido deve-se ao facto de o sistema não ter qualquer produto preventivo para a lagarta, ou seja, o produto PYR 5G provoca a inconsistência do plano.

Consequentemente, não existe qualquer alternativa para as observações listadas, impedindo de saber o custo do plano e de testar o desempenho do sistema para obter um plano diferente do original.

6.2.4 Cenário 4

Ao contrário dos cenários anteriores, a cultura utilizada para este cenário encontra-se instalada em estufa, mantendo o destino final o consumo. Além disso, o produto VIT

Tabela 6.9: Plano de tratamento do cenário 3

Dia	Tipo	Inimigo	Nível	Substância ativa	Produto
74	Curativo	Infestantes	Forte	fluazifope-P-butilo	FUS MAX
96	Preventivo	Míldio		mancozebe	PENN WG
96	Curativo	Afídeos	Fraco	pimetrozina	PLE 50 WG
96	Preventivo	Oídio		enxofre	STU WG ADVANCE
118	Curativo	Mosca Branca	Fraco	acetamiprida	EPI SG
118	Preventivo	Lagarta		clorpirifos	PYR 5G
118	Curativo	Míldio	Fraco	mandipropamida	REV
133	Preventivo	Oídio		enxofre	BAG
133	Curativo	Mosca Branca	Forte	lambda-cialotrina	KAR ZEON
133	Curativo	Míldio	Forte	cimoxanil+folpete+mancozebe	MIL
148	Curativo	Lagarta	Fraco	Bacillus thuringiensis	DIP
148	Curativo	Míldio	Forte	ciazofamida	RAN
158	Curativo	Lagarta	Forte	Bacillus thuringiensis	BEL
158	Preventivo	Míldio		cimoxanil+mancozebe	TOR
163	Curativo	Lagarta	Fraco	clorantraniliprol	ALT
163	Curativo	Míldio	Fraco	cimoxanil+folpete	VIT
170	Preventivo	Míldio		folpete	FOL 80 WDG
170	Curativo	Ácaros	Fraco	hexitiazox	NIS

está declarado como preferência de vendedor. O plano correspondente a este cenário encontra-se na tabela 6.10.

Tabela 6.10: Plano de tratamento do cenário 4

Dia	Tipo	Inimigo	Nível	Substância ativa	Produto
74	Curativo	Infestantes	Forte	metribuzina	SEN WG
93	Preventivo	Míldio		cimoxanil+mancozebe	MIC PLUS AZUL
93	Preventivo	Bacteriose		cobre (hidróxido)	KAD
93	Preventivo	Oídio		enxofre	COS WDG
93	Curativo	Afídeos	Fraco	tiametoxame	ACT 25 WG
105	Curativo	Mosca Branca	Fraco	cipermetrina	CYT 10 EC
105	Curativo	Míldio	Fraco	folpete+iprovalicarbe	MEL
115	Curativo	Mosca Branca	Forte	alfa-cipermetrina	FAS
115	Curativo	Míldio	Forte	cimoxanil+folpete	VIT
115	Curativo	Lagarta	Fraco	indoxacarbe	STE
128	Curativo	Míldio	Forte	cimoxanil+folpete	VIT
128	Curativo	Lagarta	Forte	clorantraniliprol	ALT
143	Preventivo	Míldio		ciazofamida	RAN
143	Curativo	Lagarta	Forte	clorpirifos	PYR 48 EC
155	Preventivo	Oídio		enxofre	BAG
168	Curativo	Míldio	Fraco	cimoxanil+folpete	VIT
168	Curativo	Lagarta	Fraco	clorantraniliprol	COR
180	Preventivo	Míldio		cobre (sulfato de cobre e cálcio)	CAL
180	Curativo	Ácaros	Fraco	hexitiazox	NISS

Do mesmo modo que o cenário anterior, o sistema indicou que este plano de tratamento não se encontra satisfazível. Desta vez, as razões para o plano ser inválido são diferentes. Sendo que a cultura se encontra instalada em estufa, existe uma regra que impede a utilização de produtos com *folpete* ou *captana* mais do que 2 vezes, ao longo de todo o ciclo cultural. Neste plano existe 4 utilizações produtos que contêm *folpete*, que são o MEL e o VIT.

Existe também uma outra razão pelo plano ser inválido, tanto para cultura de estufa, como cultura em ar livre. Os dois últimos tratamentos efetuados no dia 180 não podem ser realizados, pois a previsão para a colheita é no dia 181 e ambos os produtos possuem um intervalo de segurança que é superior ao tempo que resta desde a aplicação até à

colheita.

Como não existem alternativas às observações existentes, para o plano apresentado se tornar válido, não foram efetuados testes quanto ao desempenho do sistema para gerar um plano diferente.

