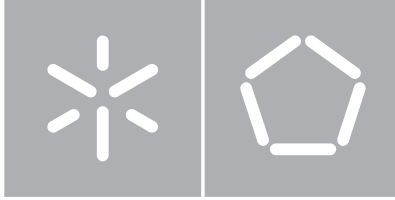


**Universidade do Minho**  
Escola de Engenharia

António José Linhares da Silva

**Representação de Informação Temporal  
em Sistemas de Apoio à Decisão Clínica**



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

António José Linhares da Silva

**Representação de Informação Temporal  
em Sistemas de Apoio à Decisão Clínica**

Dissertação de Mestrado

Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

**Professor Paulo Jorge Freitas de Oliveira Novais**

**Tiago José Martins Oliveira**

*É autorizada a reprodução integral desta tese apenas para efeitos de investigação, mediante declaração escrita do interessado, que a tal se compromete.*

## **Title**

Representation of Temporal Information in Clinical Decision Support Systems.

## **Abstract**

Temporal information is one of the fundamental aspects of decision making in the clinical context. The decisions of health care professionals are made based on the duration, periodicity and frequency of observations. The very nature of documents such as clinical protocols is based on temporal logic, in a continuous evolution of the patient which should be monitored over time.

Different moments require particular measures. However, in the domain of Clinical Decision Support Systems and Computer-Interpretable Guideline (CIG) models, these temporal instants are not easy to model, especially when there is the need for a representation capable of accommodating a large variety of temporal patterns. A temporal pattern is understood as any restriction placed on the execution of clinical tasks or conditions about the state of a patient used in rules for inference. It was possible to verify that the main temporal patterns observable in clinical protocols belong to a set consisting of durations, periodicities, waiting times and temporal restrictions on conditions about the state of a patient. In existing CIG models there are significant shortcomings when it comes to the representation of periodic events and restrictions about the state of a patient. Furthermore, some CIG models do not possess appropriate temporal constructors and adopt parts of existing languages to represent temporal patterns, which means that they also assimilate the limitations of said languages.

To address these limitations a new temporal representation model was developed for the CompGuide CIG model. This new model is capable of accommodating the majority of temporal patterns in clinical protocols. Moreover, an execution engine for machine-readable clinical protocols represented in CompGuide was developed, along with an application that allows the user to execute and visualize protocols in a temporal context. These represent significant advancements to the current state of the art.

## **Título**

Representação de Informação Temporal em Sistemas de Apoio à Decisão Clínica.

## **Resumo**

A informação temporal é um dos aspetos fundamentais da tomada de decisão em contexto clínico. As decisões de profissionais de saúde são tomadas com base na duração, periodicidade e frequência das observações. Assim, a própria natureza de documentos como protocolos clínicos assenta numa lógica temporal, numa evolução contínua do paciente que deve ser monitorizada ao longo do tempo.

Instantes diferentes requerem medidas particulares, contudo no domínio dos Sistemas de Apoio à Decisão Clínica e dos modelos de Computer-Interpretable Guidelines (CIG), estes instantes temporais não são fáceis de modelar, principalmente quando se pretende uma representação capaz de acomodar uma grande variedade de padrões temporais. Entende-se por padrão temporal qualquer restrição temporal que é colocada à execução de tarefas clínicas ou a condições sobre o estado do paciente que são utilizadas em regras para inferência.

É possível verificar que os principais padrões temporais que se podem observar em protocolos clínicos integram um conjunto composto por durações, periodicidades, tempos de espera e restrições temporais em condições relacionadas com o estado do paciente. No entanto, nos modelos CIG atuais existem lacunas significativas principalmente ao nível da representação de eventos periódicos e de restrições sobre condições do estado do paciente. Para além disso, alguns modelos CIG não possuem construtores temporais apropriados e adotam partes de linguagens existentes para representar padrões temporais, o que faz com que assimilem também as limitações dessas linguagens.

Como resposta a estas lacunas desenvolveu-se um modelo de representação temporal para o modelo CIG CompGuide que se revelou capaz de acomodar os padrões temporais existentes em protocolos clínicos. Adicionalmente, desenvolveu-se um motor de execução para versões digitais de protocolos clínicos representados no modelo CompGuide e uma aplicação que permite a um utilizador executar e visualizar os protocolos num contexto temporal, o que representa avanços relativamente ao estado da arte atual.

## Índice

Title .....	iii
Abstract.....	iii
Título .....	iv
Resumo.....	iv
Notação e Terminologia .....	viii
<i>Notação</i> .....	viii
<i>Acrónimos</i> .....	viii
Índice de Figuras .....	x
Índice de Tabelas.....	xii
1 Introdução .....	1
1.1 Enquadramento e Motivação.....	1
1.1.1 Sistemas de Apoio à Decisão Clínica .....	2
1.1.2 Protocolos Clínicos.....	5
1.2 Tema e Objetivos .....	6
1.3 Método de Trabalho e Investigação.....	7
1.4 Estrutura do Documento .....	8
2 Computer-Interpretable Guidelines .....	10
2.1 Importância, Vantagens e Desvantagens.....	10
2.2 Principais Abordagens à Modelação .....	12
2.2.1 Arden Syntax.....	13
2.2.2 GLIF .....	13
2.2.3 Asbru.....	14
2.2.4 EON.....	15
2.2.5 PROforma .....	17
2.2.6 GLARE .....	18

2.3	Importância da Representação Temporal no Raciocínio Médico.....	20
2.4	Discussão e Análise dos Modelos de Representação Temporal.....	22
3	Proposta de um Modelo de Representação Temporal .....	27
3.1	Ontologia CompGuide .....	27
3.2	Modelo de Representação Temporal.....	30
3.2.1	Restrições Temporais à Execução de Tarefas.....	32
3.2.2	Restrições Temporais sobre o Estado do Paciente .....	37
3.3	Discussão e Análise do Modelo Proposto .....	39
4	Análise do Problema de Execução de Protocolos Clínicos.....	42
4.1	Modelo de Domínio	42
4.2	Apresentação dos Atores do Sistema .....	45
4.3	Análise e levantamento de Requisitos .....	46
4.3.1	Requisitos Funcionais.....	46
4.3.2	Requisitos Não Funcionais .....	47
4.4	Casos de Uso .....	48
4.4.1	Diagrama de Casos de Uso .....	48
4.4.2	Descrição dos Casos de Uso .....	51
4.5	Discussão e Análise .....	53
5	Desenvolvimento e Implementação do Sistema de Execução de Protocolos Clínicos....	54
5.1	Tecnologias e Ferramentas Utilizadas.....	54
5.2	Arquitetura de Software.....	55
5.3	Modelo Relacional.....	58
5.4	Diagrama de Classes .....	58
5.5	Diagramas de Sequência.....	62
5.6	Desenvolvimento Baseado em Padrões de Software .....	65
5.7	Apresentação da Aplicação Web .....	67

5.8 Discussão e Análise da Solução .....	70
6 Conclusões e Trabalho Futuro .....	75
6.1 Objetivos Realizados .....	75
6.2 Limitações e Trabalho Futuro .....	76
6.3 Divulgação de Resultados.....	77
Referências.....	78
Anexos .....	83
A. Descrição dos Casos de Uso .....	84
B. Modelo Relacional.....	99
C. Diagramas de Sequência .....	100
D. Apresentação da Aplicação Web.....	105



## Notação e Terminologia

### ***Notação***

Ao longo do documento utiliza-se texto em itálico para palavras em língua estrangeira. Alguns termos não foram traduzidos, nomeadamente os que dizem respeito aos nomes modelos de representação de protocolos clínicos e a construtores específicos que fazem parte destes modelos, de forma a representá-los de forma fiel.

### ***Acrónimos***

API	Application Programming Interface
CIG	Computer-Interpretable Guidelines
GLIF	Guideline Interchange Format
CPG	Clinical practice guidelines (protocolos clínicos de prática clínica)
GLARE	Guideline Acquisition, Representation and Execution
HL7	Health Level Seven
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IsLab	Intelligent Systems Lab
JSON	Javascript Object Notation
JPA	Java Persistence API
JSF	Java Server Faces
MLM	Medical Logic Module
MVC	Model-View-Controller
NCCN	National Comprehensive Cancer Network

OWL	Ontology Web Language
RSE	Registo de Saúde Eletrónico
SADC	Sistema de Apoio à Decisão Clínica
SAGE	Standards-Based Sharable Active Guideline Environment
SGBD	Sistema de Gestão de Base de Dados
STP	Simple Temporal Problem
TNM	Task Network Model
UML	Unified Modelling Language
$R^2L$	Red Representation Language
XML	Extensible Markup Language

## Índice de Figuras

Figura 1 - Modelo Geral dos Sistemas de Apoio à Decisão Clínica. ....	3
Figura 2 - Representação esquemática das principais classes do modelo EON. ....	16
Figura 3 - Representação esquemática das principais classes de tarefas do modelo PROforma. ....	17
Figura 4 - Entidades básicas definidas no modelo de representação de GLARE (figura retirada de [46]).....	19
Figura 5 - Arquitetura Geral de GLARE (figura retirada de [46]). ....	19
Figura 6 - Definição inicial de um protocolo clínico da National Comprehensive Cancer Network para o tratamento de cancro do cólon na ontologia CompGuide. ....	28
Figura 7 - Definição de um protocolo clínico do National Comprehensive Cancer Network para o tratamento de cancro do cólon na ontologia CompGuide, evidenciando as propriedades que permitem estabelecer uma ordem relativa entre as tarefas: hasFirstTask, nextTask, alternativeTask e parallelTask .....	30
Figura 8 - Diagrama das classes OWL do modelo temporal na ontologia CompGuide. ....	31
Figura 9 - Representação esquemática da classe WaitingTime. ....	33
Figura 10 – Representação esquemática da classe Periodicity.....	37
Figura 11 - Modelo de domínio. ....	43
Figura 12 - Diagrama de casos de uso relativo ao package Gestão de Profissionais de Saúde. .	49
Figura 13 - Diagrama de casos de uso relativo ao package Gestão de Guidelines. ....	49
Figura 14 - Diagrama de casos de uso relativo ao package Gestão de Tarefas.....	50
Figura 15 - Diagrama de casos de uso relativo ao package Gestão de Pacientes. ....	50
Figura 16 - Arquitetura típica segundo um modelo Model-View-Controller (figura retirada de [63]). .....	56
Figura 17 - Arquitetura do sistema para execução de protocolos clínicos.....	57
Figura 18 - Diagrama de classes do package Persistence Gestão de Tarefas.....	60
Figura 19 - Diagrama de classes do package Controllers Gestão de Tarefas.....	61
Figura 20 - Diagrama de classes do package SessionBeans Gestão de Tarefas. ....	62
Figura 21 - Diagrama de sequência Processar Aspectos Temporais.....	63
Figura 22 - Diagrama de sequência Processar Tempo de Espera. ....	64
Figura 23 - Diagrama de classes Integração do Padrão Adapter.....	66
Figura 24 - Diagrama de classes Integração do Padrão Facade.....	67

Figura 25 - Página de login. ....	68
Figura 26 - Página de registo. ....	68
Figura 27 - Página cronograma com as tarefas.....	69
Figura 28 - Página calendário com tarefas.....	69
Figura 29 - Página de informação detalhada da tarefa .....	70
Figura 30 - Caixa de mensagens e pop-ups de notificação. ....	71
Figura 31 - Cronograma de tarefas.....	72
Figura 32 - Calendário de tarefas .....	72
Figura 33 - Modelo relacional.....	99
Figura 34 - Diagrama de sequência Processar Durações. ....	100
Figura 35 - Diagrama de sequência Processar Periodicidades.....	102
Figura 36 - Diagrama de sequência Processar Parte de Cada Ciclo.....	103
Figura 37 - Diagrama de sequência Processar Periodicidade de Cada Parte de um Ciclo. ....	104
Figura 38 - Página de visualização da informação pessoal. ....	105
Figura 39 - Página de edição da informação pessoal. ....	106
Figura 40 - Página criação de Profissionais de Saúde. ....	106
Figura 41 - Página edição de Profissionais de Saúde. ....	107
Figura 42 - Página eliminar Profissionais de Saúde.....	107
Figura 43 - Página para a gestão da informação de uma tabela da Base de Dados. ....	108
Figura 44 - Separador para gerir os protocolos clínicos.....	108
Figura 45 - Separador para gerir os Pacientes. ....	108

## Índice de Tabelas

Tabela 1 - Restrições temporais apresentadas nos modelos de representação de protocolos clínicos.....	26
Tabela 2 – Número de ocorrências de cada tipo de tarefa no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide.....	39
Tabela 3 – Número de ocorrências de cada tipo restrição temporal à execução de tarefas no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide. ....	40
Tabela 4 – Número de ocorrências das restrições temporais sobre condições do estado do paciente no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide. ....	41
Tabela 3 - Descrição do caso de uso Pesquisar Profissional de Saúde. ....	51
Tabela 4 – Descrição do caso de uso Pesquisar Protocolos clínicos. ....	52
Tabela 5 - Descrição caso de uso Pesquisar Paciente. ....	84
Tabela 6 - Descrição caso de uso Adicionar Paciente.....	85
Tabela 7 - Descrição caso de uso Criar Protocolos clínicos .....	86
Tabela 8 - Descrição caso de uso Adicionar Profissional de Saúde .....	87
Tabela 9 - Descrição caso de uso Editar Profissional de Saúde. ....	88
Tabela 10 - Descrição do caso de uso Editar Protocolos clínicos. ....	89
Tabela 11 - Descrição do caso de uso Editar Paciente. ....	90
Tabela 12 - Descrição do caso de uso Listar Profissionais de Saúde. ....	91
Tabela 13 - Descrição do caso de uso Listar Protocolos clínicos.....	92
Tabela 14 - Descrição do caso de uso Listar Pacientes. ....	92
Tabela 15 - Descrição do caso de uso Remover Profissional de Saúde. ....	93
Tabela 16 - Descrição do caso de uso Remover Protocolos clínicos. ....	93
Tabela 17 - Descrição do caso de uso Remover Paciente.....	94
Tabela 18 - Descrição do caso de uso Visualizar Tarefas. ....	94
Tabela 19 - Descrição do caso de uso Visualizar Informação Tarefa. ....	95
Tabela 20 - Descrição do caso de uso Verificar Tarefa. ....	95
Tabela 21 - Descrição do caso de uso Editar Tarefa.....	96
Tabela 22 - Descrição do caso de uso Criar Tarefa .....	98

# 1 Introdução

O presente trabalho de dissertação, desenvolvido no âmbito da unidade curricular de Dissertação de Mestrado em Engenharia Informática na Universidade do Minho, tem como tema a *Representação de Informação Temporal em Sistemas de Apoio à Decisão Clínica*.

Este estudo abrange áreas das ciências da computação, particularmente da Inteligência Artificial (IA) que apresentam uma vertente mais aplicacional, como a eHealth, a Informática Médica, e os Sistemas de Apoio à Decisão Clínica (SADC).

Na *Introdução*, na secção *Enquadramento e Motivação*, pretende-se fornecer um enquadramento teórico do trabalho e da motivação que lhe esteve subjacente. O mote para o trabalho é tratado na secção *Tema e Objetivos* sob a forma de um conjunto de questões de investigação que depois foram transformadas em objetivos a concretizar. A secção *Método de Trabalho e Investigação* descreve a metodologia que permitiu atingir os objetivos propostos. Por fim, a secção *Estrutura* descreve a organização do documento e os tópicos abordados em cada um dos capítulos.

## 1.1 Enquadramento e Motivação

O tema da presente dissertação envolve áreas científicas de relevo, pelo que importa enquadrar o problema que se pretende abordar nas diferentes áreas científicas que este toca.

A área dominante desta proposta de trabalho é *eHealth*. *eHealth* é um termo recente que descreve a prática de cuidados de saúde suportada por processos eletrónicos [1]. A investigação nesta área visa desenvolver processos eletrónicos/digitais em saúde e, como tal, considera-se que se trata de uma subárea da Informática Médica [2] [3].

Outra das áreas de grande interesse prático para este trabalho é IA, uma vez que é uma área de investigação da computação dedicada a encontrar métodos ou dispositivos computacionais, que possuam ou multipliquem a capacidade racional do ser humano de resolver problemas, pensar ou raciocinar, de forma ampla. Também pode ser definida, segundo Rich *et al.* [4], como o ramo das ciências da computação que se ocupa do comportamento inteligente ou ainda, o estudo de como fazer os computadores realizarem tarefas que atualmente são mais bem desempenhadas por seres humanos [5].

Neste âmbito, a IA tem uma palavra a dizer relativamente às técnicas utilizadas nas tecnologias *eHealth* através do melhoramento das suas implementações, o que se pode concretizar através dos sistemas de apoio à decisão.

Os SADCs são sistemas especialistas projetados para ajudar os médicos e outros profissionais de saúde a desempenhar tarefas do processo clínico, como, por exemplo, determinar o diagnóstico com base nos dados do paciente [6]. Este aspeto evidencia-se na definição proposta por Musen *et al.* em [7], segundo a qual os SADCs ligam as observações clínicas aos conhecimentos médicos, de modo a influenciar de forma positiva as escolhas dos profissionais de saúde e contribuir para uma melhoria dos cuidados. Contudo, tal como expõe a definição, é necessário que haja um suporte para a representação do conhecimento médico, que permita o seu cruzamento, de forma automática, com as observações do estado do paciente. Este suporte pode existir nas mais diversas formas, sendo que uma das mais utilizadas é a de algoritmos baseados em protocolos clínicos em formatos que possibilitam a sua interpretação automática.

De seguida, nas secções 1.1.1 e 1.1.2, clarificam-se alguns aspetos relativamente às funções e importância dos SADCs e ao papel que os protocolos clínicos podem desempenhar no seu desenvolvimento.

### **1.1.1 Sistemas de Apoio à Decisão Clínica**

A investigação com vista a criar sistemas de computador artificialmente inteligentes tem sido uma das mais ambiciosas e, não surpreendentemente, controversas atividades no domínio das ciências da computação [8]. Desde muito cedo, que os investigadores na área da saúde também foram atraídos pelo potencial que tal tecnologia pode ter na medicina [9]. Desta forma, o foco inicial da Inteligência Artificial na Medicina tem sido o desenvolvimento de sistemas capazes de realizar o diagnóstico e fazer recomendações de terapias.

Por definição, os SADCs são sistemas de conhecimento ativo que usam itens de dados do paciente para gerar recomendações específicas para cada caso. Estes sistemas geralmente incluem uma base de conhecimento médico, um suporte para os dados dos pacientes e um motor de inferência [7]. Um SADC é qualquer programa de computador projetado para ajudar os profissionais de saúde a tomar decisões clínicas. De certa forma, qualquer sistema de computador que trabalhe com dados clínicos ou conhecimento médico destina-se a fornecer

apoio à decisão [6]. Assim sendo, importa considerar os quatro tipos de funções que estes sistemas podem desempenhar [10]:

- Administrativa: A função administrativa procura apoiar a codificação clínica e a documentação, a autorização de procedimentos e referências;
- Gestão da Complexidade Clínica: A função de gestão da complexidade clínica procura gerir os pacientes em protocolos de investigação e tratamento bem como o acompanhamento de receituário, referências de acompanhamento, e cuidados preventivos;
- Controlo de Custos: O objetivo do controlo de custos é a monitorização das prescrições de medicamentos, evitando duplicações ou exames desnecessários;
- Apoio à Decisão: Esta função consiste em prestar apoio aos processos de diagnóstico e planeamento de tratamentos clínicos, bem como promover a utilização das melhores práticas clínicas.

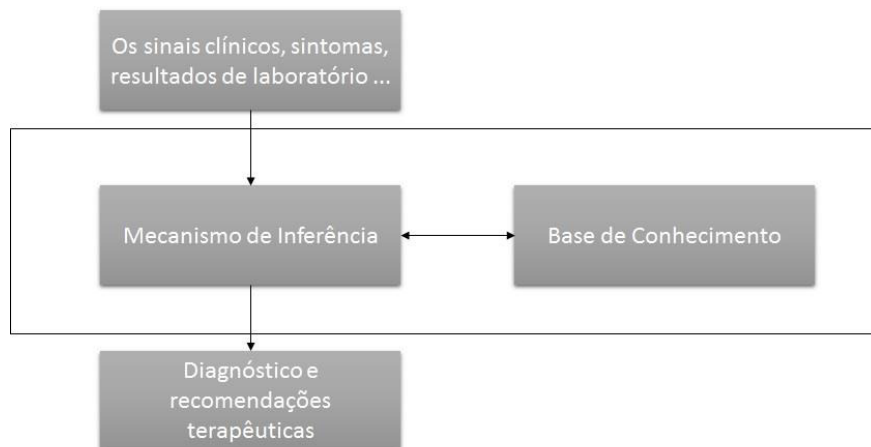


Figura 1 - Modelo Geral dos Sistemas de Apoio à Decisão Clínica.

Após a breve descrição das funções dos SADCs é igualmente importante abordar o modelo geral destes sistemas. Um modelo geral de SADCs pode ser observado na Figura 1.

Tal como ilustrado na Figura 1, as entradas do sistema são compostas por sinais, sintomas e exames laboratoriais entre outros, e as saídas incluem o diagnóstico e recomendações terapêuticas. O sistema apresenta dois componentes básicos: uma base de conhecimento e um mecanismo de inferência. A base de conhecimento é um conjunto estruturado de conhecimento médico especializado. O mecanismo de inferência é um conjunto



de algoritmos computacionais usados para processar sinais clínicos, sintomas e resultados de exames laboratoriais em relação à base de conhecimento.

De acordo com o modelo geral de um SADC descrito na Figura 1, os utilizadores do sistema interagem com o SADC de forma iterativa, introduzindo seletivamente a informação e utilizando as recomendações das saídas dos SADC para auxiliar a formulação de diagnósticos e a tomada de decisão nos processos terapêuticos [11].

De seguida, serão apresentados alguns projetos de investigação que constituem exemplos destes sistemas.

A solução KARDIO é um sistema de aprendizagem automática que foi desenvolvido para interpretar eletrocardiogramas (ECGs), com a finalidade de diagnosticar arritmias cardíacas. Dado um conjunto de casos clínicos que atuam como exemplos, o sistema é capaz de produzir uma descrição sistemática dessas características clínicas que caracterizam determinadas condições de saúde. Este conhecimento pode ser expresso sob a forma de regras simples, ou muitas vezes como uma árvore de decisão [8].

O sistema HELP é um sistema de informação hospitalar integrado desenvolvido no Hospital LDS em Salt Lake City [6]. Este sistema não só suporta as aplicações de rotina de um sistema de informação hospitalar, incluindo a gestão de admissões e altas médicas e de entrada de pedidos, mas também oferece funções de apoio à decisão. O suporte à decisão fornece aos profissionais de saúde alertas e lembretes, interpretação de dados e ambientes de diagnóstico do paciente, sugestões de gestão de pacientes e protocolos clínicos [8].

DXplain é um SADC desenvolvido no Hospital Geral de Massachusetts, no laboratório de ciências da computação, utilizado para auxiliar o diagnóstico. Com base num conjunto de resultados clínicos, incluindo sinais, sintomas e dados laboratoriais, o sistema produz uma lista de classificação de diagnósticos que podem explicar (ou estar associados) às manifestações clínicas, fornecendo uma justificação para cada uma das doenças que possam ser consideradas [8].

Uma abordagem diferente ao suporte à decisão foi incorporada em MYCIN, um sistema de consulta que se foca numa gestão adequada de pacientes que têm infeções em detrimento das tarefas de diagnóstico [6]. O sistema MYCIN caracteriza-se por ter sido projetado para recomendar o tratamento de certas infeções do sangue, baseando-se em regras do tipo *IF-THEN*.

O conhecimento sobre as doenças infecciosas no MYCIN é representado como regras de produção, cada uma contendo um pacote de conhecimento derivado de discussões entre profissionais de saúde [6].

O presente trabalho de dissertação foca-se em SADCs que fornecem apoio à decisão através de recomendações específicas para cada paciente com base em versões de protocolos clínicos para interpretação automática, e de como o conhecimento dos protocolos clínicos pode ser modelado para que um sistema possa fornecer recomendações num contexto temporal.

### **1.1.2 Protocolos Clínicos**

Protocolos clínicos são documentos desenvolvidos de forma sistemática que visam melhorar a qualidade dos cuidados de saúde, reduzir as variações da prática médica e reduzir os custos de cuidados de saúde. Para que sejam eficazes, devem ser integrados no fluxo de cuidados e fornecer conselhos específicos para cada paciente, quando e onde for necessário [12].

Desta forma, a sua formalização em versões para interpretação automática, as chamadas *Computer-Interpretable Guidelines* (CIGs), torna possível o desenvolvimento de sistemas de apoio à decisão baseados em CIGs que oferecem uma melhor possibilidade de afetar o comportamento clínico em relação aos documentos narrativos das correspondentes versões em texto. As CIGs são um dos suportes possíveis para o conhecimento médico em sistemas de apoio à decisão. Existem, inclusive, alguns modelos de representação com esse propósito, dos quais são exemplo *Arden Syntax* [13], *Guideline Interchange Format* (GLIF) [14], *Asbru* [15], *PROforma* [16] e o *Standards-based Active Guideline Environment* (SAGE) [17].

Exceto o *Arden Sintaxe*, que é um modelo que codifica especificamente pequenos fragmentos de conhecimento médico sob a forma de regras, a maior parte dos modelos utiliza redes de tarefas para representar e expor o conhecimento dos protocolos clínicos. O modelo de rede de tarefas parece ser o que melhor se adapta à informação veiculada por protocolos clínicos, permitindo uma separação clara entre o conhecimento processual, ou seja, a ordem relativa das recomendações e o conhecimento médico.

Um modelo é, geralmente, acompanhado de um mecanismo de execução que é responsável por interpretar as restrições clínicas e as restrições temporais colocadas às tarefas. Ferramentas como *Arezzo™* (para *PROforma*), a *Digital Electronic Guideline Library* (DeGeL) (para *Asbru*), o *Guideline Execution Engine* (GLEE) (para GLIF3) e *SAGEDesktop* são utilizados para

fornecer recomendações de forma interativa e para o armazenamento de execuções de protocolos clínicos representados nos seus respectivos modelos [18].

No entanto, a maioria destes modelos carece de construtores de representação (unidades básicas de representação) apropriadas para lidar com padrões temporais. Estes padrões temporais podem assumir três níveis:

- Execução de tarefas: trata-se das restrições temporais colocadas à realização de tarefas e procedimentos clínicos que podem incluir padrões como duração, periodicidade, tempos de inicialização e de finalização;
- Observações clínicas: corresponde à representação das observações clínicas num padrão temporal que indique a sua duração e frequência;
- Condições para diagnóstico e para a recomendação de tarefas: como em qualquer modelo CIG é necessário representar condições que se devem verificar para a aplicação de determinadas tarefas ou para chegar a um determinado diagnóstico já que estas condições incidem diretamente sobre o estado do paciente e, portanto, requerem uma avaliação temporal das manifestações das observações clínicas.

Assim sendo, no âmbito da representação de CIGs, surge este trabalho de dissertação, motivado pela necessidade de melhorar o processo de decisões clínicas. Pretende-se, deste modo, apresentar um sistema de apoio à decisão que procura melhorar a qualidade e, se possível, diminuir as ocorrências de casos de más práticas, como erros médicos [19], através do fornecimento de recomendações num contexto temporal, uma vez que variável temporal é uma componente que não deve ser descartada e uma importante valência na representação de protocolos clínicos.

## 1.2 Tema e Objetivos

Este trabalho de dissertação apresenta como tema a *Representação de Informação Temporal em Sistemas de Apoio à Decisão Clínica*. Tendo como ponto de partida SADCs que utilizam CIGs como suporte para a sua base de conhecimento, procurou-se estudar os aspetos temporais da representação de protocolos clínicos para interpretação automática. Deste modo, a questões de investigação que guiaram a execução do trabalho foram:

- A. Os aspetos temporais da execução de tarefas clínicas são tratados de forma adequada nos modelos CIG atuais?
- B. Quais os aspetos temporais que carecem de desenvolvimento nos modelos CIG e de que forma devem ser desenvolvidos?

As questões de investigação especificadas anteriormente permitiram formular os seguintes objetivos a concretizar:

1. Identificar os principais aspetos para a representação temporal de recomendações com base em protocolos clínicos em sistemas de apoio à decisão;
2. Identificar as principais lacunas em termos de representação temporal nos modelos CIG atuais;
3. Conceber um modelo de representação temporal que preencha as lacunas identificadas e respetiva integração num modelo CIG;
4. Conceber um motor de execução que trate de restrições temporais aplicadas às observações clínicas e às tarefas recomendadas em protocolos clínicos;
5. Conceber uma interface que possibilite a um profissional de saúde visualizar as recomendações clínicas num contexto temporal.

### **1.3 Método de Trabalho e Investigação**

Relativamente ao método de trabalho será adotada a metodologia ação-investigação [20]. Numa fase inicial fez-se um levantamento de informação crucial para o processo de conceção da solução

Em seguida, iniciou-se a pesquisa dos conceitos e projetos relevantes para o trabalho. A assimilação dos conceitos e projetos foi sujeita a uma constante renovação, à medida que surgiam novas ideias e informação. A última parte do trabalho consistiu no desenvolvimento de um modelo e protótipo funcionais que permitissem atingir os objetivos traçados.

A metodologia aplicada engloba as seguintes fases de desenvolvimento:

- Definição do problema e respetivas características;
- Atualização constante do estado da arte e objetivos do trabalho;
- Desenvolvimento de um protótipo que permita alcançar os objetivos definidos;

- Análise e correção do protótipo com base nos resultados obtidos;
- Difusão do conhecimento e resultados obtidos pela comunidade científica.

Ainda inerente às metodologias de trabalho utilizou-se uma metodologia adaptada do *SCRUM* [21] no desenvolvimento das soluções de *software* necessárias à concretização dos objetivos.

## **1.4 Estrutura do Documento**

A presente dissertação foi estruturada em seis capítulos, organizados da seguinte forma:

### **1. Introdução**

No primeiro capítulo realiza-se uma descrição breve da situação atual e o respetivo enquadramento do trabalho, uma introdução aos principais conceitos e uma apresentação da motivação, tema, objetivos e metodologia de investigação. Também é realizada uma breve descrição da estrutura da dissertação.

### **2. Computer-Interpretable Guidelines**

O segundo capítulo trata de CIGs, referindo a sua importância, os benefícios e as desvantagens da sua utilização. Estes aspetos são utilizados posteriormente para apontar as principais características de alguns modelos, como *Arden Syntax*, *Guideline Interchange Format (GLIF)*, *Asbru*, *EON*, *PROforma* e *GLARE*. No final do capítulo realiza-se uma análise aos construtores de representação temporal destes modelos e identifica-se as suas principais lacunas.

### **3. Proposta de um Modelo de Representação Temporal**

Com base nas abordagens discutidas no capítulo dois, propõe-se um modelo de representação temporal a integrar na ontologia *CompGuide* para a representação de protocolos clínicos. Começa-se por apresentar as principais classes e propriedades do modelo e posteriormente aborda-se a representação ao nível computacional com um exemplo de um protocolo clínico.

### **4. Análise do Problema de Execução de Protocolos Clínicos**

Este capítulo foca-se na problemática subjacente à execução de protocolos clínicos atendendo às suas restrições temporais, abordando um conjunto de elementos que descrevem a compreensão do domínio, as funcionalidades do sistema e os atores do sistema. É também fornecida uma descrição dos diferentes casos de uso no sistema de execução de CIGs.

## **5. Desenvolvimento e Implementação do Sistema de Execução de Protocolos Clínicos**

Com base na análise do problema e segundo os artefactos estudados em engenharia de *software*, neste capítulo aborda-se o sistema segundo um ponto de vista de implementação e desenvolvimento de *software*. Assim sendo, torna-se pertinente abordar a arquitetura de *software*, as tecnologias e ferramentas utilizadas, o modelo relacional, diagrama de classes, diagramas de sequência, entre outros.

## **6. Conclusões e Trabalho Futuro**

Por fim, o último capítulo desta dissertação sintetiza o trabalho realizado e as principais conclusões a retirar. São também mencionadas as perspetivas de trabalho futuro e a divulgação de resultados.

## 2 Computer-Interpretable Guidelines

Ao efetuar o estudo da problemática subjacente à representação de informação temporal em SADCs tornou-se necessária a realização de uma análise das soluções disponíveis na literatura e das diversas aplicações que existem atualmente.

A metodologia utilizada para elaborar a pesquisa das temáticas abordadas neste capítulo, consistiu sobretudo na análise de artigos de conferências e revistas disponíveis nas bases de dados do *Google Scholar*, *Science Direct*, *Pubmed*, entre outras. De entre os termos pesquisados há que destacar: *Clinical Decision Support Systems*, *Computer-Interpretable Guidelines*, *Temporal Reasoning*, *Models of Temporal Reasoning*.

O objetivo desta análise é não só identificar os pontos relevantes das várias soluções que pudessem ser incorporados também na solução apresentada, mas também fazer uma reflexão sobre os principais aspetos do domínio deste projeto.

Neste capítulo, na secção *Importância, Vantagens e Desvantagens* aborda-se o impacto das CIG nos sistemas de apoio à decisão. De seguida, na secção *Principais Modelos* fornece-se uma descrição dos principais modelos CIG e dos seus construtores de representação. Na secção *Importância da Representação Temporal no Raciocínio Médico*, discute-se a relevância dos aspetos temporais na prática clínica. Por fim, na secção *Discussão e Análise dos Modelos de Representação Temporal*, avalia-se a expressividade dos modelos CIG selecionados quanto à representação de padrões temporais.

### 2.1 Importância, Vantagens e Desvantagens

As CIGs são representações de protocolos clínicos num formato digital estruturado e interpretável por máquinas. São também representações de protocolos clínicos que podem ser integradas em SADCs e visam facilitar a implementação de boas recomendações na prática clínica diária, uma vez que se apresentam disponíveis no momento do ato clínico [22]. A implementação dos protocolos clínicos em sistemas de apoio à decisão pode levar a uma melhor aceitação e aplicação destas boas práticas médicas, pois estes sistemas são capazes de monitorizar as ações e observações dos profissionais de saúde [23][24].

Os protocolos clínicos desempenham um papel importante no suporte à qualidade, à auditoria e à investigação, através de um conjunto de especificações e requisitos utilizados na avaliação dos cuidados de saúde [25]. Isto permite afirmar que os protocolos clínicos possibilitam a medição dos cuidados de saúde de forma a avaliar o seu grau de adequação. Assim sendo, as CIGs permitem aumentar a precisão, a correção sintática e a correção semântica dos protocolos clínicos através de testes automáticos de verificação com o objetivo de assegurar a sua integridade. Estes testes garantem que não existem erros nos protocolos, socorrendo-se para o efeito a ambientes de simulação, nos quais o protocolo é aplicado a diferentes casos clínicos, fornecidos por registos existentes de pacientes.

Tal como já foi referido as CIGs surgiram com o intuito de serem formatos digitais de protocolos clínicos. A utilização de tal suporte traz vantagens uma vez que os computadores estão prontamente disponíveis e são utilizados de forma frequente nas consultas. Versões CIG de protocolos clínicos oferecem vantagens como a fácil atualização, baixos custos de produção, a possibilidade de incluir outras bases de dados e material audiovisual, a possibilidade de integração em sistemas de apoio à decisão, bem como a capacidade de monitorizar o uso de protocolos [26]. A reutilização do conhecimento também é uma mais-valia das CIGs. Se o modelo utilizado organizar uma CIG em módulos, cada um contendo fragmentos de conhecimento, torna-se fácil incluir estes fragmentos de conhecimento noutros protocolos clínicos e até referenciar os protocolos clínicos mais específicos no contexto de protocolos clínicos com maior abrangência;

O desenvolvimento de SADCs com base em CIGs tem sido proposto como estratégia de promoção da implementação de protocolos clínicos, contudo existem alguns obstáculos à implementação destes sistemas.

A conversão de protocolos clínicos em algoritmos informáticos a partir das suas versões em texto não é uma tarefa fácil, pois estas versões não foram originalmente concebidas para serem interpretáveis por computadores e, em alguns casos, contêm instruções complexas, que manipulam demasiadas variáveis e que dificilmente são traduzíveis em algoritmos eficientes [27].

A desvantagem das CIGs surge devido à natureza condensada dos algoritmos, sendo muitas vezes rígidos, não fornecendo todas as informações presentes nas suas versões textuais e que podem ser necessárias aos médicos não especialistas. Além disso, não há espaço para a



explicação e justificação de recomendações que vão contra a intuição dos profissionais de saúde [26].

Por vezes, o vocabulário utilizado nos documentos possui palavras evasivas para quantificar medidas, em vez de limites numéricos, e os critérios nos pontos de decisão nem sempre são explícitos e indicam o que fazer. A falta de precisão dos conceitos dá lugar a ambiguidade e vazios de conhecimento com os quais os computadores não conseguem lidar [27]. Quanto maior a simplicidade e assertividade de um protocolo, mais fácil será a sua adaptação ao formato de CIG.

A integração de sistemas baseados em CIGs com Registos de Saúde Eletrónicos (RSEs) também é uma tarefa complexa, pois cada RSE possui a sua própria forma de organização da informação que raramente coincide com a forma como a CIG organiza o seu conhecimento [28]. O RSE pode ter uma orientação:

- Cronológica: os dados clínicos do paciente e as observações são registados de forma cronológica;
- Por origem: os dados clínicos são armazenados consoante a sua origem, ou seja, a proveniência da informação determina a sua catalogação e conseqüente registo;
- Orientação por problema: organização da informação por problema de saúde, sendo necessário um mapeamento dos conceitos do registo clínico para os conceitos da CIG.

Uma possível solução para os problemas acima referidos é o desenvolvimento de um modelo *standard* de representação de CIGs. Um modelo formal para representação de protocolos que forneça um conhecimento aprofundado dos processos clínicos que constam em CIGs, devendo, portanto, ser um *standard* em três aspetos essenciais da representação de protocolos: representação de informação administrativa (autoria, instituição, data de criação, etc), representação e mecanismos de controlo de fluxos de tarefas clínicas, representação de condições relativas ao estado do paciente, e representação de informação temporal.

## 2.2 Principais Abordagens à Modelação

Pode considerar-se que a primeira abordagem à modelação de CIGs, embora rudimentar, foi a implementada no sistema HELP em 1967 [29]. A componente de CIGs do sistema consistia

num conjunto de módulos, denominados *help sectors*, que continham o conhecimento clínico na forma de regras lógicas. Deste então, o desenvolvimento de abordagens de CIGs proliferou.

De seguida, serão apresentadas diferentes abordagens à modelação de CIGs, dando especial atenção ao modelo de representação de cada uma. A seleção das abordagens seguiu dois critérios: consenso dos autores sobre quais as mais relevantes e a disponibilidade de informação na literatura sobre as abordagens. As abordagens selecionadas foram: *Arden Syntax* [13], GLIF [14], Asbru [15], EON [30], PROforma [16] e *Guideline Acquisition, Representation and Execution* (GLARE) [31].

### 2.2.1 Arden Syntax

Um dos formalismos mais conhecidos na representação de CIGs e suporte à decisão é o *Arden Syntax*, baseado no sistema HELP. Este formalismo foi desenvolvido em 1989 e o seu desenvolvimento resulta de um consenso que reuniu instituições de saúde e académicos, com o objetivo de criar um modelo que permitisse a partilha de protocolos entre diferentes instituições, com diferentes sistemas de informação [32].

*Arden Syntax* foi reconhecido como um modelo padrão em 1992 pela *American Society for Testing and Materials* (ASTM) [33]. Atualmente, a versão de *Arden Syntax* em desenvolvimento é 3.0 e é distribuída pelo grupo *Health Level Seven* (HL7) [34]. Esta abordagem foca-se na partilha de protocolos simples e independentes sob a forma de módulos. Não é um formato apropriado para protocolos complexos, sendo utilizado para representar regras simples.

Em *Arden Syntax*, cada protocolo é modelado como um Medical Logic Module (MLM) que codifica conhecimento para uma única decisão. Cada MLM é um ficheiro ASCII que contém componentes agrupados em três categorias: *maintenance*, *library* e *knowledge* [35]. O conhecimento médico é armazenado na categoria *knowledge*, através dos seus componentes: *type*, *data*, *evoke*, *logic* e *action* [35]. A representação de informação temporal é feita através do compartimento *logic*, uma vez que possui operadores temporais (e.g., *before*, *after*, *ago*). Esta representação temporal é assegurada, sobretudo, através da invocação de um módulo do modelo Asbru, que se irá abordar com mais detalhe na secção 2.4.

## 2.2.2 GLIF

GLIF [36] representa um esforço da organização *Intermed Collaboratory* (uma colaboração das Universidades de Harvard, Stanford e Columbia) no desenvolvimento de uma representação partilhável de CIGs, cuja primeira publicação data de 1998. Recebeu influências de outras abordagens existentes, como *Arden Syntax* [32], GEODE CM [37], MBTA [38] e EON [30].