6.2.5 Cenário 5

Para este cenário, voltou-se à cultura habitual, instalada ao ar livre e destinada ao consumo. Como preferência foi escolhido o produto STE para o vendedor. Além disso, foi trocada a ordem da prioridade das otimizações entre a maximização da margem total dos planos e a minimização do custo total dos planos, mantendo-se as outras. Desta vez ao testar o desempenho, supõe-se que se obtém um plano mais barato que o plano original.

O plano utilizado para a verificação da sua validade encontra-se na tabela 6.11. Sendo este plano válido quanto às regras implementadas, o sistema devolveu o custo do plano que se encontra na tabela 6.12.

Tabela 6.11: Plano de tratamento do cenário 5

Dia	Tipo	Inimigo	Nível	Substância ativa	Produto
74	Curativo	Infestantes	Forte	quizalofope-P-etilo	TAR GOLD
100	Preventivo	Alternariose		clortalonil	BRA 500
100	Preventivo	Míldio		mancozebe	MAN
123	Curativo	Mosca Branca	Fraco	imidaclopride	CON O-TEQ
123	Curativo	Míldio	Fraco	cimoxanil+famoxadona	GAL
129	Curativo	Oídio	Fraco	azoxistrobina	ORT
144	Curativo	Míldio	Forte	cimoxanil+folpete	VIT
156	Curativo	Lagarta	Fraco	clorantraniliprol	COR
156	Preventivo	Oídio		azoxistrobina	ORT
167	Curativo	Tuta	Forte	indoxacarbe	STE
167	Curativo	Míldio	Forte	cimoxanil+folpete	VIT
173	Curativo	Ácaros	Fraco	acrinatrina	RUF

Tabela 6.12: Custo/margem total do plano original do cenário 5

Dia	Produto	Custo/ha	Margem/ha
74	TAR GOLD	64,87€	6,49€
100	BRA 500	37,60€	3,76€
100	MAN	20,72€	2,07€
123	CON O-TEQ	39,65€	3,97€
123	GAL	35,00€	3,50€
129	ORT	65,07€	6,51€
144	VIT	40,00€	4,00€
156	COR	70,04€	7,00€
156	ORT	65,07€	6,51€
167	STE	48,05€	4,81€
167	VIT	40,00€	4,00€
173	RUF	106,20€	10,62€
Totais:		632,27€	63,24€

Passando para o teste de desempenho do sistema, os tratamentos a que foram retirados os produtos aplicados estão listados na tabela 6.13. É de notar que foram retirados

muitos mais tratamentos que nos outros cenários, sobrando apenas dois tratamentos curativos. Consequentemente, o sistema irá produzir mais modelos.

Tabela 6.13: Tratamentos substituídos no cenário 5

Dia	Tipo	Inimigo	Substância ativa	Produto
100	Preventivo	Alternariose	mancozebe	MAN SAP
100	Preventivo	Míldio	mancozebe	MAN SAP
123	Curativo	Mosca Branca	lambda-cialotrina	KAR+
123	Curativo	Míldio	mancozebe	MAN SAP
129	Curativo	Oídio	enxofre	ALA MICRO
144	Curativo	Míldio	cobre (hidróxido)	KOC 2000
156	Preventivo	Oídio	enxofre	ALA MICRO
156	Curativo	Lagarta	indoxacarbe	STE
167	Curativo	Míldio	mancozebe	MAN SAP
167	Curativo	Tuta	indoxacarbe	STE
173	Curativo	Ácaros	hexitiazox	VIR

Devido à prioridade máxima das otimizações ser o número de produtos declarados como preferências, o plano resultante possui mais tratamentos com o produto STE que é uma preferência de vendedor.

O sistema gerou 499 modelos em, aproximadamente, 12 minutos antes de encontrar o modelo ótimo para as observações indicadas. Na tabela 6.14 observa-se que este plano é mais económico que o original, graças à alteração das prioridades de otimização. Além disso, foram repetidos muitos produtos como MAN SAP e ALA MICRO devido ao seu baixo custo por hectare e à minimização do número de produtos/substâncias ativas a utilizar.

Tabela 6.14: Custo/margem total do novo plano do cenário 5

Dia	Produto	Custo/ha	Margem/ha
74	TAR GOLD	64,87€	6,49€
100	MAN SAP	14,40€	1,44€
123	KAR+	8,59€	0,86€
123	MAN SAP	14,40€	1,44€
129	ALA MICRO	6,50€	0,65€
144	KOC 2000	19,04€	1,90€
156	ALA MICRO	6,50€	0,65€
156	STE	48,05€	4,81€
167	MAN SAP	14,40€	1,44€
167	STE	48,05€	4,81€
173	VIR	46,00€	4,60€
Totais:		282,21€	29,09€

O tempo total de execução é menor graças às preferências de vendedor ser uma prioridade máxima, onde o *clingo* tenta, imediatamente, colocar os produtos comerciais declarados como tal, para as doenças em que é possível a sua utilização, tentando não infringir as regras implementadas.