Esta abordagem foi desenvolvida de forma a refletir um fluxograma de passos estruturados e agendados, que representassem decisões e ações clínicas. As primeiras versões de GLIF careciam de uma especificação formal dos passos de um protocolo e de um mapeamento dos dados para o registo de saúde eletrónico. O principal objetivo deste modelo é a partilha de protocolos clínicos e, por isso, estes são modelados de forma a serem perceptíveis, tanto por especialistas do domínio clínico como por *parsers* automáticos utilizados em diferentes sistemas de apoio à decisão.

Na sua última versão, GLIF3 [14], define cinco tipos de passos: *decision steps*, *patient state steps*, *branch steps*, *synchronization steps* e *action steps*. Cada protocolo em GLIF consiste num conjunto de pontos que representam cada um dos cinco tipos de passos, ligados entre si num fluxograma. *Decision steps* modelam pontos de decisão num protocolo, direcionando o fluxo de trabalho de um passo para passos alternativos. Dentro de uma *decision* existem duas subclasses: *case step* e *choice step*. Em *case step* existe um conjunto de expressões lógicas que direcionam o fluxo de trabalho para um dos passos alternativos. As expressões lógicas correspondem a um excerto de *Arden Syntax*, que assume a designação *Guideline Expression Language* (GEL) [39]. Um *choice step* aplica-se em situações em que o protocolo sugere múltiplas alternativas, mas deixa a escolha a cargo de um agente externo, no caso o utilizador. *Patient state steps* funcionam como etiquetas que contêm atributos utilizados para descrever o estado do paciente. Este tipo de passo pode ser utilizado como ponto de entrada de dados no sistema. Os *branch steps* modelam um conjunto de passos concorrentes, direcionando o fluxo de tarefas para passos paralelos, depois sincronizados em *synchronization steps*. Por fim, os *action steps* modelam tarefas que devem ser realizadas.

## 2.2.3 Asbru

O formalismo Asbru [40] permite expressar objetivos, que no contexto deste modelo, adquirem particular relevância e são designados por intenções. Foi desenvolvido pela Universidade de

Stanford e pela Universidade de Tecnologia de Viena e é particularmente avançado na modelação dos aspetos temporais de protocolos clínicos.

Asbru faz parte do projeto Asgaard [41] e é um formalismo para representação de *protocolos clínicos* como planos, especializando-se na representação de padrões e anotações temporais, desenvolvendo um método de visualização de *protocolos clínicos* num eixo temporal.

Um dos aspetos chave de Asbru é a representação dos objetivos de um plano em *intentions*. A definição de *intentions* auxilia a seleção do plano mais apropriado e é crucial no apoio à decisão. Um exemplo destas situações é o caso em que um profissional de saúde pretende tratar a hipertensão, para a qual um tratamento possível é a administração de betabloqueadores adrenérgicos. Contudo, o médico pode desejar evitar a sua utilização e seguir por outro plano. Com base nas *intentions*, caso o objetivo de baixar a hipertensão não seja atingido, o sistema é capaz de fazer uma apreciação crítica do procedimento do profissional de saúde, aconselhando-o. Por outro lado, se o resultado do tratamento estiver de acordo com as *intentions*, não será gerada a apreciação crítica e o sistema aceitará o plano seguido como correto. As *intentions* são definidas como padrões temporais de ações dos profissionais de saúde e estados dos pacientes que devem ser mantidos, alcançados ou evitados.

Na representação de protocolos em Asbru, as anotações temporais são de extrema importância [42], uma vez que especificam quatro pontos no tempo relativamente a um ponto de referência, o qual pode ser um ponto específico ou abstrato no tempo ou uma transição de estados de um plano. Estes quatro pontos são: *earliest starting shift* (ESS), *latest starting shift* (LSS), *earliest finishing shift* (EFS) e *latest finishing shift* (EFS). Podem ser especificadas duas durações: *minimum duration* (MinDu) e *maximum duration* (MaxDu). Estas referências representam as restrições temporais para a execução dos planos ou verificação de condições.

Assim sendo, ao contrário de outras abordagens como GLIF e PROforma, a visualização de protocolos clínicos em Asbru não acontece na forma de um fluxograma, pois a visualização do tempo e intenções através de um fluxograma é difícil e complexa. Para o efeito, foi desenvolvida uma ferramenta denominada AsbruView [11] que utiliza gráficos para a visualização de *protocolos clínicos*. Em AsbruView, os planos são visualizados como pistas e os vários tipos de condições são vistos como sinais de tráfego, segundo uma vista topológica. Em AsbruView os planos também podem ser visualizados na vista temporal, que se foca na sua

dimensão temporal e respetivas condições. Esta vista faz uso das referências temporais acima mencionadas.

## 2.2.4 EON

O formalismo EON foi um dos modelos de CIGs que destacou a importância de o utilizador compreender o fluxo de tarefas apresentadas. O EON [30] foi desenvolvido na Universidade de Stanford e é o precursor de formalismos como GLIF [39] e SAGE [43], continuando a ser desenvolvido como um sistema de investigação. Tal como GLIF representa os protocolos na forma de fluxogramas.

O modelo EON consiste em vários componentes que facilitam a aquisição e execução de protocolos clínicos e é orientado aos objetos [43]. Dharma consiste em classes que descrevem entidades de um protocolo clínico na forma de passos temporalmente estruturados. O modelo Dharma não é estático, o que significa que o modelo pode ser expandido com classes adicionais que captem novos aspetos dos protocolos clínicos.

Neste modelo os protocolos clínicos gerem o comportamento do paciente, consistindo em decisões e ações que podem levar a mudanças no estado do paciente ao longo do tempo. Neste modelo, as decisões são tomadas durante os encontros dos profissionais de saúde com os pacientes. Ações, como escrever receitas ou a solicitação de um exame de laboratório, são executadas durante estes encontros. Para a representação de restrições temporais, EON utiliza um excerto de Asbru.

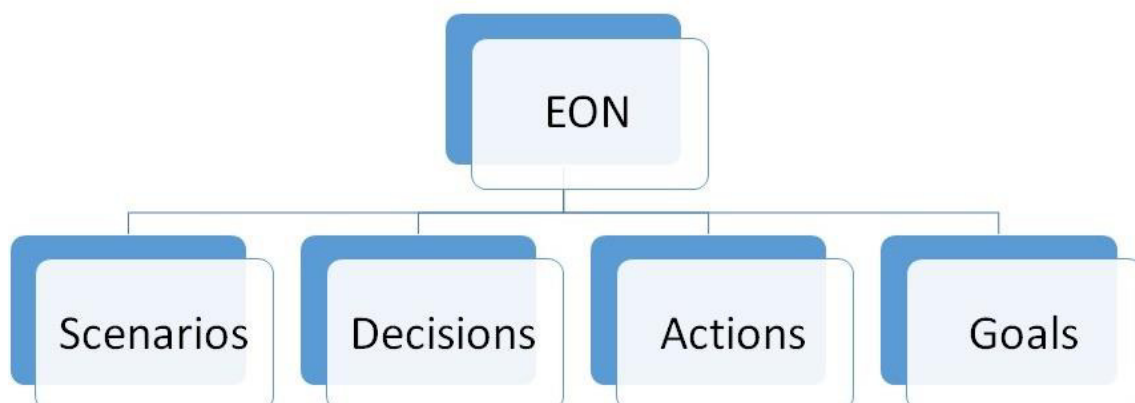


Figura 2 - Representação esquemática das principais classes do modelo EON.

As classes primitivas em EON encontram-se representadas na Figura 2 e são: *scenarios*, *decisions*, *actions* e *goals*. Estas classes formam a ontologia do modelo.

As *actions* são definidas como atos instantâneos que conduzem a alterações no estado geral, como por exemplo, recolher dados do paciente, mostrar uma mensagem ao utilizador ou começar um novo tratamento. Em EON também existe o conceito de *activity*, porém, ao passo que *actions* são instantâneas, as *activities* modelam processos que decorrem ao longo do tempo. As *activities* possuem estados que se podem alterar ao longo do tempo, como resultado de *actions*.

EON também possui *actions* que referenciam um conjunto de outras ações ou um sub-protocolo clínico. Exemplos deste tipo de *actions* são aquelas que modelam passos de ramificação e sincronização.

### 2.2.5 PROforma

No Reino Unido, o Advanced Computation Laboratory of Cancer Research iniciou em 1998 o desenvolvimento do formalismo PROforma [44][45]. O objetivo deste formalismo é o desenvolvimento de sistemas especialistas mais confiáveis que possam auxiliar nos cuidados de saúde a um paciente de forma ativa, através de suporte à decisão e gestão do fluxo de tarefas.

PROforma resulta da concatenação das palavras *proxy* (que significa permissão para atuar como outrem) e *formalize* (que significa dar forma definitiva a).

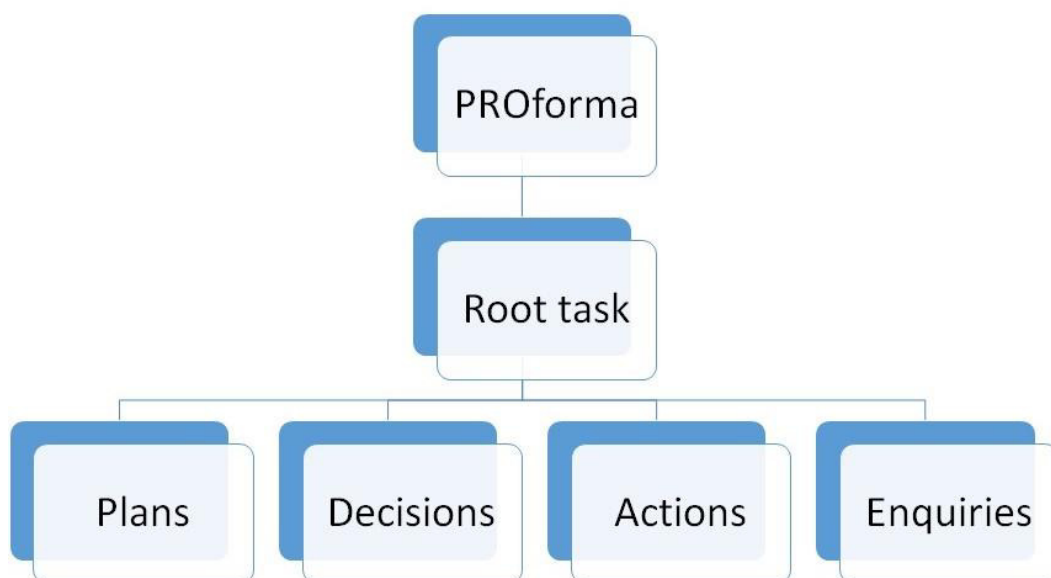


Figura 3 - Representação esquemática das principais classes de tarefas do modelo PROforma.

Tal como GLIF, PROforma também representa protocolos como um fluxograma, nos quais os elementos são instâncias de classes pré-definidas. PROforma define 4 classes de tarefas, representadas na Figura 3, sendo que a classe *plans* permite definir sub-protocolos clínicos e determina uma sequência ordenada de tarefas, restrições lógicas, restrições temporais à sua execução e as circunstâncias em que um plano deve terminar ou abortar, isto está explícito no tipo de atributos que a classe *plans* acrescenta aos já referidos: *components*, *scheduling constraints*, *temporal constraints*, *abort conditions* e *termination conditions*.

O atributo *components* alberga um conjunto de referências a tarefas que constituem um plano (e.g., *history*, *diagnosis*, *therapy*, *follow-up*), cuja ordem é definida em *scheduling* e *temporal constraints*. É possível definir que uma tarefa ocorre especificamente após outra, ou que uma tarefa ocorre após um dado período de tempo. Esta restrição temporal à realização de tarefas é o que demarca PROforma de outras abordagens como GLIF.

Os protocolos clínicos em PROforma são armazenados, utilizando *Red Representation Language (R<sup>2</sup>L)*, uma linguagem de representação orientada ao tempo. Para realizar a execução dos protocolos clínicos é necessário convertê-los para outra linguagem, denominada *Logic of R<sup>2</sup>L (L<sub>R<sup>2</sup>L</sub>)*. *L<sub>R<sup>2</sup>L</sub>* é uma linguagem de representação do conhecimento baseada em predicados.

## 2.2.6 GLARE

The *Guideline Acquisition, Representation and Execution (GLARE)* [31] é um projeto que inclui um modelo de representação de protocolos clínicos e um sistema para adquiri-los e executá-los. Este projeto foi desenvolvido pelo Departamento de Ciências da Computação da Universidade de Piemonte Orientale, Alessandria, Itália.

O modelo de representação não usa uma representação padrão. Em vez disso, define uma estrutura proprietária baseada em grafos que exhibe protocolos clínicos, onde uma ação clínica é representada por um nó. As principais entidades do modelo GLARE encontram-se representadas na Figura 3.

É possível definir ações atômicas que representam tarefas simples, como *queries* para obter informações externas, *work actions* que representam procedimentos médicos, *decision actions* como um conjunto de condições para selecionar alternativas e conclusões que descrevem a tomada de decisão.

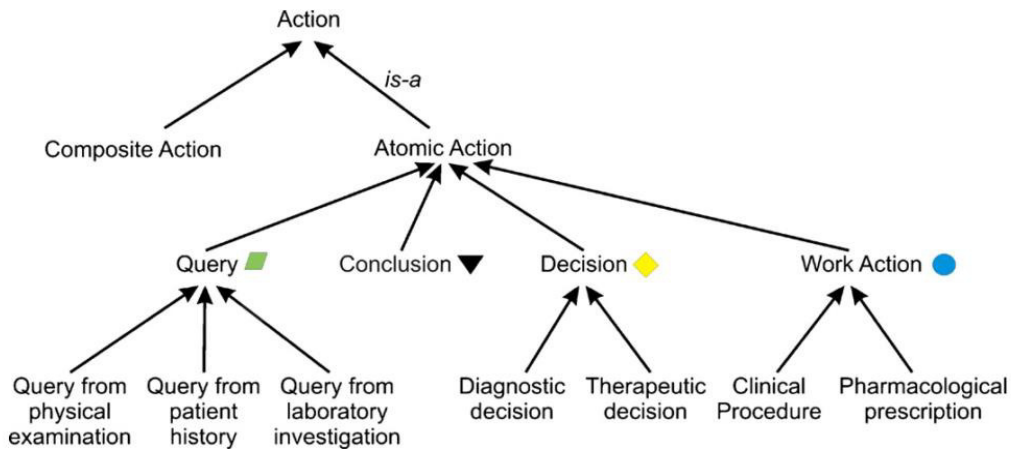


Figura 4 - Entidades básicas definidas no modelo de representação de GLARE (figura retirada de [46]).

*Decision actions* são tipos específicos de ações que contêm os critérios utilizados para selecionar caminhos alternativos de um protocolo clínico. Estes critérios são representados como conjuntos triplos na forma (*diagnóstico, parâmetro, resultado*) e, por sua vez, um parâmetro é outro conjunto triplo (*dados, atributo, valor*).

Também é possível, em GLARE, definir ações compostas, que são conjuntos de ações atômicas ou outras ações compostas. GLARE foi projetado para lidar com diferentes tipos de restrições temporais e implementa algoritmos de raciocínio temporal especializados.

No mecanismo de execução de GLARE [31], distingue-se entre a fase de aquisição e a fase de execução dos protocolos clínicos. GLARE define três níveis de arquitetura representados na Figura 5, designadamente *System*, *eXtensible Markup Language (XML)* e *DBMS*.

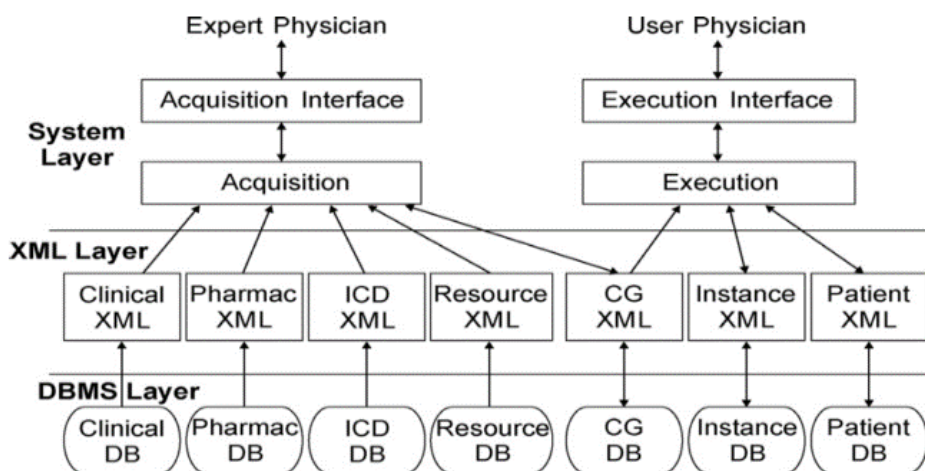


Figura 5 - Arquitetura Geral de GLARE (figura retirada de [46]).



O nível *System* engloba os módulos de aquisição e de execução. O nível *XML* é responsável pelas trocas de dados entre o nível *System* e o nível de DBMS. DBMS é o nível inferior, responsável por estabelecer uma conexão física entre os níveis superiores e as bases de dados onde a informação para a criação e execução de protocolos clínicos é armazenada. Esta informação inclui instâncias abertas de protocolos clínicos, um repositório de protocolos e registos médicos dos pacientes. GLARE usa ICD-9 como terminologia padrão.

## **2.3 Importância da Representação Temporal no Raciocínio Médico**

A representação temporal e o raciocínio em medicina têm sido explorados como temas de investigação desde o final da década de 1980 [47].

A maioria das tarefas clínicas exige a medição e registo de numerosos dados dos pacientes, muitas vezes em suporte eletrónico. Os profissionais de saúde que têm de fazer diagnósticos ou tomar decisões terapêuticas baseadas nesses dados podem ter dificuldades ao serem sobrecarregados pelo seu volume. A maioria dos dados armazenados inclui uma marca temporal que corresponde à data em que a observação é válida. Um padrão emergente ao longo de um do tempo tem muito mais significado do que um resultado isolado ou mesmo um conjunto de resultados. Os profissionais de saúde mais experientes são capazes de combinar várias descobertas simultâneas, abstrair tais descobertas em conceitos clinicamente pertinentes de uma forma sensível ao contexto e detetar as tendências de evolução do estado do paciente.

Assim, é desejável proporcionar informações sucintas, resumos sensíveis num contexto temporal e orientados a dados clínicos armazenados em suporte eletrónico, que sejam capazes de responder a perguntas sobre conceitos e relações presentes nos dados. Fornecer este tipo de suporte beneficiaria tanto um médico como uma ferramenta de apoio à decisão automatizada, que recomenda terapias e diagnósticos baseados na história clínica do paciente até ao presente [48].

Lidar com o tempo para gerir informações médicas pode envolver conhecimento proveniente de diferentes áreas como como a lógica modal, as redes de restrição, modelação de dados e consultas, sistemas dedutivos, sistemas de tempo real, lógica *fuzzy*, *data mining* entre outras [47]. Ainda assim não se pode colocar de parte um outro componente de enorme

importância na área médica e que diz respeito às diferentes tarefas médicas que devem ser tidas em consideração, como por exemplo, o suporte à decisão para o diagnóstico, conselhos sobre terapia, sumarização de dados clínicos, monitorização e estudos epidemiológicos. Por conseguinte, o raciocínio temporal em IA aborda o problema de inferir o estado de um sistema em vários momentos no tempo à medida que as variáveis mudam como resposta aos eventos.

No caso de o raciocínio temporal na medicina, os estados de interesse são os estados dos pacientes e os acontecimentos que provocam mudanças de estado são associados a processos de desordem, intervenções terapêuticas, fatores ambientais/contextuais, ou até mesmo eventos espontâneos/aleatórios. O apoio à decisão com base no período em que são aplicados os cuidados de saúde [10,43] aborda as tarefas de diagnóstico, tratamento, prognóstico e acompanhamento dos pacientes.

Os vários processos que interagem e que estão relacionados com a doença, bem como a terapia a um paciente traduzem a complexidade do raciocínio médico. As diferentes interações numa situação específica, e os eventos causadores dessa situação, devem ser determinados de forma dinâmica. Compreensão e raciocínio conjugados explicitamente com o tempo fornecem uma representação mais precisa da realidade e podem ajudar a filtrar cenários impossíveis [47].

Devido à problemática da imprecisão e à omissão de detalhes temporais em contexto clínico, vários investigadores da IA têm tradicionalmente utilizado a álgebra de intervalo de Allen [49]. No entanto, a incerteza em contexto clínico, que acontece quando a informação temporal precisa está em falta, não é possível ser expressa pela álgebra de Allen, isto é, a álgebra de Allen não permite expressar a incerteza sobre a ocorrência de eventos ou seus relacionamentos qualitativo-temporais. No entanto, a incerteza é uma característica de muitos problemas onde a informação temporal precisa está em falta. Para resolver este problema, é necessário estender a álgebra de intervalo de Allen para se incorporar a incerteza do raciocínio [49]. Para solucionar esta situação uma metodologia poderá ser admitir que um determinado evento é mais provável de acontecer em um determinado momento, mesmo que a informação temporal seja imprecisa. Desta forma, isto leva a descrições de processos temporais - como representado com a lógica de Allen - estendida com informação probabilística [49]. Neste sentido algumas *frameworks* foram propostas, maioritariamente implementadas em programação em lógica (por exemplo Prolog).

Outros projetos abordam esta problemática, admitindo que existem duas formas de solucionar o problema da informação temporal imprecisa e/ou incoerente. Os modelos

temporais lineares e ramificados são uma mais-valia na solução do problema e a escolha de qual a metodologia a adotar deve ser condicionada pelo tipo de programas, políticas e propriedades de execução que se deseja estudar. O tempo linear é o modelo correto a ser usado quando se pretende caracterizar o conjunto de todas as sequências de execução que um programa gera e para estudar as propriedades que possuem uniformemente para todas as sequências de execução de um programa [50]. A abordagem de ramificação do tempo, por outro lado, considera para um determinado programa o conjunto de todas as árvores de execução geradas pelo programa [50].

Neste trabalho de dissertação a abordagem escolhida foi a de tempo linear, ou seja, a de considerar um processo clínico gerado pela aplicação de um protocolo clínico como uma sucessão de eventos ao longo de um eixo temporal, representando-se apenas os eventos selecionados para aplicação. Dada a complexidade, em termos de caminhos possíveis, que os protocolos clínicos podem apresentar, a abordagem de ramificação do tempo seria computacionalmente dispendiosa e pouco prática.

## 2.4 Discussão e Análise dos Modelos de Representação Temporal

A dimensão temporal é um fator importante quando se realiza qualquer procedimento clínico. É importante, então, a capacidade de representar o tempo ao executar os protocolos clínicos resultantes da prática clínica.

Após o estudo dos diferentes modelos CIG na secção 2.2, importa fazer uma análise dos projetos mencionados, bem como analisar a metodologia de representação temporal das recomendações clínicas. Ainda é de relativa importância comparar estes modelos de raciocínio temporal e encontrar limitações.

Na maioria das recomendações que constam de protocolos clínicos, os procedimentos devem ser realizados de acordo com um conjunto de restrições temporais [51], tais como:

- **Durações:** restrições que representam durante quanto tempo uma tarefa deve ser realizada;
- **Periodicidades:** restrições que especificam de quanto em quanto tempo uma tarefa deve ser realizada;

- **Repetições:** restrições que definem o número de vezes que uma tarefa se deve repetir;
- **Tempos de espera:** atrasos na realização de tarefas;
- **Condições de repetição:** condições relativas ao estado do paciente e que ditam se uma tarefa se deve repetir ou não.

Para além das restrições que recaem especificamente sobre as tarefas, existem as restrições que incidem sobre condições do estado do paciente. Estas restrições são utilizadas para expressar o horizonte temporal ao longo do qual o paciente manifestou, ou manifestará sinais e sintomas.

A gestão do tempo é uma das principais preocupações na modelação de CIGs, uma vez que os processos clínicos são cadeias de eventos que se desenrolam ao longo do tempo. Algumas abordagens CIG foram especificamente concebidas para lidar com restrições temporais.

As restrições temporais sobre tarefas e sobre condições do estado do paciente constituem os principais aspetos da representação temporal de protocolos clínicos e a sua identificação constitui o primeiro objetivo deste trabalho de dissertação. Os modelos CIG abordados anteriormente serão agora discutidos tendo em conta estes aspetos.

Todas as abordagens possuem uma forma de representação temporal, contudo determinadas abordagens não possuem construtores proprietários para esta representação, recorrendo a excertos de outras linguagens para este efeito. A estrutura mais complexa, e porventura mais completa, é a de Asbru. GLIF [52] e EON [30] adotam um excerto de Asbru para representação temporal. De forma a ser compatível com Arden Syntax, GLIF [52] define um conjunto de operadores comuns a Arden Syntax, tais como *before*, *after* e *ago*, que apresentam paralelismos com os presentes em PROforma, permitindo definir a ordem relativa de tarefas. PROforma define uma linguagem de expressão de restrições à execução de planos que permite definir a sua duração e condições de repetição.

GLIF3 [52] aborda tanto as restrições temporais colocadas em condições de estado do paciente, bem como as durações de ações. Asbru [41] também fornece um modelo de representação abrangente para durações. Na verdade, este modelo de CIG apresenta os protocolos clínicos como planos estruturados com uma orientação temporal para os quais é

possível definir anotações de tempo, que podem ser restrições sobre a hora de início e hora de término de tarefas (com o início mais breve possível e a finalização mais breve possível), durações máximas e mínimas e momentos cíclicos (por exemplo, todas as manhãs, a cada dia, etc.). Este modelo faz um tratamento explícito e extenso de restrições temporais nas durações e atrasos entre as ações[53], fazendo também uma distinção entre as ações paralelas, ações sequenciais e ações cíclicas.

Conforme definido em Asbru, as restrições temporais sobre a execução de tarefas sequenciais e em paralelo podem ser representadas em duas dimensões, ou seja, ordenando restrições, que podem assumir os valores paralelos como *any order* ou *total order*, e *continuation condition* que pode assumir os valores *all completed* ou *some completed*.

As combinações dessas duas dimensões resultam em cinco restrições temporais representadas pelos construtores: *DO-ALL-TOGETHER*, *DO-SOME-TOGETHER*, *DO-ALL-ANY-ORDER*, *DO-SOME-ANY-ORDER*, e *DO-ALL-SEQUENTIALLY* [54].

Como já foi referido, Asbru também fornece uma terceira categoria de restrições temporais, utilizada para definir periodicidades. Todas estas restrições são representadas dentro do plano principal do protocolo. A mesma abordagem é utilizada por EON e depois adotada por GLIF através um excerto de Asbru para representação temporal.

Em GLARE registou-se uma evolução em termos de representação temporal [55]. Neste modelo introduziram-se construtores mais completos para a representação de periodicidades e esquemas de repetição. Este formalismo foi posteriormente expandido por Anselma *et al.* em [51]. A nova versão fornece um formalismo melhorado para expressar periodicidades, com a possibilidade de definir os atrasos entre os ciclos do evento periódico, tornando-se também possível definir padrões de periodicidade mais complexos. Por exemplo, cada ciclo de um evento periódico pode ter uma periodicidade associada.

Em PROforma [44], os protocolos clínicos são modelados como planos e cada plano pode definir restrições sobre a realização de tarefas, bem como a duração da tarefa e os atrasos entre as tarefas. Além disso, as construções temporais também podem ser utilizadas a fim de especificar as pré-condições sobre ações.

EON [30] usa expressões temporais para permitir o planeamento de etapas sobre os protocolos clínicos e lida com restrições de duração sobre as atividades. Por outro lado, através

da incorporação do sistema *RESUME*, proporciona-se uma abordagem poderosa para lidar com a abstração temporal. Em EON e Arden Syntax existe a representação de atrasos entre o evento que ativa um MLM, e entre MLMs [56].

Outro desenvolvimento interessante é o mapeamento dos padrões temporais a um nível elevado de abstração, realizado por Anselma *et al.* em [51], para um *Simple Temporal Problem* (STP) [57], uma *framework* de raciocínio em que é fornecida uma representação na forma de grafo do protocolo clínico, resultante do cálculo das limitações de tempo relativas a cada tarefa. Apesar de ser uma das técnicas de raciocínio temporal em IA mais utilizadas, o STP não permite representar padrões temporais complexos, uma vez que a estrutura *STP-tree* produzida não é adequada para eventos que se repetem ao longo do tempo [57]. Este aspeto reflete a visão dominante de que as abordagens IA puras [9], decorrentes principalmente da lógica, deparam-se com obstáculos quando são aplicadas ao domínio médico.

De seguida, apresenta-se um quadro de síntese relativamente aos modelos de representação temporal de protocolos clínicos. A Tabela 1 representa a comparação dos modelos CIG tendo como base as restrições temporais na execução de tarefas e as restrições temporais sobre condições o estado do paciente. De modo geral, todas as abordagens apresentam um número razoável de construtores temporais, associados às restrições temporais na execução de tarefas.

Relativamente ao padrão temporal de Duração, todas as abordagens apresentam este construtor, porém Asbru tem uma visão mais completa, na medida em que permite definir este construtor, através de um conjunto de intervalos.

No que toca à representação do padrão temporal repetição de tarefas, EON, GLIF3 e Arden Syntax não representam este construtor. As restantes abordagens definem este tipo de construtor nas restrições temporais dos protocolos clínicos. EON, PROforma e Luca Anselma representam condições de repetição, sendo que as outras abordagens não definem este construtor.

Acerca da representação de eventos periódicos, Abru, PROforma, GLARE e a abordagem de Anselma *et al.* [51] representam este padrão temporal dos protocolos clínicos, sendo que em GLARE, é dado maior ênfase, uma vez que é possível definir este construtor temporal através de um conjunto detalhado de intervalos.

Relativamente à representação das restrições temporais sobre o estado do paciente, apenas GLIF3 se foca neste ponto, pelo que as restantes abordagens apresentam uma lacuna neste aspeto.

Em jeito de conclusão, é possível admitir, através da análise da Tabela 1, que existe a falta de uma abordagem que engloba todos estes aspetos de restrições temporais, o que comprova a necessidade de criar um modelo abrangente, capaz de acomodar os padrões temporais anteriormente identificados. A identificação de lacunas na representação temporal dos modelos CIG atuais é um dos objetivos deste trabalho de dissertação e fica, deste modo, concretizado.

Tabela 1 - Restrições temporais apresentadas nos modelos de representação de protocolos clínicos.

Modelo	Restrições Temporais na Execução de Tarefas					Restrições Temporais sobre o Estado do Paciente
	Durações	Repetições	Periodicidades	Tempos de Espera	Condições de Repetição	
<b>Arden Syntax</b>	✓	✗	✗	✓	✗	✗
<b>GLIF3</b>	✓	✗	✗	✗	✗	✓
<b>Asbru</b>	✓	✓	✓	✓	✗	✗
<b>EON</b>	✓	✗	✗	✓	✓	✗
<b>PROforma</b>	✓	✓	✗	✓	✓	✗
<b>GLARE</b>	✓	✓	✓	✓	✗	✗
<b>Anselma et. al. , 2006 [51]</b>	✓	✓	✓	✓	✓	✗

✗ - Não apresenta a característica ✓ - Apresenta a característica

## 3 Proposta de um Modelo de Representação Temporal

Este capítulo destina-se a apresentar o modelo de representação temporal de CIGs desenvolvido. Uma vez que se pretende que este seja parte integrante do modelo CompGuide, na secção *Ontologia CompGuide* procede-se a uma breve descrição dos elementos e princípios de *design* que lhe estão subjacentes e que foram, posteriormente, adotados no desenvolvimento do modelo temporal. De seguida, na secção *Modelo de Representação Temporal*, apresenta-se as suas principais classes e propriedades. Por fim, na secção *Discussão e Análise do Modelo Proposto*, fornece-se detalhes sobre a representação de um protocolo clínico com uma avaliação do mapeamento dos padrões temporais encontrados.

### 3.1 Ontologia CompGuide

A ontologia CompGuide [19] fornece uma representação de protocolos clínicos na forma de uma rede de tarefas, em *Ontology Web Language* (OWL). De forma a cumprir este propósito, é seguida uma lógica em que os elementos de informação complexos são representados como instâncias de classes com várias propriedades e as informações simples são representadas com propriedades de dados. No entanto, a informação simples que é reutilizável e que, provavelmente, será necessária em múltiplas partes de protocolo clínico, é representada na forma de instâncias de classes específicas. Toda a representação é semelhante a uma lista ligada de procedimentos.

Como tal, um protocolo clínico é representado como uma instância da classe *ClinicalPracticeGuideline*. Os indivíduos desta classe têm um conjunto de propriedades de dados e objetos que permitem a representação descritiva e administrativa de informações dos protocolos clínicos, tais como: o nome do protocolo, a sua descrição geral, a data de criação e última atualização, versão, especialidade clínica, categoria, utilizadores alvo e população alvo. A definição de um protocolo clínico é apresentada na Figura 6 e como exemplo foi utilizado o protocolo clínico da *National Comprehensive Cancer Network* (NCCN) para tratamento do cancro do cólon [58].



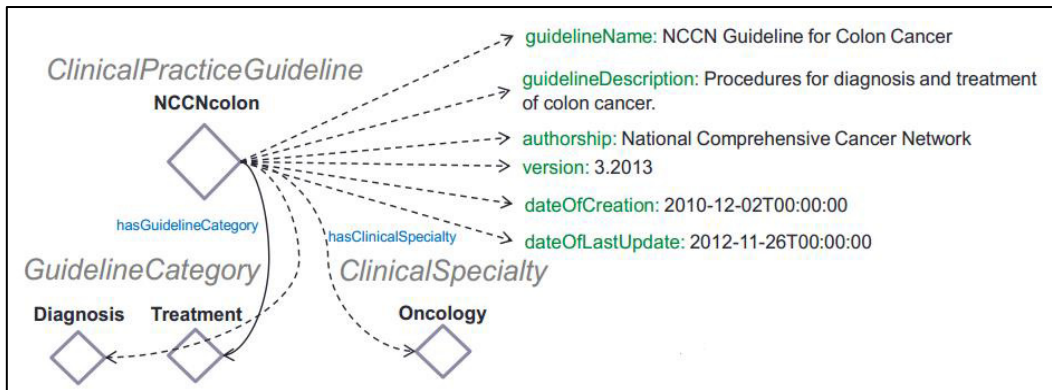


Figura 6 - Definição inicial de um protocolo clínico da National Comprehensive Cancer Network para o tratamento de cancro do cólon na ontologia CompGuide.

Cada instância que representa um protocolo clínico está ligada a uma instância da classe *Plan*, que é um contentor de tarefas, uma tarefa complexa. Por sua vez, uma instância de *Plan* está ligada a outras instâncias que representam tarefas básicas. Cada *Plan* pode, portanto, conter instâncias de outras tarefas inclusive outros *Plans*, assegurando assim a possibilidade de nidificar planos e trabalhar em diferentes níveis de execução.

Estas tarefas básicas são representadas através de três classes: *Action*, *Decision* e *Question*.

O objetivo destas classes é criar um plano de recomendações que contém referências a tipos de tarefas específicas. A classe *Action* expressa um procedimento que deve ser realizado por um profissional saúde. Existem vários subtipos de ações na ontologia que especificam a sua natureza com mais detalhes, nomeadamente exames, procedimentos, recomendações de medicação e recomendações simples.

A classe *Decision* é utilizada para realizar inferências sobre o estado do paciente. O exemplo mais óbvio de tal tarefa é o diagnóstico clínico.

A tarefa *Question* é utilizada para obter informações sobre os sinais e sintomas, condição de saúde ou outros parâmetros que podem ajudar a caracterizar o estado de um paciente. Esta classe também é usada para registar informações das observações do médico, e para guardar os resultados de exames clínicos. Este tipo de tarefa reúne toda a informação necessária para a execução do algoritmo clínico.

Através das propriedades de objetos, é possível definir as diferentes relações de controlo que podem existir entre tarefas, a sequência de execução de tarefas ou se elas devem ser

executadas simultaneamente ou paralelamente. Estas relações podem ser observadas na Figura

7. Relativamente a este ponto, é possível:

- Indicar a primeira tarefa de um *Plan*: neste caso a instância de *Plan* encontra-se ligada à instância da primeira tarefa através da propriedade *hasFirstTask*;
- Definir a execução sequencial de tarefas: se duas tarefas devem ocorrer uma após a outra, a primeira encontra-se ligada à segunda através da propriedade *nextTask*;
- Definir a execução paralela de tarefas: se tarefas devem ocorrer em simultâneo, a tarefa que as antecede encontra-se ligada a todas elas através da propriedade *parallelTask*. De modo a definir um ponto de sincronização para os caminhos de execução que se geram a partir desta situação, a tarefa que antecede as tarefas paralelas também se encontra ligada a uma tarefa de sincronização através da propriedade *syncTask*. Esta tarefa de sincronização representa o ponto no fluxo de execução para onde os diferentes caminhos de execução convergem;
- Definir a execução de tarefas alternativas: se o motor de execução deve seleccionar de forma automática, com base em condições sobre o estado do paciente, uma tarefa para executar de entre um conjunto de alternativas, a tarefa que as antecede encontra-se ligada a todas as alternativas através da propriedade *alternativeTask*. Por outro lado, se deve ser o profissional de saúde a seleccionar a tarefa alternativa a executar, a propriedade a utilizar é *preferenceAlternativeTask*.

Além disso, a ontologia fornece diferentes tipos de restrições clínicas expressas na forma de condições sobre o estado do paciente. Estas condições podem assumir a forma de:

- *TriggerConditions*: condições sobre parâmetros do estado do paciente expressas em termos quantitativos ou qualitativos que estão associadas a tarefas alternativas e que ditam a sua escolha. Uma tarefa alternativa é seleccionada somente se as suas *TriggerConditions* forem validadas;
- *PreConditions*: condições sobre parâmetros do estado do paciente expressas em termos quantitativos ou qualitativos que definem os casos em que uma tarefa pode ser executada.

- *Outcomes*: condições sobre parâmetros do estado do paciente expressas em termos quantitativos ou qualitativos que definem os objetivos de um *Plan* ou *Action*.

As *TriggerConditions*, *PreConditions*, *Outcomes* e condições de decisões são expressas através da classe *Condition* que é explicada com maior detalhe na secção 3.2.1.

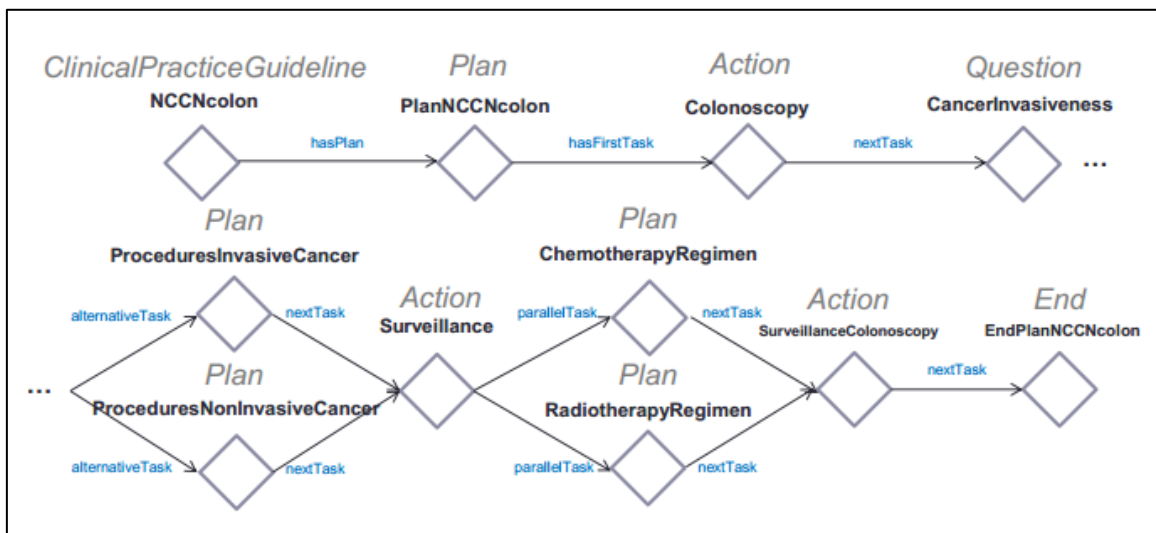


Figura 7 - Definição de um protocolo clínico do National Comprehensive Cancer Network para o tratamento de cancro do cólon na ontologia CompGuide, evidenciando as propriedades que permitem estabelecer uma ordem relativa entre as tarefas: hasFirstTask, nextTask, alternativeTask e parallelTask

Comparativamente com outros modelos CIG, CompGuide não requer qualquer proficiência em linguagens de programação, para definir estas condições, ao contrário das abordagens existentes. Além disso, proporciona uma maior expressividade na definição das tarefas e das relações de controlo.