6.2.6 Cenário 6

Para o seguinte cenário é aproveitado o plano de tratamento referido no cenário 4, excluindo os tratamentos no dia 180, para se tornar num plano satisfazível. Além disso, manteve-se o produto VIT como preferência de utilizador e alterou-se o local de cultura para ar livre. As otimizações são as mesmas do cenário anterior, isto é, a prioridade da minimização do custo total, por hectare, dos planos de tratamento é superior à da maximização da margem total, por hectare, dos mesmos.

Os objetivos deste cenário são demonstrar a eficácia com que o sistema consegue obter planos de tratamento quando tem de propor produtos comerciais para poucas observações e, novamente, provar as vantagens de ter produtos declarados como preferência.

A execução deste cenário foi feita através da inserção das observações uma à uma. A cada resultado é retirado o produto que foi aconselhado pelo sistema, juntamente com o tempo total, em segundos, que demorou a gerar o plano e o número de modelos gerados. Posteriormente, é colocado o produto igual ao plano do cenário 4 para a respetiva observação e passa-se à observação seguinte. A tabela 6.15 lista os resultados obtidos pela ordem com que foram inseridas as observações. Foram colocados apenas os produtos que foram aconselhados pelo sistema e qual o produto que foi escolhido pelo utilizador, sem a informação da substância ativa de cada um, uma vez que o plano resultante é válido.

Tabela 6.15: Plano de tratamentos do cenário 6

Dia	Tipo	Inimigo	Produto aconselhado	Produto escolhido	Tempo (s)	Modelos
74	Curativo	Infestantes	SEN WG	SEN WG	0,047	5
93	Curativo	Afídeos	KOH 20 SL	ACT 25 WG	0,047	5
93	Preventivo	Bacteriose	NUC	KAD	0,047	2
93	Preventivo	Míldio	VIT	MIC	0,062	8
93	Preventivo	Oídio	KUM S	COS WDG	0,062	9
105	Curativo	Míldio	VIT	MEL	0,062	14
105	Curativo	Mosca Branca	KAR+	CYT 10 EC	0,078	7
115	Curativo	Mosca Branca	KAR+	FAS	0,078	7
115	Curativo	Míldio	VIT	VIT	0,125	16
115	Curativo	Lagarta	BEL	STE	0,141	5
128	Curativo	Míldio	VIT	VIT	0,203	14
128	Curativo	Lagarta	BEL	ALT	0,219	5
143	Preventivo	Míldio	VIT	RAN	0,344	23
143	Curativo	Lagarta	BEL	PYR 48 EC	0,375	7
155	Preventivo	Oídio	ALA MICRO	BAG	0,391	20
168	Curativo	Míldio	VIT	VIT	0,562	22
168	Curativo	Lagarta	BEL	COR	0,609	12

Após uma análise dos resultados, observa-se que os tempos de execução foram sempre menores do que 1 segundo. No entanto, quantas mais observações eram inseridas, mais tempo o sistema demorava a gerar os modelos, pois são mais factos a serem trabalhados pelo *clingo*. Acrescenta-se que as observações relativas ao Míldio geram sempre muitos modelos relativamente às outras, pois é uma doença que possui muitas alternativas para o seu tratamento. Além disso, o sistema aconselhou sempre o produto VIT para essa doença, pois é um produto declarado como preferência de vendedor.

Com todas as observações inseridas, juntamente com os produtos utilizados nas mesmas, o sistema teve um resultado dos custos indicados na tabela 6.16.

Tabela 6.16: Custo/margem total do plano original do cenário 6

Dia	Produto	Custo/ha	Margem/ha
74	SEN WG	44,62€	4,46€
93	ACT 25 WG	38,50€	3,85€
93	COS WDG	5,20€	0,52€
93	KAD	37,50€	3,75€
93	MIC	62,49€	6,25€
105	CYT 10 EC	37,00€	3,70€
105	MEL	48,50€	4,85€
115	FAS	16,30€	1,63€
115	STE	48,05€	4,81€
115	VIT	40,00€	4,00€
128	ALT	74,75€	7,48€
128	VIT	40,00€	4,00€
143	PYR 48 EC	24,75€	2,48€
143	RAN	35,13€	3,51€
155	BAG	7,60€	0,76€
168	COR	70,04€	7,00€
168	VIT	40,00€	4,00€
Totais:		670,43€	67,05 €

Procedeu-se para o teste ao desempenho do sistema, com a maioria dos produtos utilizados removidos. O sistema resultou do plano representado pela tabela 6.17.

Tabela 6.17: Tratamentos substituídos no cenário 6

Dia	Tipo	Inimigo	Substância ativa	Produto
93	Preventivo	Oídio	enxofre	KUM S
93	Preventivo	Míldio	cimoxanil+folpete	VIT
105	Curativo	Mosca Branca	lambda-cialotrina	KAR+
105	Curativo	Míldio	cimoxanil+folpete	VIT
115	Curativo	Lagarta	Bacillus thuringiensis	BEL
115	Curativo	Mosca Branca	lambda-cialotrina	KAR+
115	Curativo	Míldio	cimoxanil+folpete	VIT
128	Curativo	Lagarta	Bacillus thuringiensis	BEL
128	Curativo	Míldio	cimoxanil+folpete	VIT
143	Curativo	Lagarta	Bacillus thuringiensis	BEL
143	Preventivo	Míldio	cimoxanil+folpete	VIT
155	Preventivo	Oídio	enxofre	ALA MICRO
168	Curativo	Lagarta	Bacillus thuringiensis	BEL
168	Curativo	Míldio	cimoxanil+folpete	VIT

Através dos resultados obtidos, nota-se que o produto VIT foi aconselhado em todos os tratamentos do Míldio, graças ao produto ser uma preferência de vendedor. Por outro lado, o produto BEL também foi bastante aconselhado para minimizar o número de produtos e substâncias ativas utilizadas.

A tabela 6.18 contém os custos e margens que resultaram deste plano. Novamente confirma-se que ao colocar a minimização de custos com uma prioridade superior à de maximização de margens, obtém-se um plano mais económico que o original.

Para obter o plano ilustrado, o sistema gerou 269 modelos diferentes em apenas 6 segundos. A rapidez com que foi detetado o modelo ótimo deve-se ao facto de existir uma

Tabela 6.18: Custo/margem total do novo plano do cenário 6

Dia	Produto	Custo/ha	Margem/ha
74	SEN WG	44,62€	4,46€
93	ACT 25 WG	38,50€	3,85€
93	KAD	37,50€	3,75€
93	KUM S	3,64€	0,36€
93	VIT	40,00€	4,00€
105	KAR+	8,59€	0,86€
105	VIT	40,00€	4,00€
115	BEL	13,74€	1,37€
115	KAR+	8,59€	0,86€
115	VIT	40,00€	4,00€
128	BEL	13,74€	1,37€
128	VIT	40,00€	4,00€
143	BEL	13,74€	1,37€
143	VIT	40,00€	4,00€
155	ALA MICRO	6,50€	0,65€
168	BEL	13,74€	1,37€
168	VIT	40,00€	4,00€
Totais:		442,90€	44,27€

preferência de um produto para tratar o Míldio, como é o caso do VIT, pois é uma doença que possui muitas alternativas e, assim, o sistema consegue escolher mais facilmente o produto a aconselhar ao utilizador.

6.3 Conclusão

Após os seis cenários terem sido testados no sistema desenvolvido, pode-se concluir que os resultados são satisfatórios relativamente ao modo como são feitas as otimizações e se obtém o resultado mais conveniente às necessidades do utilizador. No entanto, ainda existem alguns pontos negativos por melhorar.

Devido aos longos tempos de execução que se sofreram nos vários testes, em certos cenários teve de se reduzir o número de tratamentos retirados para serem utilizados no teste ao desempenho, especialmente tratamentos ao Míldio, que possui muitas possibilidades de tratamento diferente, como foi o caso do primeiro cenário. No entanto, através do cenário 6, provou-se que não é um ponto negativo muito grave, uma vez que o sistema foi desenvolvido para ser utilizado de forma progressiva, onde os técnicos vão inserindo as observações aos poucos e, assim, os resultados surgem de forma bastante rápida, uma vez que não são muitas observações a considerar de cada vez.

Neste cenário concluiu-se também que existe muitas vantagens com a utilização de preferências de vendedor/comprador, quanto aos tempos de execução. Desta forma, o sistema consegue escolher os produtos a aconselhar de forma mais rápida, caso a otimização pelas preferências tenham uma prioridade altera relativamente às outras, mesmo para os inimigos que contenham muita diversidade de produtos de possível utilização.

Através cenário 3 notou-se a falta de produtos registados no sistema, especialmente para os tratamentos preventivos às lagartas. A tarefa de completar a lista de produtos para cada substância ativa não é trivial, pois existem muitos produtos comerciais no

mercado, bem como muitas empresas fornecedoras dos mesmos.



Conclusão e trabalhos futuros

Neste documento é apresentado o desenvolvimento de um sistema de aconselhamento de aplicações de produtos fitofarmacêuticos. Demonstrou-se que o problema inicial é muito complexo devido à grande quantidade de informação diferente que foi necessário integrar, bem como a forma de estruturar o sistema para que possibilite a geração de planos de tratamento de forma automática.

São apresentados diversos modelos que expressam a forma como está organizado o sistema em relação aos dados armazenados, aos componentes da aplicação desenvolvida e às funcionalidades oferecidas. Além disso, também foi esclarecido o modo como foi reduzida a integração com as tecnologias utilizadas, fazendo com que as diversas execuções das diferentes componentes seja transparente para o utilizador.