### 3.2 Modelo de Representação Temporal

A definição de construtores de representação temporal segue a lógica do conteúdo descrito na secção 0. As classes do modelo de representação temporal encontram-se ilustradas na Figura 8. Todas as edições na ontologia CompGuide foram realizadas com a ferramenta de edição de ontologias Protégé. Assim, para representar as restrições temporais nos protocolos clínicos na ontologia CompGuide criou-se a classe *TemporalElement* e considerou-se que as principais classes são representadas como subclasses de *TemporalElement*. Uma dessas subclasses é *TemporalUnit*, que representa as diferentes granularidades que uma restrição

temporal pode ter, como segundo, minuto, hora, dia, semana, mês e ano. *TemporalUnit* é uma classe enumerada constituída apenas pelas instâncias *second*, *minute*, *hour*, *day*, *week*, *month* e *year*. Estas instâncias constituem as unidades de uma restrição temporal.

A classe *TemporalElement* possui sete subclasses que permitem especificar os padrões temporais: duração, periodicidade, repetição de tarefas, tempo de espera (que representam restrições à realização de tarefas) e restrições temporais em condições sobre o estado do paciente.

A representação do padrão duração baseia-se em elementos de Asbru [40] e para a representação das periodicidades, repetições e tempos de espera recorreu-se a elementos apresentados em Anselma *et al.* [51]. As classes que permitem representar estes padrões temporais são: *CyclePartDefinition*, *CyclePartPeriodicity*, *Duration*, *Periodicity*, *TemporalOperator*, *TemporalRestriction*, e *WaitingTime*.

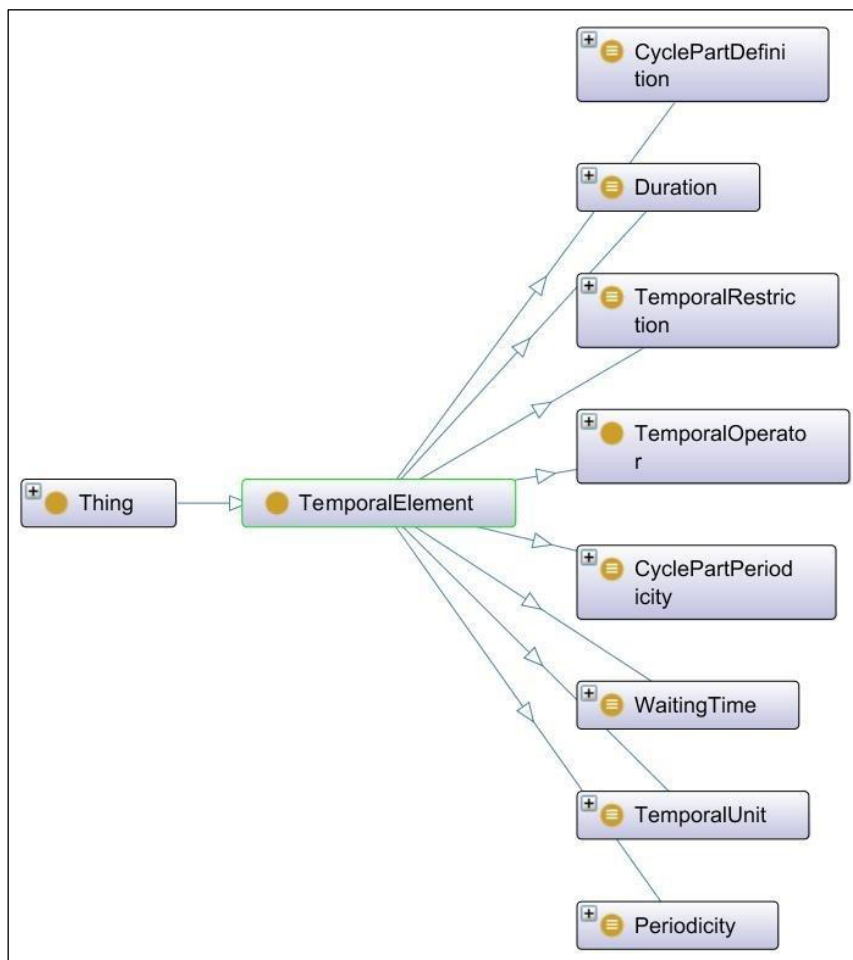


Figura 8 - Diagrama das classes OWL do modelo temporal na ontologia CompGuide.

As relações de controlo referidas no capítulo 3.1 são responsáveis não só pelo estabelecimento de um fluxo de trabalho de tarefas, mas também pela definição de restrições temporais qualitativas, ou seja, a ordem relativa de tarefas dentro do protocolo clínico. No momento da execução, o mecanismo de execução de protocolos clínicos analisa essas restrições e constrói um mapa de execução da tarefa. Como este é um aspeto que já foi incluído na ontologia e está em conformidade com as abordagens existentes, este capítulo apenas aborda as restrições temporais quantitativas.

### 3.2.1 Restrições Temporais à Execução de Tarefas

Expressar durante quanto tempo uma tarefa deve ser executada é um dos principais padrões temporais nos protocolos clínicos. No modelo proposto, a expressão do tempo é conseguida através da classe *Duration*. Os atributos que caracterizam esta classe são codificados como condições necessárias em OWL (como acontece com todas as outras classes). A classe *Duration* representa o padrão temporal de duração e especifica quanto tempo deve durar uma tarefa. É definida para *Plans* e *Actions*, uma vez que estas tarefas são as únicas que se podem desenrolar ao longo do tempo. As propriedades que definem esta classe são: *durationValue*, uma propriedade de dados do tipo *decimal* que corresponde à duração exata da tarefa; *minDurationValue* e *maxDurationValue*, também propriedades de dados do tipo *decimal*, que permitem especificar um tempo mínimo e máximo que uma *Action* ou *Plan* podem durar. A propriedade de objeto *hasDuration* permite ligar a tarefa em questão à instância de *Duration* que a caracteriza. Relativamente às propriedades de dados, o seu contradomínio assume valores numéricos decimais. Independentemente do tipo de valor, é sempre necessário definir uma unidade temporal para uma *Duration*, o que é feito através da propriedade de objeto *hasTemporalUnit* que liga as instâncias de *Duration* a instâncias de *TemporalUnit*. Em comparação com Asbru [59], esta forma de representação temporal é mais simples, pois não dispõe de anotações sobre horas de início e horas de término das tarefas, tornando-se mais fácil definir este padrão temporal.

De facto, na maioria dos casos, as informações sobre a duração das tarefas é transmitida quer como um valor exato ou como um intervalo, tal como no caso da expressão de linguagem natural "executar terapia neoadjuvante durante 2-3 meses" extraída do protocolo clínico para tratamento do cancro do cólon da NCCN [58], referindo uma ação que consiste na administração de agentes farmacêuticos antes do tratamento principal com a duração expressa

em um *minDurationValue* (2), um *maxDurationValue* (3), e uma *TemporalUnit* (mês). Desta forma, exclui-se a necessidade de anotações de tempo complexas. Por exemplo, na abordagem de Anselma *et al.* [51], as durações são sempre representadas por intervalos. Para expressar durações exatas é necessário definir que os limites do intervalo têm o mesmo valor. Deste modo, a estrutura de dados para durações exatas em CompGuide permite fazer o processamento de tais restrições de modo mais simples.

Neste sentido, é também possível expressar atrasos entre tarefas através da classe *WaitingTime*. A classe *WaitingTime* define o tempo a esperar até uma tarefa ser executada. Este padrão temporal pode ser necessário para verificar a manifestação dos efeitos de uma tarefa anteriormente aplicada, como pode ser observado na representação esquemática da Figura 9. Desta forma, se o tempo de espera de uma tarefa for um valor exato, utiliza-se a propriedade de dados *exactWaitingTime*. Caso o tempo de espera esteja vinculado a um intervalo de tempo, utiliza-se as propriedades de dados *minWaitingTime* e *maxWaitingTime*. Estas três propriedades de dados devem ser expressas em valores numéricos do tipo decimal. A unidade temporal é definida através da propriedade de objeto *hasTemporalUnit* que liga a instância de *WaitingTime* à respectiva instância de *TemporalUnit*. Para relacionar as tarefas com este padrão temporal existe a propriedade de objeto *hasWaitingTime*. Um *WaitingTime* pode ser definido para as tarefas *Action*, *Decision*, *Question* e *Plan*. Este padrão temporal permite representar situações como a descrita na seguinte expressão “reavaliação para cirurgia ao cólon 2 meses após o fim da quimioterapia” [58] em que claramente existem duas ações, a primeira é a quimioterapia e a segunda é a reavaliação que deve ser executada 2 meses (um *exactWaitingTime*) após o fim da primeira.

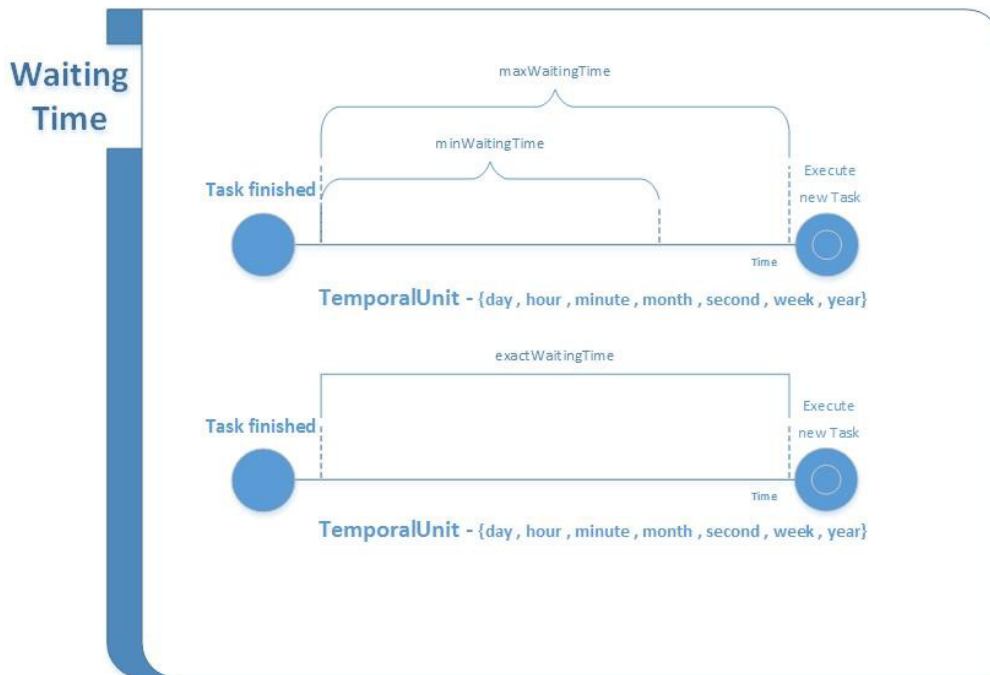


Figura 9 - Representação esquemática da classe *WaitingTime*.

A representação de tarefas periódicas é o padrão que apresenta maior complexidade. Em CompGuide, este padrão é representado através da classe *Periodicity*.

A classe *Periodicity* permite definir um ciclo de execução de uma tarefa, pelo que desta forma é possível especificar a periodicidade com que uma determinada tarefa deve ser aplicada, originando, assim, múltiplos eventos de execução na aplicação de um protocolo. Uma periodicidade pode ser definida para *Actions* e *Plans*. A frequência do evento periódico é definida através da propriedade de dados *periodicityValue* que deve assumir a forma de um valor numérico decimal, já a unidade temporal é definida através da propriedade de dados *hasTemporalUnit* e a respetiva referência a uma instância de *TemporalUnit*. Por último, a periodicidade de *Plans* e *Actions* define-se através da propriedade de dados *hasPeriodicity* que liga estas tarefas a uma instância de *Periodicity*.

Uma instância de *Periodicity* também pode estar ligada a uma instância da classe *Duration* através da propriedade de objeto *hasDuration*, determinando, assim, por quanto tempo uma tarefa periódica deve ocorrer. É de salientar que todos os eventos desta tarefa devem ser incluídos no intervalo de tempo definido pela duração, e que o valor da duração não pode ser inferior à frequência do evento periódico. Um exemplo de tal situação é a expressão “exame

físico todos os 6 meses durante 2 anos” [58], em que há claramente uma ação (exame físico), um *periodicityValue* (6) e *TemporalUnit* (mês), e um *durationValue* (2) e *TemporalUnit* (ano).

Quando se pretende indicar o número de vezes que um evento deve ser realizado, é necessário formular uma restrição de repetição, que é possível através da propriedade de dados *repetitionValue* com um intervalo de valores numéricos inteiros. Esta situação está patente na expressão “realizar uma colonoscopia anualmente, até perfazer um total de 3 colonoscopias” [58], em que há claramente uma ação (colonoscopia), um *periodicityValue* (1) e *TemporalUnit* (ano), e um *repetitionValue* (2).

Em caso de a tarefa periódica apenas ocorrer enquanto uma condição sobre o estado de um paciente se mantiver, é possível definir esta situação através da propriedade de objeto *hasStopConditionSet* que permite definir condições de paragem da tarefa. Esta propriedade liga uma instância de *Periodicity* a uma ou mais instâncias de *Condition* que representam as referidas condições. Depois de um evento da tarefa ter sido executado, de acordo com a sua periodicidade, cada uma das condições de paragem deve ser verificada e, se alguma delas se verificar, a tarefa deve ser interrompida. Cada instância de *Condition* é definida pelas seguintes propriedades:

- *numericalValue* ou *qualitativeValue*: ambas são propriedades de dados. A primeira é utilizada para o valor do parâmetro do estado do paciente a que se refere a condição, se este se expressar em termos numéricos. Deste modo, o contradomínio da propriedade é do tipo decimal. Por outro lado, se o parâmetro se expressar em valores nominais, utiliza-se a segunda propriedade, que apresenta assim um contradomínio do tipo *string*;
- *hasComparisonOperator*: é uma propriedade de objeto que liga uma instância da classe *Condition* a uma instância da classe *ComparisonOperator*. *ComparisonOperator* é uma classe enumerada, constituída pelas seguintes instâncias: *Different\_from*, *Equal\_to*, *Greater\_or\_equal\_than*, *Greater\_than*, *Less\_or\_equal\_than*, e *Less\_than*. Estes constituem operadores de comparação utilizados para expressar condições;
- *conditionParameter*: é uma propriedade de dados do tipo *string* que representa o parâmetro do estado clínico do paciente utilizado na condição;



- *unit*. é uma propriedade de dados do tipo *string* utilizada para expressar as unidades dos valores relativos aos parâmetros do estado do paciente.

Um exemplo de uma ação com uma condição de paragem está patente na expressão “realizar colonoscopia de 3 em 3 meses, durante 2 anos, parar se existirem indícios de adenoma” [58] em que existe uma tarefa periódica com uma duração (colonoscopia de 3 em 3 meses, durante 2 anos), mas depois expressa-se uma condição de paragem (indícios de adenoma).

Uma tarefa periódica é sempre limitada por um destes elementos: uma *Duration*, um *StopConditionSet*, um *repetitionValue*, uma *Duration* e um *StopConditionSet*, ou um *repetitionValue* e um *StopConditionSet*. A representação esquemática de uma *Periodicity* pode ser observada na Figura 10. Embora seja possível ter uma duração e uma condição de repetição, um valor de repetição e uma condição de repetição, ou apenas uma condição de repetição, não é possível ter uma duração e um valor de repetição ao mesmo tempo. A condição de paragem sobrepõe-se sempre às restantes restrições.

Cada evento, ou sub-tarefa de uma tarefa periódica pode apresentar, ele próprio, uma duração ou uma periodicidade. Para representar este padrão existe a propriedade de objeto *hasCyclePartDefinition* que liga uma instância de *Periodicity* a uma instância de *CyclePartDefinition*. Esta classe permite especificar a duração ou periodicidade com que cada sub-tarefa deve ocorrer em cada ciclo. Cada instância de *CyclePartDefinition* pode apresentar a propriedade *hasDuration*, uma propriedade de objeto que permite especificar a duração da sub-tarefa, através da reutilização da classe *Duration*, ou a propriedade *hasCyclePartPeriodicity* que permite especificar a periodicidade de cada sub-tarefa através da classe *CyclePartPeriodicity*. Cada instância de *CyclePartPeriodicity* é definida pela propriedade *hasDuration* ou por um *repetitionValue* (tal como na classe *Periodicity*), por um *periodicityValue* e uma *TemporalUnit*. No fundo, trata-se de definir uma periodicidade dentro de outra periodicidade. Neste caso não se optou pela reutilização da classe *Periodicity*, pois pretendeu-se limitar a nidificação de periodicidades, de modo a impossibilitar a definição de uma infinidade de ciclos dentro de ciclos. Um exemplo de uma *CyclePartDefinition* está presente na expressão “CapeOX de 3 em 3 semanas durante 3 meses, com a administração de capecitabina de 12 em 12 horas durante 14 dias” [58], em que é possível identificar uma ação (CapeOX, um regime de quimioterapia) com uma periodicidade principal que apresenta uma duração (3 em 3 semanas durante 3 meses),

definida através de *Periodicity*, e uma sub-tarefa (administração do fármaco capecitabina) que apresenta uma nova periodicidade definida através de *CyclePartDefinition* com uma *CyclePartPeriodicity*. Na instância de *CyclePartPeriodicity* correspondente existe um *periodicityValue* (12) e uma *TemporalUnit* (hora), bem como a definição de uma nova duração com *durationValue* (14) e *TemporalUnit* (dia).

Em termos de expressividade, esta abordagem permite a representação das restrições temporais que não podem ser representadas em Asbru, que é um dos modelos CIG dominantes. Trata-se, ao mesmo tempo, de uma adaptação (principalmente nos componentes de periodicidade) do formalismo apresentado em [51].

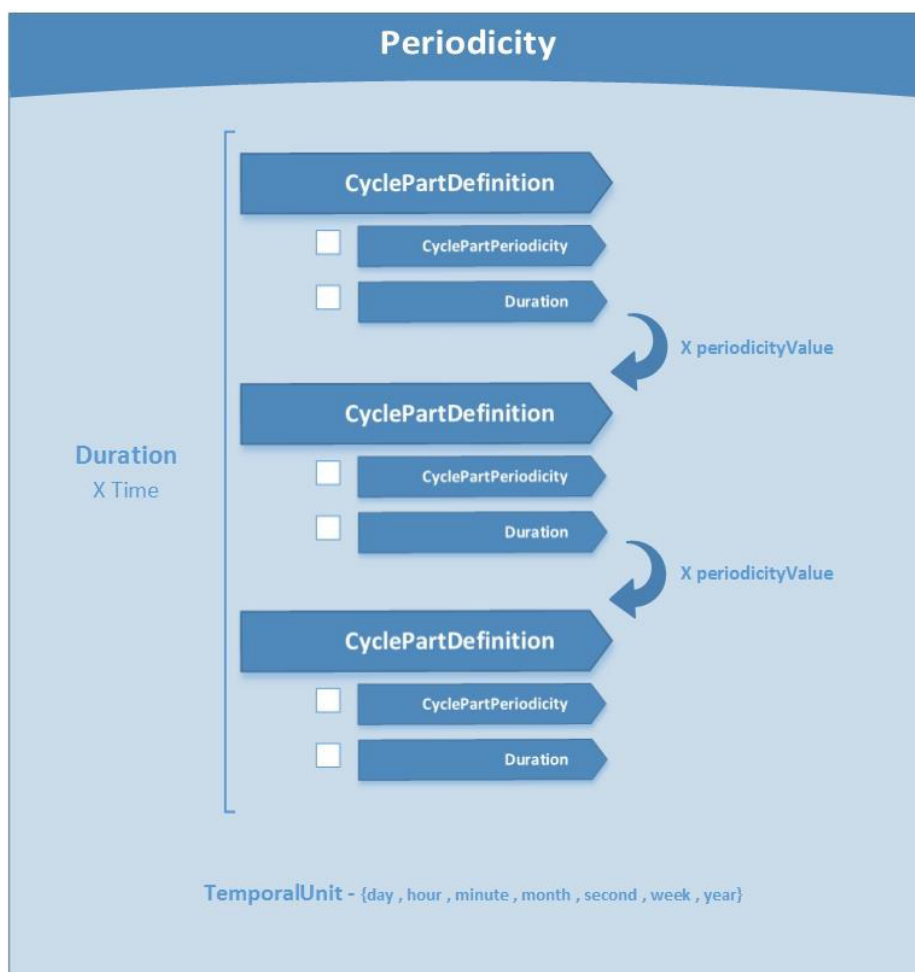


Figura 10 – Representação esquemática da classe Periodicity.