Foram listadas as diferentes maneiras possíveis de exprimir regras lógicas que permitem a geração de planos de tratamento de forma automática, juntamente com regras de Proteção Integrada, as doses e os custos dos produtos utilizados. Demonstrou-se muitas vantagens no uso da linguagem de programação por conjuntos de resposta para o desenvolvimento da aplicação, fazendo com que não haja necessidade de proceder à criação de algoritmos específicos, para geração e otimização de vários planos de tratamento. No entanto, devido à linguagem não exigir uma ordem nas regras implementadas e poder haver várias relações entre elas, assim como, várias regras com a mesma propriedade, mas com argumentos diferentes, é necessário um certo cuidado na organização das regras lógicas.

O comportamento do sistema implementado foi testado através de vários cenários de tratamento da cultura do tomate diferentes. Verificou-se que na maioria dos casos, o sistema tem bom desempenho para o cenário esperado de utilização.

Como trabalho futuro está planeado adicionar um novo fator de otimização quanto

à quantidade de produtos a utilizar numa determinada área limitada de cultivo, medida em hectares, consoante o tamanho dos pacotes de produtos fitofarmacêuticos existentes, bem como os que se encontram disponíveis numa determinada empresa fornecedora. Este fator facilita ao utilizador para saber a quantidade certa que deve comprar/vender para os seus tratamentos. Ao mesmo tempo melhora o cálculo dos planos de tratamentos gerados.

Além disso, é necessária a inserção de todos os produtos em falta para as substâncias ativas existentes, bem como a extensão das regras de Proteção Integrada, substâncias ativas e produtos para as outras culturas existentes e listadas em documentos disponibilizados pela DGAV, que não foram incluídas nesta dissertação.

Por fim, está planeado a instalação do sistema num servidor *web*, com registos de utilizadores para ser posta em produção e facilmente ser acedido por diversos utilizadores em qualquer computador com acesso à Internet.

Bibliografia

- [1] G. Antoniou e F. V. Harmelen, *A semantic web primer*, 2ª ed. The MIT Press, 2004, ISBN: 9780262012423. endereço: http://books.google.com/books?hl=en&lr=&id=cIYAis1wbRsC&oi=fnd&pg=PR13&dq=A+semantic+web+primer&ots=i9ASCuTizv&sig=jbpV_B8Q7tjFAF-1KQHbOMdraIA.
- [2] C. Baral, *Knowledge representation, reasoning and declarative problem solving with Answer sets*. Cambridge University Press, 2003, ISBN: 0521818028.
- [3] H. Beck, "Agricultural enterprise information management using object databases, Java, and CORBA", *Computers and Electronics in Agriculture*, vol. 32, nº 2, pp. 119–147, ago. de 2001, ISSN: 01681699. DOI: 10.1016/S0168-1699(01)00162-4. endereço: <http://linkinghub.elsevier.com/retrieve/pii/S0168169901001624>.
- [4] G. Brewka, T. Eiter e M. Truszczyński, "Answer Set Programming at a Glance", *Communications of the ACM*, vol. 54, nº 12, pp. 92–103, dez. de 2011. DOI: 10.1145/2043174.2043195. endereço: <http://people.scs.carleton.ca/~bertossi/KR11/material/communications201112ASP.pdf>.
- [5] E. F. Codd, "A relational model of data for large shared data banks", *Commun. ACM*, vol. 13, nº 6, pp. 377–387, jun. de 1970, ISSN: 0001-0782. DOI: 10.1145/362384.362685. endereço: <http://doi.acm.org/10.1145/362384.362685>.
- [6] S. Das, S. Sundara e R. Cyganiak, "R2RML: RDB to RDF Mapping Language", W3C, W3C Recommendation, set. de 2012. endereço: <http://www.w3.org/TR/2012/REC-r2rml-20120927/>.
- [7] I. Davis e T. Steiner, "RDF 1.1 JSON Serialisation (RDF/JSON)", W3C, W3C Draft, jan. de 2013. endereço: <https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-json/index.html>.