### 3.2.2 Restrições Temporais sobre o Estado do Paciente

A maioria das abordagens CIG não representa restrições temporais para as condições sobre o estado de um paciente. Contudo, quando o fazem, representam-nas como *strings* nos campos

de descrição. Considerando as diferentes condições que podem existir no CompGuide, tais como condições para decisões, *TriggerConditions*, *Outcomes*, e *PreConditions*, seria vantajoso desenvolver uma forma de raciocínio temporal automatizado sobre elas, tornando necessário o desenvolvimento de uma forma estruturada para representar as respectivas restrições temporais.

No modelo CompGuide, uma restrição temporal para uma instância de *Condition* é definida através da propriedade de objeto *hasTemporalRestriction* que liga esta instância a uma instância da classe *TemporalRestriction*. No entanto, não é obrigatório que uma *Condition* apresente uma restrição temporal. Para cada instância de *TemporalRestriction*, é necessário especificar um operador temporal através da propriedade de objeto *hasTemporalOperator*. Esta propriedade aponta para indivíduos da classe *TemporalOperator*, uma classe enumerada que tem um número limitado de instâncias, nomeadamente *within\_the\_last* e *within\_the\_following*.

Os operadores temporais representam o alcance de uma restrição temporal e são acompanhados de unidades temporais, definidas através da propriedade de objeto *hasTemporalUnit*. Para além disso, é necessário definir um valor para a restrição temporal, o que é possível através das propriedades de dados: *maxTemporalRestrictionValue*, *minTemporalRestrictionValue* e *temporalRestrictionValue*. Quando o valor da restrição temporal se expressa num intervalo, utiliza-se as propriedades *maxTemporalRestrictionValue* e *minTemporalRestrictionValue* para definir respetivamente o limite superior e inferior do intervalo. Por outro lado, se for um valor exato, utiliza-se a propriedade *temporalRestrictionValue*.

O operador temporal *within\_the\_last* é utilizado quando se deseja representar uma condição que se deve ter verificado, pelo menos uma vez, dentro de um período de tempo imediatamente anterior ao tempo de execução. Como referências para o mecanismo de raciocínio automático do motor de execução, utiliza-se o tempo de registo da observação correspondente ao parâmetro da condição e a data corrente da execução do protocolo. Este operador pode ser utilizado em restrições a condições de decisão, *TriggerConditions* e *PreConditions*, pois estas incidem sobre eventos passados. Como exemplo desta situação considere-se a expressão “para terapia após terceira progressão considerar regimes de quimioterapia experimentais, se o regime regorafenib tiver sido aplicado nos últimos 12 meses” [58], em que existe uma ação (administração de um regime de quimioterapia experimental), uma *TriggerCondition* cuja validação dita a escolha da tarefa (o regime regorafenib foi aplicado) e

uma uma *TemporalRestriction* definida por um *temporalRestrictionValue* (12), uma *TemporalUnit* (mês) e um *TemporalOperator* (*within\_the\_last*).

No entanto, quando se pretende expressar os resultados esperados da execução de uma tarefa, é necessário formular uma condição sobre o futuro, na qual se espera para observar o efeito que uma tarefa clínica tem depois de ser aplicada a um paciente. Para tal, é possível utilizar o operador *within\_the\_following*, o qual limita a observação de uma condição a um determinado período no futuro, a partir do momento de finalização de uma tarefa. Um exemplo de uma restrição temporal aplicada a um *Outcome* pode ser observado na expressão "o tumor deve tornar-se operável após 6 meses da finalização dos regimes de quimioterapia FOLFOX ou CapeOX " [58], em que existe uma ação (quimioterapia FOLFOX ou CapeOX), uma condição que expressa um *Outcome* (o tumor deve tornar-se operável) e uma *TemporalRestriction* definida por um *temporalRestrictionValue* (6), uma *TemporalUnit* (mês) e um *TemporalOperator* (*within\_the\_following*).

### 3.3 Discussão e Análise do Modelo Proposto

Os exemplos utilizados ao longo deste capítulo foram retirados de um protocolo clínico da NCCN para o tratamento do cancro do cólon [58]. Este protocolo compreende os procedimentos desde o estadiamento do cancro até ao acompanhamento do paciente após o tratamento e foi selecionado pela sua elevada complexidade bem como variedade dos seus padrões temporais, sobretudo no que toca à recomendação de regimes de quimioterapia. Deste modo, procedeu-se à representação da totalidade do protocolo clínico na ontologia CompGuide, produzindo-se assim um ficheiro *owl* que representa uma CIG. O ficheiro que resultou deste processo de representação contém 696 instâncias no total, das quais apenas 223 representam tarefas clínicas. O número de ocorrências de cada tipo de tarefa está representado na Tabela 2. De realçar que o tipo de tarefa com maior representatividade é *Action*, devido ao elevado número de regimes de quimioterapia que são recomendados e que só podem ser mapeados para ações. No outro extremo, o tipo de tarefa com menor representatividade é *Decision*, verificando-se apenas uma ocorrência deste tipo. Tal deve-se à natureza das decisões presentes no protocolo que são mais facilmente mapeáveis para situações de transição entre tarefas, em que é necessário recorrer a *TriggerConditions* para escolher a tarefa a executar de seguida. Pode-se dizer que

neste protocolo não é possível isolar tarefas de decisão pura, com exceção de um caso, dada a ausência de tarefas de diagnóstico.

Tabela 2 – Número de ocorrências de cada tipo de tarefa no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide.

<b>Tipo de Tarefa</b>	<b>Número de instâncias</b>
Action	190
Question	21
Decision	1
Plan	11
<b>Total</b>	223

Quanto à representação de restrições temporais à execução de tarefas, verificou-se que das 223 tarefas, 95 apresentam padrões temporais, na sua grande maioria do tipo *Periodicity*, como se pode ver na Tabela 3. A maior parte destas tarefas periódicas apresenta-se limitada por uma *Duration*. Contudo, verificou-se a ocorrência de todos os casos possíveis de limitação de uma tarefa periódica, com a conjugação de durações, condições de paragem e número de repetições. Estas ocorrências correspondem a casos semelhantes aos exemplos dados na secção 3.2.1. Na sua maioria, as tarefas periódicas consistem em recomendações de regimes de quimioterapia. Os casos em que estes regimes são mais complexos e em que é necessária a administração de determinados fármacos com uma certa periodicidade, dentro da periodicidade do regime de quimioterapia que está a ser seguido, dão origem a restrições do tipo *Periodicity* com *CyclePartDefinition*.

Os construtores temporais propostos no modelo revelaram-se suficientes para representar os padrões temporais do protocolo clínico, sobretudo nos casos de *Duration* e *WaitingTime* disponíveis. Deste modo, é possível equiparar este modelo aos modelos de Asbru [40] e de Anselma *et al.* [51]. No entanto, há uma limitação que deve ser reconhecida. Ao contrário de Anselma *et al.* [51], o modelo proposto não permite representar valores de periodicidade na forma de intervalos, permitindo apenas a definição de valores exatos para este padrão temporal.

Tabela 3 – Número de ocorrências de cada tipo restrição temporal à execução de tarefas no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide.

<b>Restrição Temporal</b>	<b>Número de tarefas</b>
Duration	7
WaitingTime	2
Periodicity	79
Periodicity (com CyclePartDefinition)	7
<b>Total</b>	<b>95</b>

Relativamente às restrições temporais sobre condições do estado do paciente, verificaram-se apenas 6 ocorrências em todo o protocolo, tal como se pode observar na Tabela 4. As *TemporalRestrictions* que apresentam o *TemporalOperator within\_the\_last* referem-se a *TriggerConditions* que foram colocadas a uma tarefa, tal como no exemplo dado na secção 3.2.2. O mesmo se passa com as *TemporalRestrictions* que apresentam o *TemporalOperator within\_the\_following*, pois ambas expressam *Outcomes* de tarefas tal como no exemplo. Há que referir que nos modelos existentes, exceto GLIF3 [52], não seria possível expressar este tipo de restrições para interpretação automática.

O que se pretende com esta proposta é um modelo inclusivo, capaz de englobar os diferentes padrões temporais dos protocolos clínicos, o que representa algo diferente do que existe atualmente, ou seja, os modelos existentes focam-se particularmente num padrão temporal, descorando os restantes. Assim, dado o que foi exposto, pode-se considerar que este objetivo foi atingido.

Tabela 4 – Número de ocorrências das restrições temporais sobre condições do estado do paciente no protocolo da NCCN para tratamento de cancro do cólon na ontologia CompGuide.

<b>Operador Temporal</b>	<b>Número de tarefas</b>
within_the_last	4
within_the_following	2
<b>Total</b>	<b>6</b>

## 4 Análise do Problema de Execução de Protocolos Clínicos

Neste capítulo serão abordados vários pontos essenciais para que se possa considerar que o projeto final cumpre o objetivo proposto. Pretende-se analisar a problemática subjacente à execução de protocolos clínicos, atendendo às suas restrições temporais. Para tal é crucial abordar um conjunto de elementos que descrevem a compreensão do domínio, as funcionalidades do sistema e os atores do sistema.

Na secção *Modelo de Domínio*, apresenta-se a informação e compreensão do domínio em que se insere o projeto. De seguida, na secção *Apresentação dos Atores do Sistema*, fornece-se detalhes sobre os seus utilizadores. Posteriormente, em *Análise e Levantamento de Requisitos* são abordados os requisitos básicos necessários e previstos alguns requisitos complementares, para que se possa, de alguma forma, dar um valor acrescido a essa mesma solução. Procurou-se também estabelecer alguns requisitos não funcionais que permitam que a experiência dos utilizadores possa ser valorizada. Por fim, em *Casos de Uso*, apresenta-se um diagrama separado por *packages* bem como uma descrição textual.

### 4.1 Modelo de Domínio

O termo domínio, no contexto da Engenharia de Software, é utilizado para denotar ou agrupar um conjunto de sistemas ou de áreas funcionais que exibem funcionalidades similares. Podemos então descrevê-lo como o modelo que representa a compreensão e informação adquirida acerca do domínio. Assim sendo, este modelo é uma representação de classes conceptuais do mundo real, não de componentes de *software*.

Será, portanto, feita uma descrição do modelo de domínio que foi elaborado e respetiva explicação de cada uma das entidades. Estas entidades encontram-se representadas na Figura 11. De referir que este modelo apenas se foca na representação das restrições temporais dos protocolos clínicos.

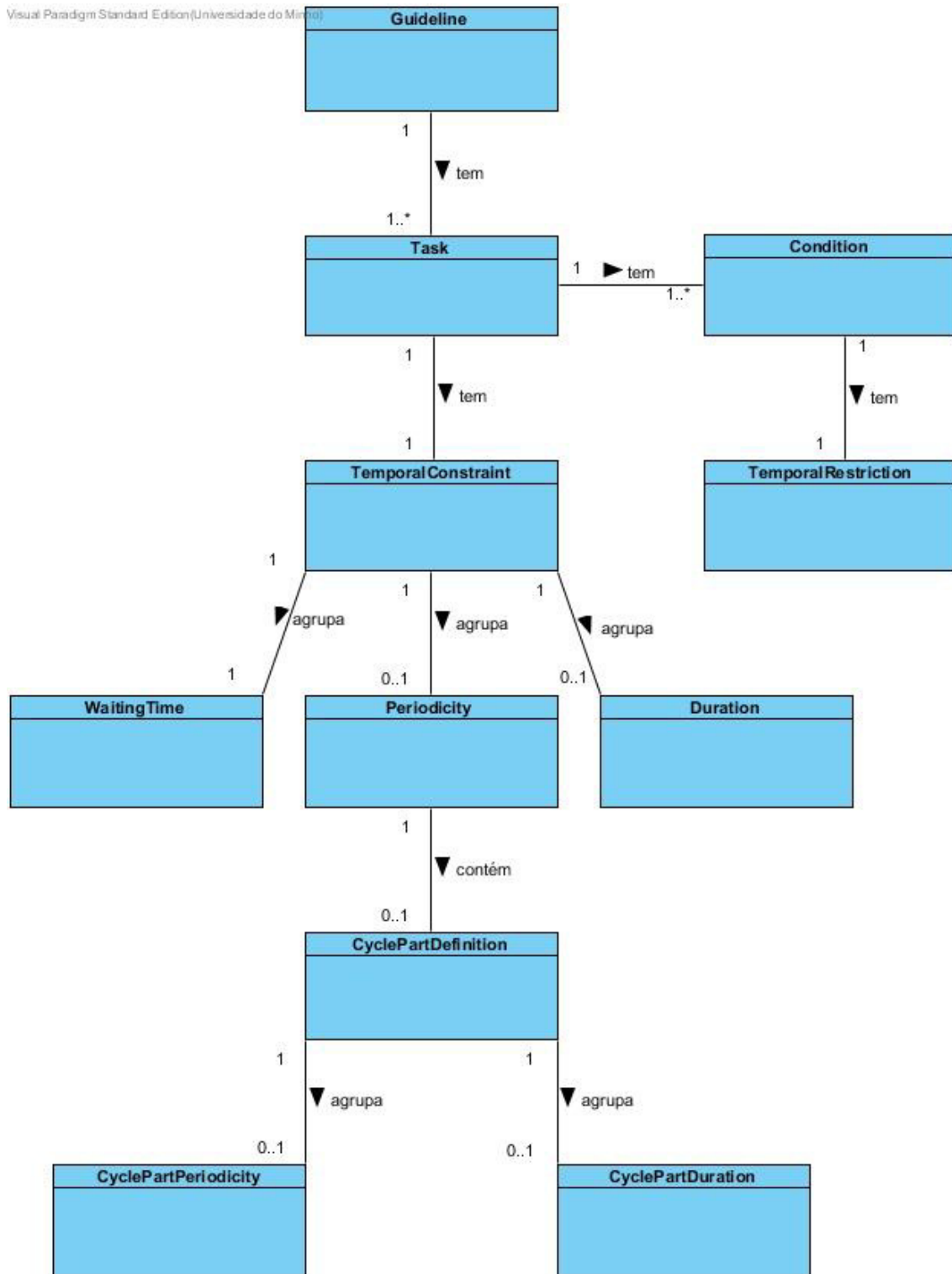


Figura 11 - Modelo de domínio.

De seguida, apresentam-se as principais entidades de domínio, bem como a sua descrição:



- ***Guideline***

*Guideline* é um conceito muito importante, uma vez que esta entidade representa os protocolos clínicos que pretendem afetar a prática clínica. Cada protocolo clínico tem várias tarefas que devem ser geridas pelos profissionais de saúde.

- ***Task***

A entidade *Task* representa as tarefas que os profissionais de saúde devem executar ao longo do tempo. Estas tarefas apresentam restrições temporais que afetam a sua lógica de execução.

- ***Condition***

A entidade *Condition* representa condições acerca de parâmetros do estado do paciente.

- ***TemporalRestriction***

*TemporalRestriction* representa as restrições temporais colocadas às condições sobre parâmetros do estado do paciente que se traduzem em horizontes temporais nos quais se terão manifestado ou manifestar-se-ão sinais ou sintomas.

- ***TemporalConstraint***

Esta entidade representa as restrições temporais presentes nas tarefas dos protocolos clínicos.

- ***WaitingTime***

A entidade *WaitingTime* representa o tempo de espera de uma determinada tarefa, até poder ser executada.

- ***Periodicity***

A entidade *Periodicity* representa o ciclo de execução de uma tarefa, isto é, a frequência com que uma determinada tarefa deve ser executada.

- ***Duration***

*Duration* representa o tempo que deve durar uma tarefa.

- ***CyclePartDefinition***

A entidade *CyclePartDefinition* representa o conceito relativo a cada parte de um ciclo que pode ocorrer num evento periódico. Cada parte de um ciclo pode ser constituída por uma duração ou por um evento periódico.

- ***CyclePartPeriodicity***

Esta entidade pretende representar a periodicidade de cada parte de um ciclo que pode ocorrer num evento periódico.

- ***CyclePartDuration***

Esta entidade pretende representar a duração de cada parte de um ciclo que pode ocorrer num evento periódico.

## **4.2 Apresentação dos Atores do Sistema**

Neste projeto foram identificados dois tipos de utilizadores: Profissionais de Saúde e Administrador.

De uma forma global, os atores são os elementos que irão interagir com o sistema de alguma forma. Estando apenas de visita ou sendo utilizadores habituais, são representados como elementos importantes do mecanismo de funcionamento do sistema.

- **Profissionais de Saúde**

São os utilizadores responsáveis por iniciar a execução de protocolos clínicos para os seus pacientes, bem como executar e verificar as tarefas, isto é, devem confirmar na plataforma se uma determinada tarefa, que diz respeito a um paciente, foi ou não efetuada e executada.

- **Administrador**

Administrador é o ator responsável por gerir toda a informação da plataforma, desde a gestão dos protocolos clínicos, gestão de pacientes, gestão dos profissionais de saúde bem como a gestão das tarefas que devem ser executadas pelos profissionais de saúde.

## 4.3 Análise e levantamento de Requisitos

Nesta secção pretende-se especificar os requisitos funcionais e não funcionais da aplicação. Para tal, não se recorreu a nenhum modelo de representação (como o modelo de Volere [60]) pelo que a sua especificação será apenas textual. Relativamente às técnicas de levantamento de requisitos foram utilizadas as técnicas de introspeção, análise do enunciado/proposta de dissertação e análise de *sites* similares, no entanto, não será especificada a técnica utilizada em cada requisito.

### 4.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que se espera que o sistema disponibilize, de uma forma completa e consistente. Deste modo, a aplicação deve:

- Permitir o registo de profissionais de saúde e pacientes na aplicação;
- Permitir a edição da informação pessoal dos utilizadores;
- Permitir a visualização das tarefas que os profissionais de saúde devem cumprir, através de um calendário e de um cronograma, tendo em conta as restrições temporais definidas no protocolo;
- Permitir a visualização da informação sobre a tarefa, nomeadamente o tempo que falta para a tarefa terminar, o número de vezes que a tarefa foi repetida, o número de vezes que a tarefa deve repetir-se;
- Notificar o utilizador quando alguma tarefa está a terminar;
- Notificar o utilizador quando alguma tarefa deve começar;
- Permitir aos profissionais de saúde começar uma determinada tarefa;
- Permitir aos profissionais de saúde confirmar que uma determinada tarefa foi cumprida (terminar a tarefa);
- Permitir ao administrador gerir os protocolos clínicos, isto é, criar, editar, e atualizar a sua informação bem como executar um determinado protocolo;
- Permitir ao administrador gerir os profissionais de saúde e pacientes, isto é, permitir editar a sua informação, adicionar e eliminar estes utilizadores;
- Permitir a execução paralela de tarefas, sendo obrigatório a execução de todas as tarefas;

- Permitir a execução alternativa de tarefas, isto é, o utilizador deverá escolher das tarefas disponíveis a que pretende executar.

### 4.3.2 Requisitos Não Funcionais

Os requisitos não funcionais são os que estão relacionados com o uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Em geral, requisitos não funcionais podem constituir restrições aos requisitos funcionais e não é necessário que o cliente os identifique, pois são características mínimas de um *software* de qualidade.

Os requisitos não funcionais elencados são os seguintes:

- Disponibilizar um portal *web*;
- Desenvolver e implementar a aplicação utilizando apenas tecnologias gratuitas;
- Desenvolver o *site* utilizando *responsive design*, permitindo ajustar automaticamente o layout das páginas *web* ao tamanho da janela;
- Aspeto: deve ter um aspeto apelativo, interativo e de fácil manuseamento.
- Usabilidade: os utilizadores deverão saber usar o sistema num curto intervalo de tempo;
- Escalabilidade: o sistema deve poder manter o mesmo desempenho (tempo de resposta) quando há um aumento no número de utilizadores e/ou pedidos simultâneos;
- A interface deve seguir um padrão de apresentação (não ser diferente de janela para janela);
- Performance/eficiência: o sistema deverá processar um evento num determinado intervalo de tempo;
- Segurança: as *passwords* devem ser encriptadas;
- Portabilidade: a solução deverá executar na maioria das plataformas;
- Mobilidade: a solução deve incluir a possibilidade de utilização a partir de dispositivos móveis;
- Requisitos éticos: o sistema não deve usar as informações dos utilizadores para fins comerciais externos;
- Rastreabilidade: Deve ser possível construir o histórico de operações dos utilizadores.

## 4.4 Casos de Uso

Os casos de uso descrevem as funcionalidades que serão desenvolvidas no projeto, ou seja, representa a interação entre um utilizador e o sistema de uma forma clara e perceptível para os *stakeholders*.

De seguida, apresenta-se as interações dos diferentes utilizadores com o sistema.

### 4.4.1 Diagrama de Casos de Uso

Um diagrama de casos de uso é um modelo que descreve como os diferentes tipos de atores interagem com o sistema para resolver um problema proposto. Como tal, descreve os objetivos dos atores, as interações entre os atores e o sistema, bem como o comportamento necessário do sistema para satisfazer esses objetivos.

Este tipo de diagrama é constituído por ligações que ligam o ator aos casos de uso em si. Segundo Ivar Jacobson [61], um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo.

Para um melhor entendimento do diagrama dividiu-se a sua apresentação por packages, representados nas Figuras 12 - 15. Os packages representam as seguintes situações:

- **Package Gestão de Profissionais de Saúde**

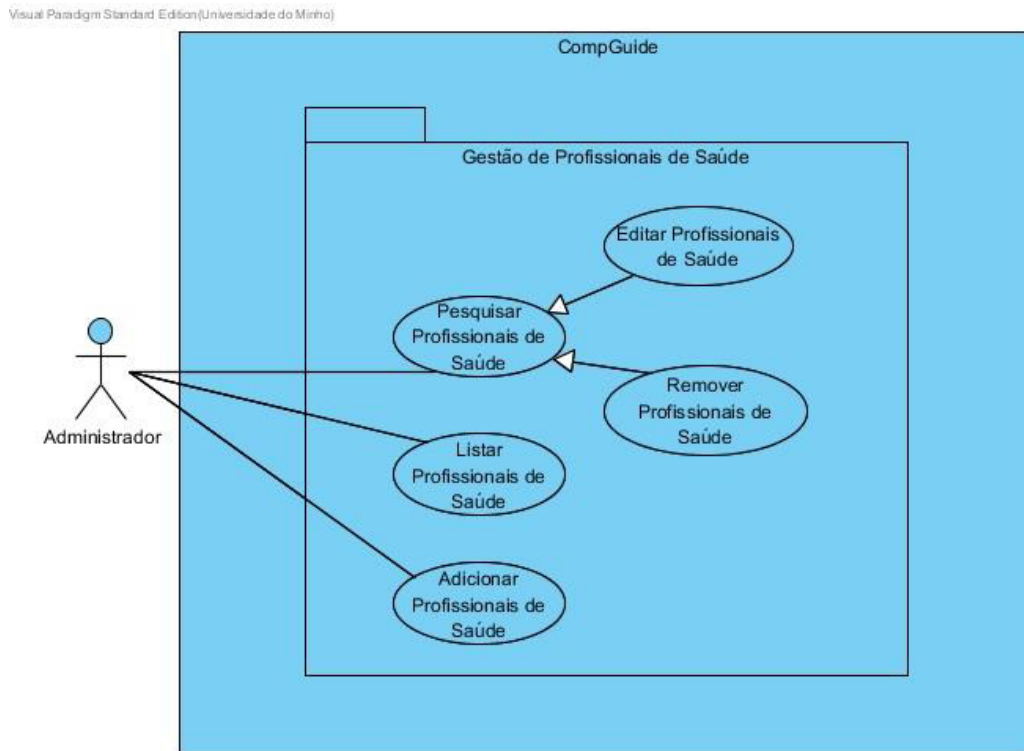


Figura 12 - Diagrama de casos de uso relativo ao package Gestão de Profissionais de Saúde.

- **Package Gestão de Protocolos clínicos**

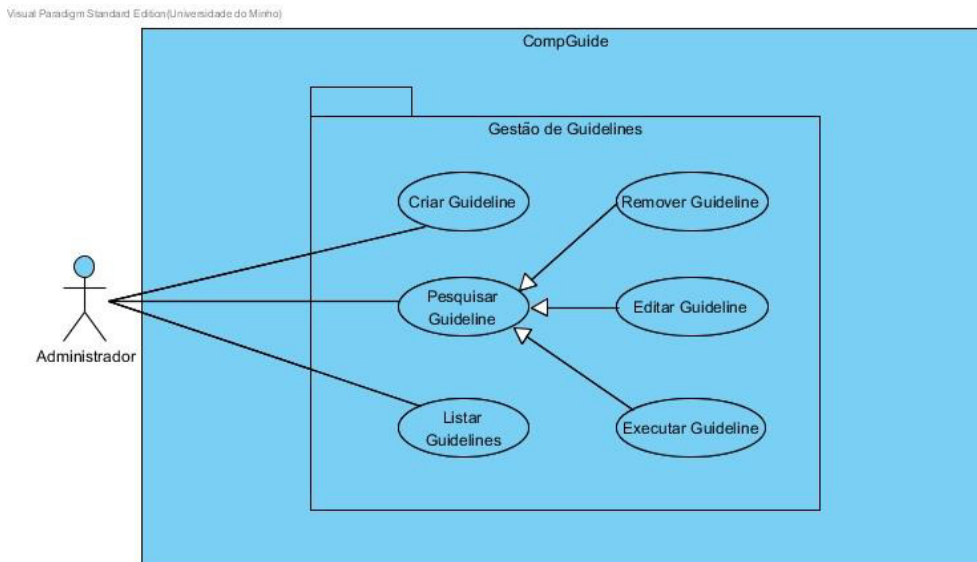


Figura 13 - Diagrama de casos de uso relativo ao package Gestão de Guidelines.

- **Package Gestão de Tarefas**

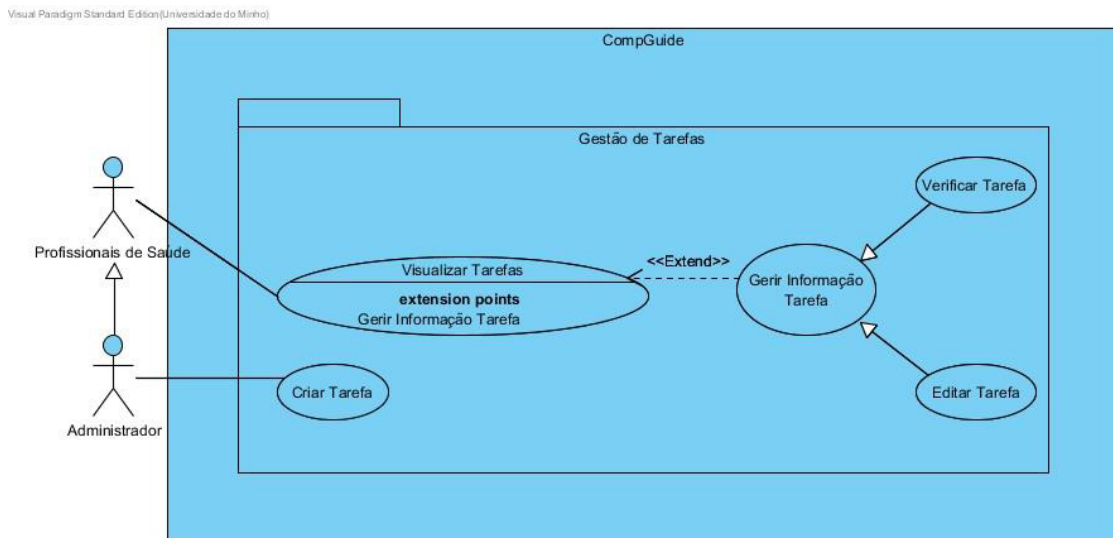


Figura 14 - Diagrama de casos de uso relativo ao package Gestão de Tarefas.

- **Package Gestão de Pacientes**

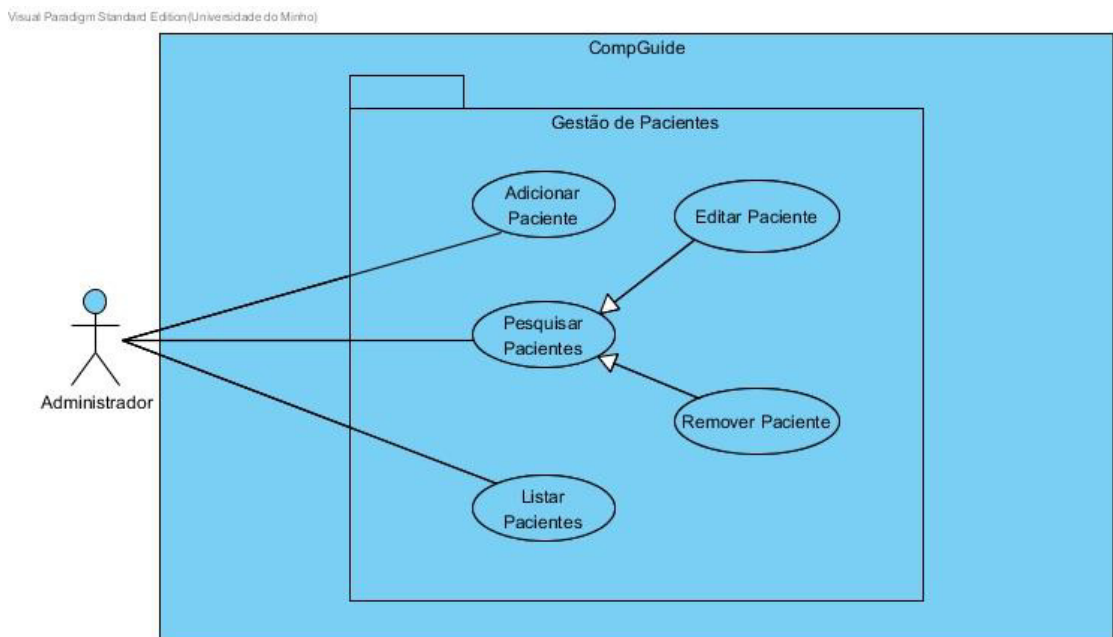


Figura 15 - Diagrama de casos de uso relativo ao package Gestão de Pacientes.

### 4.4.2 Descrição dos Casos de Uso

Neste capítulo, pretende-se fazer a descrição textual dos casos de uso. A descrição de casos de uso é uma forma de organizar os diferentes cenários, acrescentando detalhes adicionais, além do que é apresentado no diagrama. Assim sendo, a sua descrição tem como objetivo definir os requisitos para que todos os intervenientes no projeto os possam entender. Destaque para os casos de uso da Tabela 5 e Tabela 6. No Anexo A encontram-se as restantes descrições textuais.

Tabela 5 - Descrição do caso de uso Pesquisar Profissional de Saúde.

<b>Nome</b>	<b>Pesquisar Profissional de Saúde</b>
<b>Propósito</b>	Permitir pesquisar um Profissionais de Saúde que esteja registado no sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Pesquisar Profissional de Saúde”
<b>OUT</b>	2. O Sistema disponibiliza um campo para introdução do nome do Profissional de Saúde.
<b>INP</b>	3. O Administrador introduz o nome do Profissional de Saúde.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>OUT</b>	5. O Sistema apresenta a lista de Profissionais de Saúde que correspondem ao valor introduzido.
<b>UD</b>	6. O Administrador seleciona do Profissional de Saúde
<b>OUT</b>	7. O Sistema fornece as várias opções de gestão do Profissional de Saúde.



<b>ALT</b>	<b>Alternativa 4a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são inválidos.
	2. Volta a 2.
<b>ALT</b>	<b>Alternativa 5a [Profissionais de Saúde não encontrados]:</b>
	1. Sistema informa que não foram encontrados Profissionais de Saúde.
	2. Volta a 2.

Tabela 6 – Descrição do caso de uso Pesquisar Protocolos clínicos.

<b>Nome</b>	<b>Pesquisar Protocolos clínicos</b>
<b>Propósito</b>	Permitir pesquisar um Protocolos clínicos que esteja registado no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Pesquisar Protocolos clínicos”
<b>OUT</b>	2. O Sistema disponibiliza um campo para introdução do nome do Protocolos clínicos.
<b>INP</b>	3. O Administrador introduz o nome do Protocolos clínicos.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>OUT</b>	5. O Sistema apresenta a lista de Protocolos clínicos que correspondem o valor introduzido.

<b>UD</b>	6. O Administrador seleciona o Protocolos clínicos
<b>OUT</b>	7. O Sistema fornece as várias opções de gestão da Protocolos clínicos.
<b>ALT</b>	<b>Alternativa 4a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são inválidos.
	2. Volta a 2.
<b>ALT</b>	<b>Alternativa 5a [Protocolos clínicos não encontradas]:</b>
	1. Sistema informa que não foram encontrados protocolos clínicos.
	2. Volta a 2.

## 4.5 Discussão e Análise

As entidades definidas no modelo de domínio refletem a visão dominante sobre a execução automática de CIGs que seguem um modelo em rede de tarefas. Deste ponto de vista, um protocolo é representado como um conjunto de tarefas interligadas, às quais se aplicam restrições ou condições de execução, quer relativas ao estado do paciente, quer temporais. Esta organização resultou de reuniões com um profissional de saúde e da análise de diferentes protocolos que, em última análise, permitiram validar o conjunto de requisitos definidos.

Através dos requisitos funcionais definidos, nomeadamente a visualização das tarefas que os profissionais de saúde devem cumprir, a visualização da informação sobre as tarefas e as diferentes notificações temporais, a intenção foi criar um assistente pessoal que permita aos profissionais de saúde gerir os seus pacientes bem como gerir o conjunto de tarefas relativas à execução de protocolos clínicos sobre estes, mantendo-se deste modo atualizados em relação aos procedimentos que foram aplicados e que deverão ser aplicados no futuro. Os requisitos foram posteriormente transformados num conjunto de casos de uso que descrevem as funcionalidades do sistema.

## 5 Desenvolvimento e Implementação do Sistema de Execução de Protocolos Clínicos

Este capítulo descreve todo o processo de desenvolvimento e implementação da solução e compreende o delineamento das tecnologias e ferramentas utilizadas, a elaboração da arquitetura do sistema, bem como o conjunto de artefactos em UML orientados para o *design* da solução.

Após a identificação dos requisitos (funcionais e não funcionais) e identificadas as entidades de domínio será aqui dada uma visão geral da arquitetura da aplicação.

Na secção *Tecnologias e Ferramentas Utilizadas* fornece-se uma visão geral das tecnologias envolvidas no projeto. Em *Arquitetura de Software* discute-se o modelo de *software* aplicado no desenvolvimento da solução. As secções *Modelo Relacional*, *Diagrama de Classes*, *Diagramas de Sequência* e *Desenvolvimento Baseado em Padrões de Software* visam fornecer detalhes adicionais sobre a organização do sistema e o mapeamento do modelo de domínio para o código. As interfaces disponibilizadas na aplicação para execução de protocolos clínicos são apresentadas na secção *Apresentação da Aplicação Web*. Para finalizar o capítulo, em *Discussão e Análise da Solução*, realiza-se uma reflexão sobre o sistema desenvolvido tendo em conta os objetivos definidos no início deste trabalho de dissertação.

### 5.1 Tecnologias e Ferramentas Utilizadas

Para o desenvolvimento do projeto, foi necessário recorrer a várias tecnologias, que permitiram criar uma solução capaz de resolver o problema proposto, bem como proporcionar uma interface agradável para o utilizador final da aplicação. As tecnologias foram previamente analisadas antes do início do projeto com base em fatores como a portabilidade e a capacidade evolutiva.

Relativamente às tecnologias inerentes ao Servidor, foram utilizadas as seguintes:

- *Java Server Faces* (JSF) para o desenvolvimento *web*, nomeadamente desenvolvimento da lógica da aplicação;
- *Java Persistence API* (JPA) para a persistência de dados, mais concretamente, para definir o mapeamento das tabelas de bases de dados para objetos Java;

- *Hibernate* para a ligação entre as classes *Bean* e a base de dados MySQL;
- *RESTfull Web Service* para acesso aos dados da base de dados;

Relativamente às tecnologias inerentes à Interface, foram utilizadas as seguintes:

- HTML5 para o desenvolvimento de páginas *web*;
- CSS3 para definir o estilo das páginas *web*;
- *BootStrap* uma vez que é uma *framework* que facilita o desenvolvimento de aplicações *web* e mobile *responsive* com aspeto atrativo e moderno;
- *BootFaces* porque permite utilizar o melhor de *Bootstrap 3* e *jQuery* para facilitar o desenvolvimento do *front-end* da aplicação;
- *Primefaces* uma vez que permite usar *widgets* que visam facilitar a interação do utilizador com a aplicação.
- *JavaScript* uma vez que adiciona conteúdo dinâmico às páginas e enriquece as interfaces do utilizador;
- AJAX para reduzir a necessidade de recarregar páginas completas, e desta forma tornar as páginas *web* mais interativas com o utilizador, utilizando-se assim solicitações assíncronas de informações;
- *jQuery* uma vez que fornece interações, *widgets*, efeitos e temas para a criação de aplicações *web* interativas e com um aspeto apelativo.

Relativamente ao Sistema de Gestão de Base de Dados (SGBD) foi utilizado o MySQL, uma vez que é *open source* e permite definir bases de dados relacionais. O ambiente de desenvolvimento usado foi o Netbeans IDE 8.0.2 porque permite criar aplicações *web* que utilizem JSF e além disso fornece um conjunto de ferramentas que visam facilitar todo o processo de desenvolvimento.

## 5.2 Arquitetura de Software

Esta aplicação *web* baseia-se num modelo computacional *request-response* entre cliente e servidor.

O projeto contém dois componentes essenciais. O componente servidor que implementa a *framework* Model-View-Controller (MVC) JSF, bem como a camada de acesso a dados que utiliza *Hibernate* para fazer o mapeamento entre as entidades de classe e as tabelas da base de dados.

Relativamente à componente cliente, esta é responsável por tornar a interação com o utilizador mais dinâmica.

Esta separação em camadas permite uma fácil integração de novos requisitos. A implementação de padrões de *software* como o MVC não só permite uma maior escalabilidade da aplicação, mas também uma maior independência entre classes, que se concretiza numa redução do impacto, considerando as alterações que podem surgir no resto da aplicação. O modelo MVC [62] separa a representação da informação da interação com o utilizador. O *Model* consiste nos dados da aplicação, regras de negócio, lógica e funções. A *View* é o output da representação dos dados. O *Controller* é responsável pela mediação da interação entre o *Model* e a *View*. O esquema da Figura 16 apresenta a lógica de funcionamento do modelo MVC, ilustrando a forma como um pedido é tratado e direcionado para a solução.

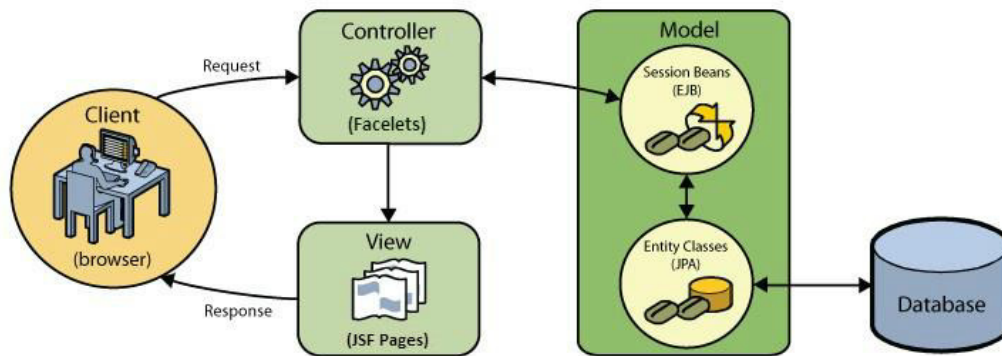


Figura 16 - Arquitetura típica segundo um modelo Model-View-Controller (figura retirada de [63]).

A Figura 17 apresenta a arquitetura idealizada para implementação deste projeto. A aplicação está separada em áreas funcionalmente isoladas, sendo que normalmente esta divisão é feita nas seguintes camadas: a camada de apresentação, a camada de negócio e a camada de dados. Para ser possível estruturar a aplicação em camadas, é necessário ter um servidor que permita a comunicação com o *browser*, recebendo pedidos e enviando as respetivas respostas. Desta forma, o servidor escolhido foi o *Wildfly*, que é um servidor de aplicações *web open source*.

O *Wildfly* implementa as especificações de *Java Facelets* e de JSF, fornecendo desta forma todas as condições para a execução do código java.