- [8] DGADR, “Atualização dos produtos fitofarmacêuticos aconselhados em proteção integrada”, pp. 1–31, 2012. endereço: http://www.dgv.min-agricultura.pt/xeov21/attachfileu.jsp?look_parentBoui=4548119&att_display=n&att_download=y.
- [9] —, “Cancelamento de AVs e APVs”, rel. téc., 2012, pp. 1–26. endereço: http://www.dgv.min-agricultura.pt/xeov21/attachfileu.jsp?look_parentBoui=4083888&att_display=n&att_download=y.
- [10] I. Füzesi e M. Herdon, “EDI - XML Standards and Technologies in the Agri-Food Industry”, *Summer University on Information Technology in Agriculture and Rural Development*, pp. 1–10, dez. de 2007. endereço: http://odin.agr.unideb.hu/su2007/papers/S2/S25_Fuzesi_Herdon.pdf.
- [11] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub e M. Schneider, “Potassco: the Potsdam answer set solving collection”, vol. 24, nº 2, pp. 105–124, 2011.
- [12] M. Gebser, B. Kaufmann, A. Neumann e T. Schaub, “Conflict-driven answer set solving”, em *Twentieth International Joint Conference on Artificial Intelligence*, MIT Press, 2007, pp. 386–392.
- [13] M. Gebser, T. Schaub e S. Thiele, “S.: gringo: a new grounder for answer set programming”, em *In Baral et al*, 2007, pp. 266–271.
- [14] M. Gelfond, “Answer Sets”, pp. 1–31, 2007. endereço: <http://www.depts.ttu.edu/cs/research/krlab/pdfs/papers/gel07b.pdf>.
- [15] G. A. Grimnes, M. Abufouda, M. Kiesel e A. Schröder, “Semantic Integration through Linked Data in the iGreen project”, em *GIL Jahrestagung*, 2012, pp. 107–110. endereço: http://gil-net.de/Publikationen/24_107.pdf.
- [16] R. V. Guha e D. Brickley, “RDF Vocabulary Description Language 1.0: RDF Schema”, W3C, W3C Recommendation, fev. de 2004. endereço: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- [17] IGreen. (2012). data.iGreen-Services.com. Acedido em: 5/11/2012, endereço: <http://data.igreen-services.com/>.
- [18] M. Kunisch, J. Frisch, D. Martini e S. Böttinger, “agroXML – a standardized language for data exchange in agriculture”, em *6th Biennial Conference of the European Federation of IT in Agriculture CDROM*, 2007. endereço: http://www.itfoodtrace.com/dateien/EFITA_Kunisch_et_al.pdf.
- [19] M. Kunisch e D. Martini, “agroXML - A Standard for Data Exchange in Agriculture”, em *The Future of Information Exchange and ICT in Agri-Food business*, 2011. endereço: http://www.agrixchange.org/sites/default/files/1445-1500Kunisch111005_KTBL_agroXML_final.pdf.

- [20] A. Lopes e A. M. Simões, *Produção Integrada em Hortícolas - Família das Solanáceas*, DGADR. 2006, ISBN: 9728649533. endereço: http://www.dgadr.pt/images/docs/prod_sust/i008495.pdf.
- [21] D. L. McGuinness e F. van Harmelen, "OWL Web Ontology Language Overview", W3C, W3C Recommendation, fev. de 2004. endereço: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [22] E. Miller e F. Manola, "RDF Primer", W3C, W3C Recommendation, fev. de 2004. endereço: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
- [23] B. Mira. (2011). AJASUL | Agricultores do Sul - proteção integrada do olival. Acedido em: 27/1/2012, endereço: <http://www.ajasul.com/index.php/proteccai-olival>.
- [24] B. Monograph e F. Biological, "Growth stages of mono-and dicotyledonous plants BBCH Monograph Edited by Uwe Meier Federal Biological Research Centre for Agriculture and Forestry", *Agriculture*, vol. 12, U. Meier, ed., 141–147 ST Geochemical study of the organic mat–, 2001, ISSN: 00791946. endereço: <http://pub.jki.bund.de/index.php/BBCH/article/view/515/464>.
- [25] B. Motik, A. Fokoue, I. Horrocks, Z. Wu, C. Lutz e B. C. Grau, "OWL 2 Web Ontology Language Profiles", W3C, W3C Recommendation, out. de 2009. endereço: <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>.
- [26] "Normes OEPP EPPO Standards Data sheets on quarantine pests Fiches informatives sur les organismes de quarantaine", *EPPO Bulletin*, vol. 38, nº 1, pp. 53–54, abr. de 2008, ISSN: 1365-2338. DOI: 10.1111/j.1365-2338.2008.01181.x. endereço: <http://doi.wiley.com/10.1111/j.1365-2338.2008.01181.x>.
- [27] "Normes OEPP EPPO Standards Efficacy evaluation of plant protection products Evaluation biologique des produits phytosanitaires", *EPPO Bulletin*, vol. 42, nº 3, pp. 345–347, dez. de 2012, ISSN: 1365-2338. DOI: 10.1111/j.1365-2338.1012.02583.x. endereço: <http://dx.doi.org/10.1111/j.1365-2338.1012.02583.x>.
- [28] Oracle. (2010). History of Java Technology. Acedido em: 8/1/2013, endereço: <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>.
- [29] F. J. Orellana, J. del Sagrado e I. M. Del Águila, "SAIFA: A web-based system for Integrated Production of olive cultivation", *Computers and Electronics in Agriculture*, vol. 78, nº 2, pp. 231–237, set. de 2011, ISSN: 01681699. DOI: 10.1016/j.compag.2011.07.014. endereço: <http://linkinghub.elsevier.com/retrieve/pii/S0168169911001773>.
- [30] T. Research, "Java Application Servers Report", rel. téc., 1999. endereço: <http://www.fscript.org/prof/javapassport.pdf>.

- [31] A. Seaborne e S. Harris, "SPARQL 1.1 Query", W3C, W3C Working Draft, out. de 2009. endereço: <http://www.w3.org/TR/2009/WD-sparql11-query-20091022/>.
- [32] A. Silva, A. Prates, F. Bento, L. Gaspar e M. Cavaco, "Nomenclatura dos inimigos das culturas para as quais se admite o recurso à luta química", Direção-Geral de Agricultura e Desenvolvimento Rural, Lisboa, rel. téc., 2012, p. 149. endereço: <http://hortinet.info/wp-content/uploads/group-documents/71/1360236496-NomenclaturadosInimigosdasCulturas.pdf>.



Anexos

A.1 Datas oficiais para comercialização e utilização dos produtos

Nome Comercial	Substância Ativa	Empresa	Apv	Av	Data Av/Apv	Data Cancelamento	Data Limite para Comercialização	Data Limite para Utilização
A SCALD	difenilamina	DECCO	3498		22/09/2003	30/05/2010	30/05/2011	30/05/2011
AAPROTURON	isoproturão	AAKO B.V.	0087		16/04/2008	17/02/2009	-	17/02/2010
ABAMACTINA SELECTIS	abamectina	SELECTIS	3777		05/09/2006	18/09/2009	31/10/2009	31/10/2010
ACARICIDA DUPLO ORMENTAL	dicofoi+tetradifão	SIPCAM QUIMAGRO	2400		07/02/1992	31/12/2002	31/12/2003	31/12/2003
ACARICIDA TOTAL PERMUTADORA	dicofoi+tetradifão	PERMUTADORA	1326		10/12/1975	31/12/2002	31/12/2003	31/12/2003
ACARIX	cihexaestanho	NUFARM P	3535		26/03/2004	04/10/2008	04/10/2009	04/10/2010
ACAROX	cihexaestanho	AGROQUISA	2265		06/09/1990	06/09/2008	04/10/2009	04/10/2010
ACARPEC	cihexaestanho	SAPEC	1470		23/04/1979	23/04/2008	04/10/2009	04/10/2010
ACARPEC 600 FL	cihexaestanho	SAPEC	3529		06/02/2004	04/10/2008	04/10/2009	04/10/2010
ACARSTIN	cihexaestanho	INGRA	2432		22/10/1992	04/10/2008	04/10/2009	04/10/2010
ACARTAL T	dicofoi+tetradifão	SAPEC AGRO	1148		28/11/1972	31/12/2002	31/12/2003	31/12/2003
ACARTOT	dicofoi+tetradifão	AGROQUISA	2238		05/04/1990	31/12/2002	31/12/2003	31/12/2003

A.2 Substâncias ativas

FUNGICIDAS							
Substância ativa	Form.	Concentração (g s.a. / hl)	IS Dias	Nome comercial	Classif.	Nº de aplicações	Condições de utilização
mildio (cont.)							
cobre (hidróxido) (8)	WG	70-200	7	KADOS, KOCIDE 35 DF, KOCIDE 2000, KOLECTIS (*), VITRA 40 MICRO, KOCIDE OPTI	Xn; N		(8) O produto tem ação inibidora em bactérias que favorecem a formação de gelo. A aplicação antes da existência de condições de geada, nas concentrações indicadas, pode proteger geadas fracas. Não se recomenda em áreas e locais onde as condições sejam favoráveis a geadas fortes.
	WP	125-250	7	CHAMPION WP, COBRE HIDRÓXIDO ADP FITOCOBRE, GYPSY 50 WP, HIDROTEC 50% WP, MACC 50	Xi; N		
cobre (oxicloreto) + iprovalicarbe (15) ♣	WG	126+609	(16)	MELODY COBRE	Xn; N	Máximo duas aplicações	(15) Não efetuar mais que 3 tratamentos, por campanha com este ou outro fungicida com o mesmo modo de ação (fungicida CAA), quando aplicável. ♣ Ver critério referente a informação insuficiente relativa aos auxiliares. (16) 20 dias ao ar livre e 7 dias em estufa.
cobre (oxicloreto) + propinebe ♣	WP	52,5Cu+105-70Cu+140	(17)	CUPROSAN P	Xn; N	Máximo duas aplicações	(17) Por força da circular DSPFSV (H/C)-2/2008, não efetuar mais de 4 aplicações; intervalo de segurança 7 dias ao ar livre, quando a produção se destine ao consumo em fresco; 28 dias em estufa e quando a produção se destine a fins industriais. ♣ Ver critério referente a informação insuficiente relativa aos auxiliares.
cobre (sulfato de cobre e cálcio – mistura bordalesa)	WP	250-520	7	CALDA BORDEAUX CAFFARO 13, CALDA BORDALESA CAFFARO 20, CALDA BORDALESA QUIMAGRO, CALDA BORDALESA RSR	Xn, N	----	
				CALDA BORDALESA NUFARM, CALDA BORDALESA VALLES, CALDA BORDALESA QUIMIGAL (APV 3852)	Xi, N		
				CALDA BORDALESA QUIMIGAL (APV 2211) (*)	Xi		
				CALDA BORDALESA SAPEC, CALDA BORDALESA SELECTIS	N		
cobre (sulfato de cobre (tribásico))	SC	247 (9)	7	CUPROXAT	N	---	(9) Aplicar ao aparecimento dos primeiros sintomas e repetir se necessário
dimetomorfe+ mancozebe (15) # ♣	WG	18+160,08	(5)	PARA-AT	Xi, N	Máximo duas aplicações	(15) Não efetuar mais que 3 tratamentos, por campanha com este ou outro fungicida com o mesmo modo de ação (fungicida CAA), quando aplicável. ♣ Ver critério referente a informação insuficiente relativa aos auxiliares.
	WP	18+160	(5)	ACROBAT MZ (*)	Xn; N		
dimetomorfe + piraclostrobina (29)	EC	14,4+8 - 18+10	3	CABRIO DUO	Xn; N	Máximo duas aplicações	(29) Não efetuar mais de 2 aplicações, com esta mistura. E não efetuar mais de 3 para o conjunto das doenças com os fungicidas CAA (dimetomorfe) e para o conjunto das doenças com os fungicidas QOI (piraclostrobina).
folpete (2)	SC	113,5	(3)	FOLPAN 500 SC, FOLTENE (*)	Xn; N	Máximo duas aplicações em estufa	(2) Em estufa não efetuar mais de 2 tratamentos com produtos que contenham captana ou folpete. (3) 7 dias ao ar livre e 14 dias em estufa.
	WG	120	(3)	AKOFOL 80 WDG (*), FOLPAN 80 WDG, FOLPETIS WG	Xn; N		
	WP	125	(3)	AKOFOL 50 WP (*), ORTHO PHALTAN (*), BELPRON F-50, FOLPAN 50 WP AZUL (*), FOLPEC 50, FOLPEC 50 AZUL	Xn; N		
<p># São admitidas até duas aplicações consecutivas de produtos fitofarmacêuticos com base em substâncias ativas do grupo dos ditiocarbamatos, simples ou em mistura. Uma terceira aplicação deve ser separada das anteriores de pelo menos três semanas. Durante o ciclo cultural, o somatório do número de aplicações com fungicidas que contêm ditiocarbamatos não pode ser superior a quatro, quando aplicável. Aconselha-se a leitura atenta da Circular DSPFSV (H/C)-2/2008 que complementa a informação sobre ditiocarbamatos.</p> <p>(*) Consultar lista de cancelamento de uso para produtos comerciais disponível no site da DGADR www.dgadr.pt.</p>							

A.3 Níveis económicos de ataque

Larvas mineiras Diptera Agromyzidae Liriomyza spp.					
Estimativa do risco		Tomada de decisão	Meios de luta		
Método de amostragem e órgãos a observar	Outros métodos		Biológico	Químico	Cultural
			Fauna auxiliar Nº de largadas e periodicidade		
Em cultura protegida: Observar 3 folhas em especial no nível médio e inferior. Contar número de folhas com picadas e galerias. Colher algumas folhas com galerias para contabilização dos parasitoides e larvas mortas.	Colocar armadilhas cromotrópicas amarelas.	Em cultura protegida: 1 - À presença, dos primeiros adultos nas armadilhas, primeiras picadas de alimentação, existência de galerias nas folhas e presença larvas, largar auxiliares.			Em cultura protegida e de ar livre: - utilizar plantas sãs. - colocar redes anti-insectos nas aberturas (em cultura protegida); - colocar armadilhas para captura em massa (em cultura de ar livre); - eliminar as infestantes; - eliminar as folhas mais baixas da planta; - eliminar os restos da cultura; - realizar solarização (em cultura protegida); - realizar rotações culturais
		① Se se observar <1 larva /10 plantas, largar auxiliares	● Largar 0,25 <i>Dacnusa sibirica</i> + <i>Diglyphus isae</i> /m ² , realizando 3 largadas curativas baixas, em intervalos semanais.		
		② Se se observar > 1 larva/10 plantas, largar auxiliares	● Largar 0,25-0,5 <i>Diglyphus isae</i> /m ² , realizando um mínimo de 3 largadas curativas altas, em intervalos semanais.		
		③ Se se verificarem infestações elevadas de mineira.	● Largar 0,1 <i>Diglyphus isae</i> /m ² , em largada curativa baixa, realizando um mínimo de 3 largadas em intervalos semanais.		
		2 - Se se verificar um aumento da população, tratar.		abamectina ciromazina	
Em cultura de ar livre: Observar a planta e pesquisar a existência de galerias e de picadas de alimentação.		Em cultura de ar livre: Logo que se detectem adultos nas armadilhas e minas na planta, tratar, tendo em consideração os organismos auxiliares presentes no campo de produção.		abamectina ciromazina	
Obs.: substância activa recomendada substância activa complementar					