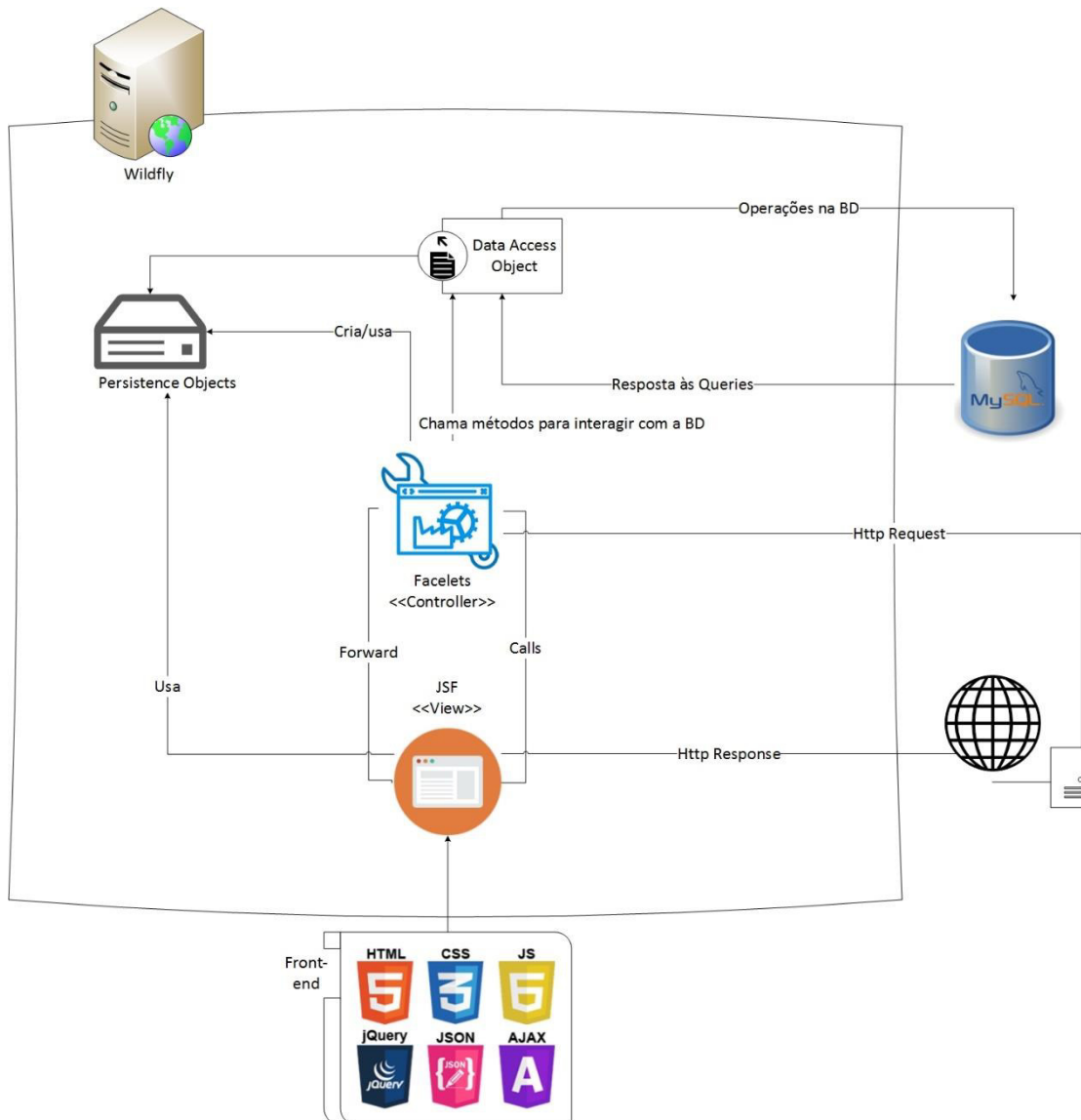


Figura 17 - Arquitetura do sistema para execução de protocolos clínicos.

A camada de apresentação consiste nos componentes que tratam da interação entre as *Views* e a camada de negócio. Esta camada disponibiliza conteúdo dinâmico (em HTML5, *Primefaces*, *Bootstrap*, *CSS3*, *JavaScript*, *jQuery* e *AJAX*) e faz a mediação das interações entre as *Views* e o *Model*, pelo controlador.

Na camada de negócio, a tecnologia utilizada foi *Facelets*. Esta tecnologia é um sistema de *templates web* que expande as funcionalidades de um servidor, gerando dados HTML e XML para a camada de apresentação. É basicamente uma classe na linguagem de programação Java

que processa dinamicamente pedidos e respostas, através dos protocolos *HttpRequest* e *HttpResponse*, fazendo assim um ponto de ligação entre as páginas *html* e o *browser*, atribuindo assim novos recursos ao servidor.

A camada de dados utiliza a JPA que permite a persistência de dados. Esta API define o mapeamento das tabelas da base de dados para objetos Java denominados de POJOS ou entidades *Bean*. A ponte de ligação entre as classes e a base de dados MySQL é gerida pelo *Hibernate* (através da JPA 2.1).

### 5.3 Modelo Relacional

O modelo relacional é um modelo de dados subjacente a um SGBD que se baseia no princípio de que todos os dados estão guardados em tabelas (ou, em termos lógicos, relações).

O modelo relacional utilizado encontra-se na Figura 33 no Anexo B. O diagrama da figura é uma representação das classes existentes na ontologia CompGuide em OWL. No entanto, em OWL a definição das relações entre classes é mais complexa e expressiva, pelo que a sua definição em base de dados é muito difícil. Neste sentido, o motor de execução é essencial, uma vez que é responsável por interpretar todo o conteúdo da ontologia CompGuide e mapear elementos de informação, como as execuções de protocolos clínicos, as tarefas executadas, as observações do estado do paciente, e as restrições temporais, para entidades do modelo relacional, para assim ser possível apresentar toda esta expressividade. Ainda assim, é crucial que a informação sobre os protocolos clínicos, como os dados sobre as tarefas e as restrições temporais das tarefas, seja guardada na base de dados. Esta informação é então armazenada nas tabelas ilustradas na Figura 33, recorrendo-se às classes que se encontram descritas na secção 5.4. A descrição de cada classe em OWL encontra-se no capítulo 3 e a definição dos diagramas de sequência que dizem respeito ao motor de execução, encontram-se na secção 0.

### 5.4 Diagrama de Classes

O diagrama de classes, que foi desenvolvido a partir do modelo de domínio definido na secção 4.1, tem como objetivo a conceção da arquitetura e a sua implementação. Nele, descreve-se a estrutura do sistema, revelando as suas classes, os respetivos atributos e métodos e as relações que as classes estabelecem entre si.

Depois do diagrama de classes estar concebido, e de as classes estarem bem identificadas, torna-se mais fácil perceber de que métodos é que cada classe irá precisar, para que se implementem as funcionalidades do sistema.

As diferentes classes ilustradas nas Figuras 18-20 permitem gerir a informação dos protocolos clínicos na base de dados, nomeadamente as suas execuções, as tarefas concluídas, as observações do estado do paciente, e as restrições temporais. Em última instância esta informação é essencial para que o motor de execução, descrito na secção 0, possa funcionar corretamente e calcular as tarefas que o profissional de saúde deve executar ou finalizar.

De seguida, apresenta-se o diagrama de classes da aplicação, dividido por *packages*. De referir que este diagrama apenas se foca nas classes que representam os aspetos temporais dos protocolos clínicos, pelo que esta abordagem foi adotada para que seja fácil a visualização do diagrama.

Os packages são os seguintes:

- **Package Persistence Gestão de Tarefas**

Relativamente às classes do *package* de Gestão de Tarefas, representadas na Figura 18, correspondem à representação das tabelas da base de dados em objetos Java. O mapeamento das tabelas para estes objetos é garantido através de objetos Java denominados de POJOS ou entidades *Bean*.



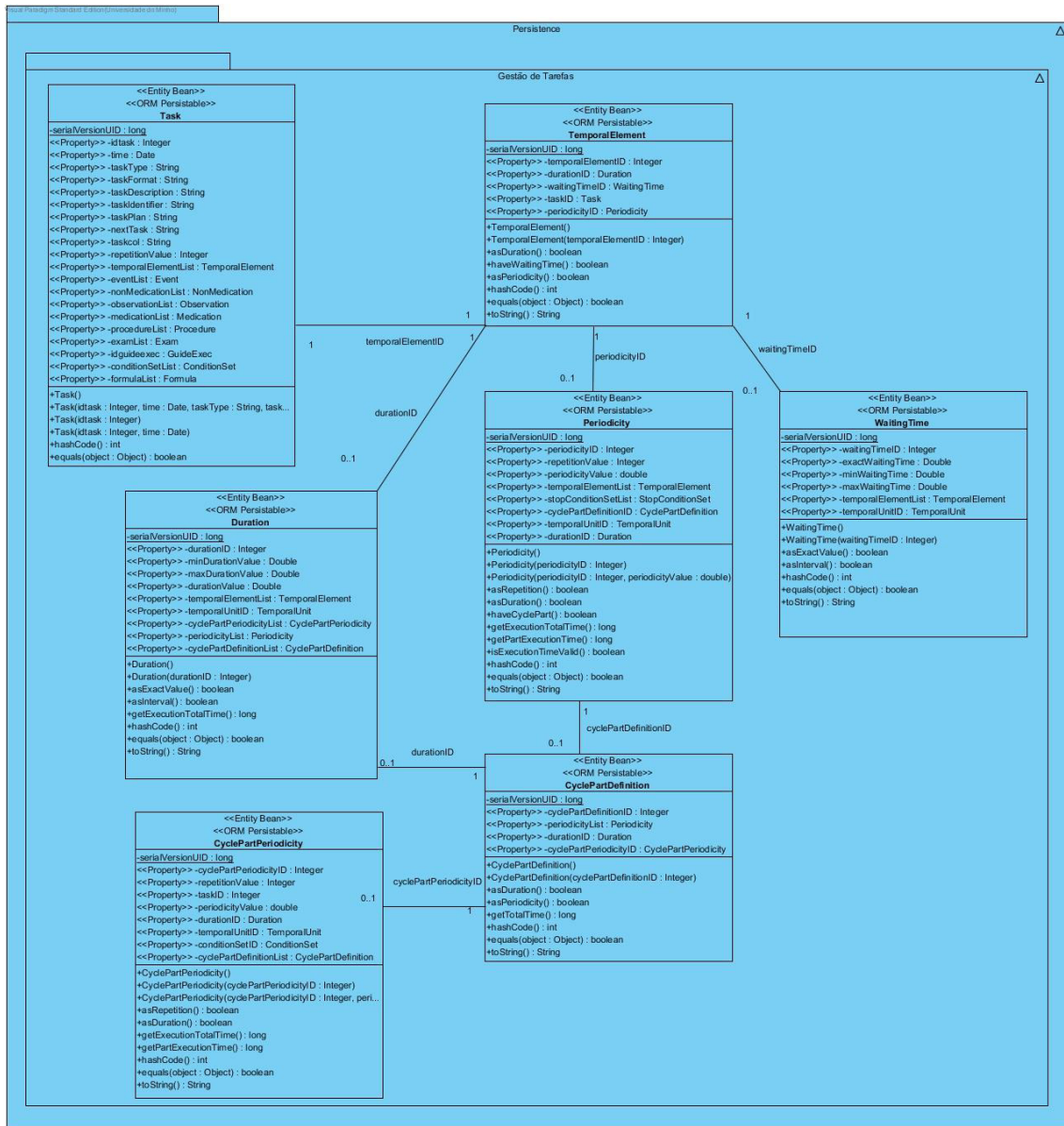


Figura 18 - Diagrama de classes do package Persistence Gestão de Tarefas.

• **Package Controllers Gestão de Tarefas**

As classes deste package, ilustradas na Figura 19, representam a camada de negócio, que tem como objetivo controlar o acesso das *Views* ao *Model* e vice-versa. Este controlo materializa-se através validação e filtragem dos dados, invocação dos dados necessários ao *Model*, bem como passagem destes dados às *Views*.

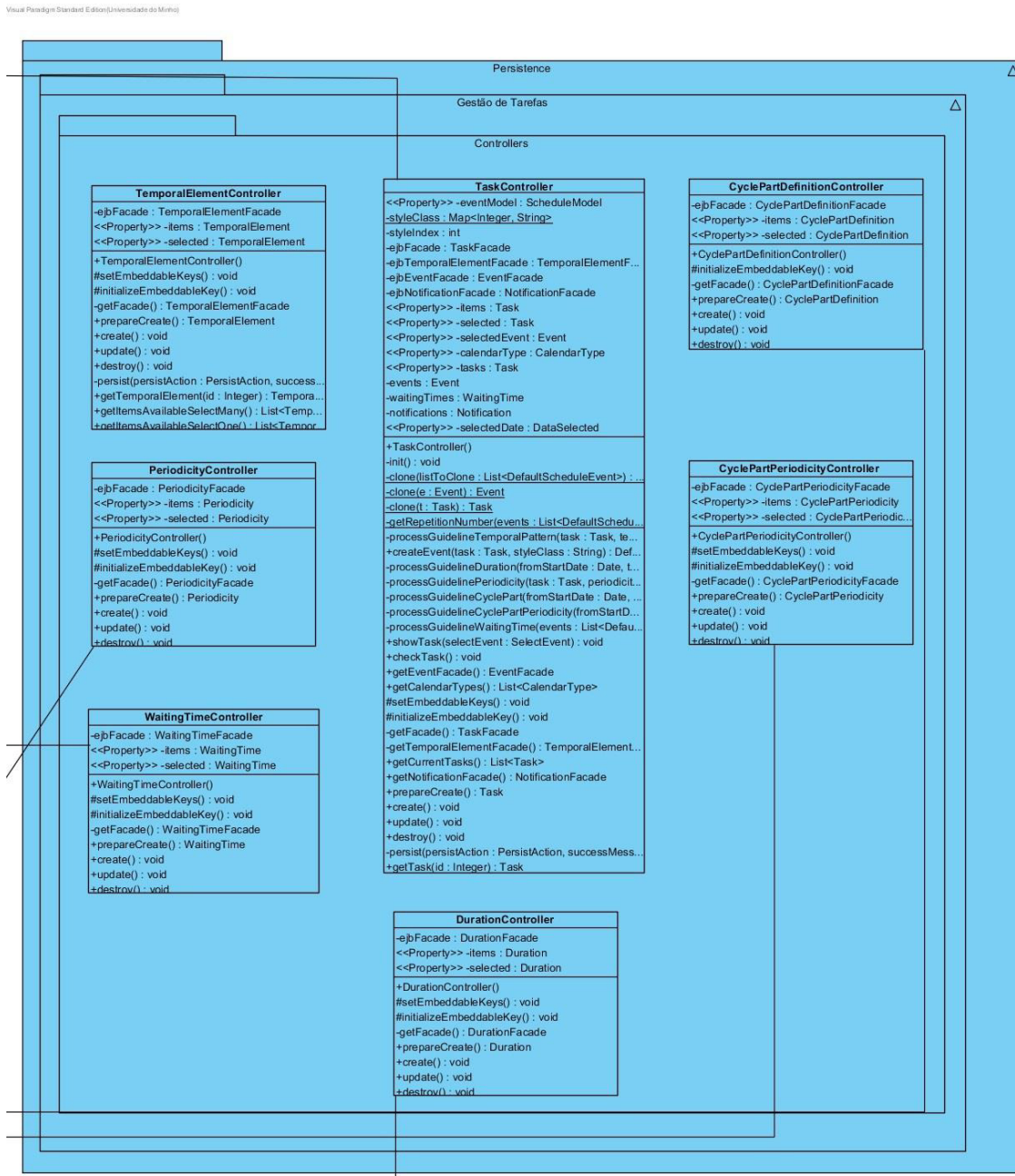


Figura 19 - Diagrama de classes do package Controllers Gestão de Tarefas.

- **Package SessionBeans Gestão de Tarefas**

As classes deste package, apresentadas na Figura 20, representam a camada de acesso a dados que utiliza JPA para assegurar a persistência de dados. Para o mapeamento entre as classes e a base de dados MySQL é utilizado o Hibernate que é uma framework que faz o mapeamento objeto-relacional.

Visual Paradigm Standard Edition (Universidade do Minho)

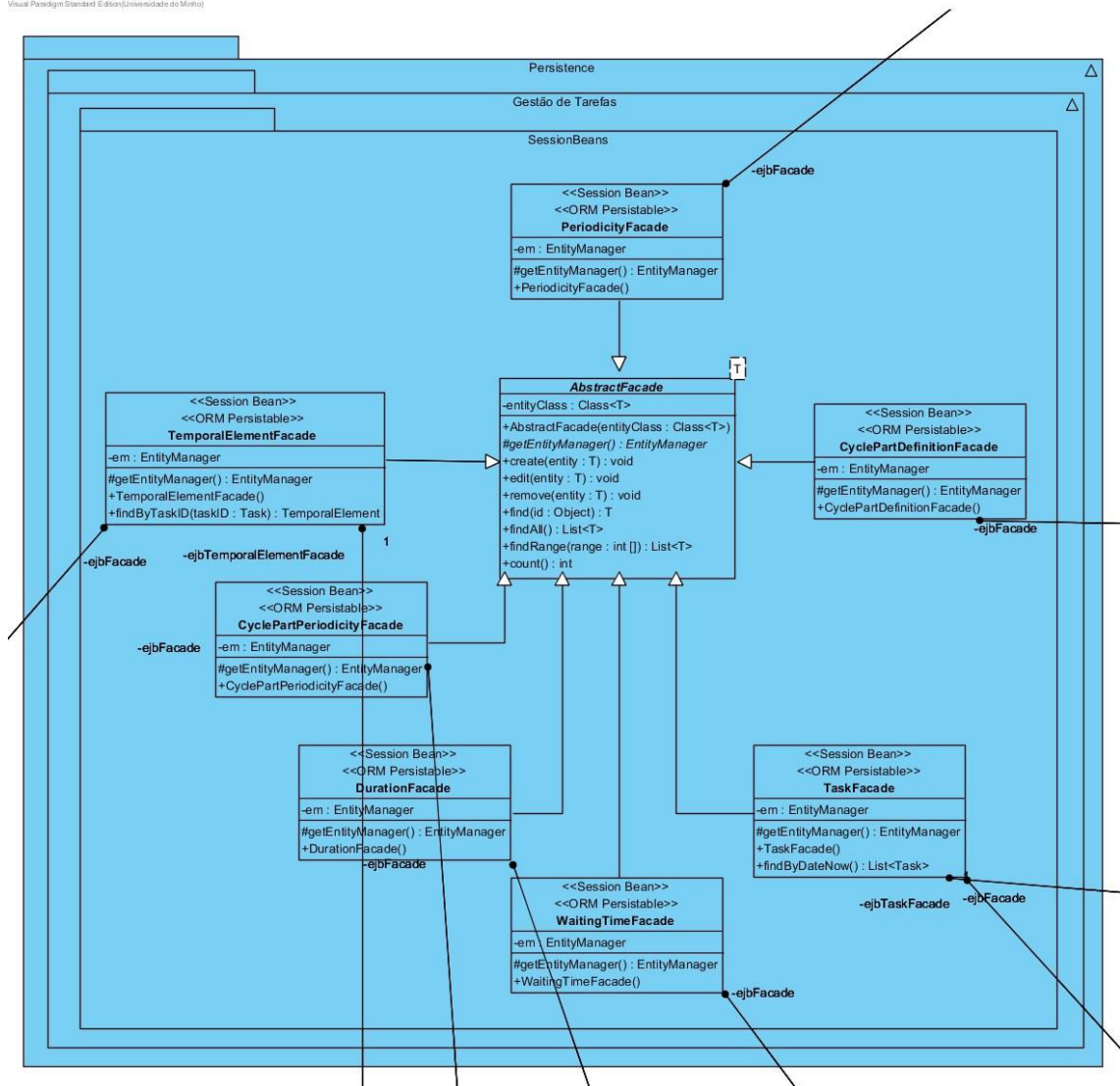


Figura 20 - Diagrama de classes do package SessionBeans Gestão de Tarefas.

## 5.5 Diagramas de Sequência

O diagrama de sequência é um diagrama que descreve a sequência de mensagens trocadas entre objetos, permitindo desta forma descrever a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Estes diagramas são utilizados para representar um cenário para um determinado caso de uso mostrando os eventos que partem do ator e chegam ao sistema.

O motor de execução é a ferramenta que vai permitir interpretar os aspetos temporais dos protocolos clínicos, nas tarefas que os profissionais de saúde têm de efetuar, pelo que os diagramas de sequência ilustrados neste capítulo representam o algoritmo responsável por mapear estes aspetos temporais. Uma vez que a representação no modelo CompGuide, em

OWL, entre entidades, é mais complexa e expressiva do que a sua representação em base de dados, nomeadamente no modelo relacional, houve a necessidade de passar esta complexidade para um algoritmo.

Para que este algoritmo funcione é crucial que o protocolo clínico esteja codificado num ficheiro *owl*, tal como é descrito na secção **Erro! A origem da referência não foi encontrada.** Destaque para os diagramas da Figura 21 e da Figura 22. No Anexo C encontram-se os restantes diagramas.

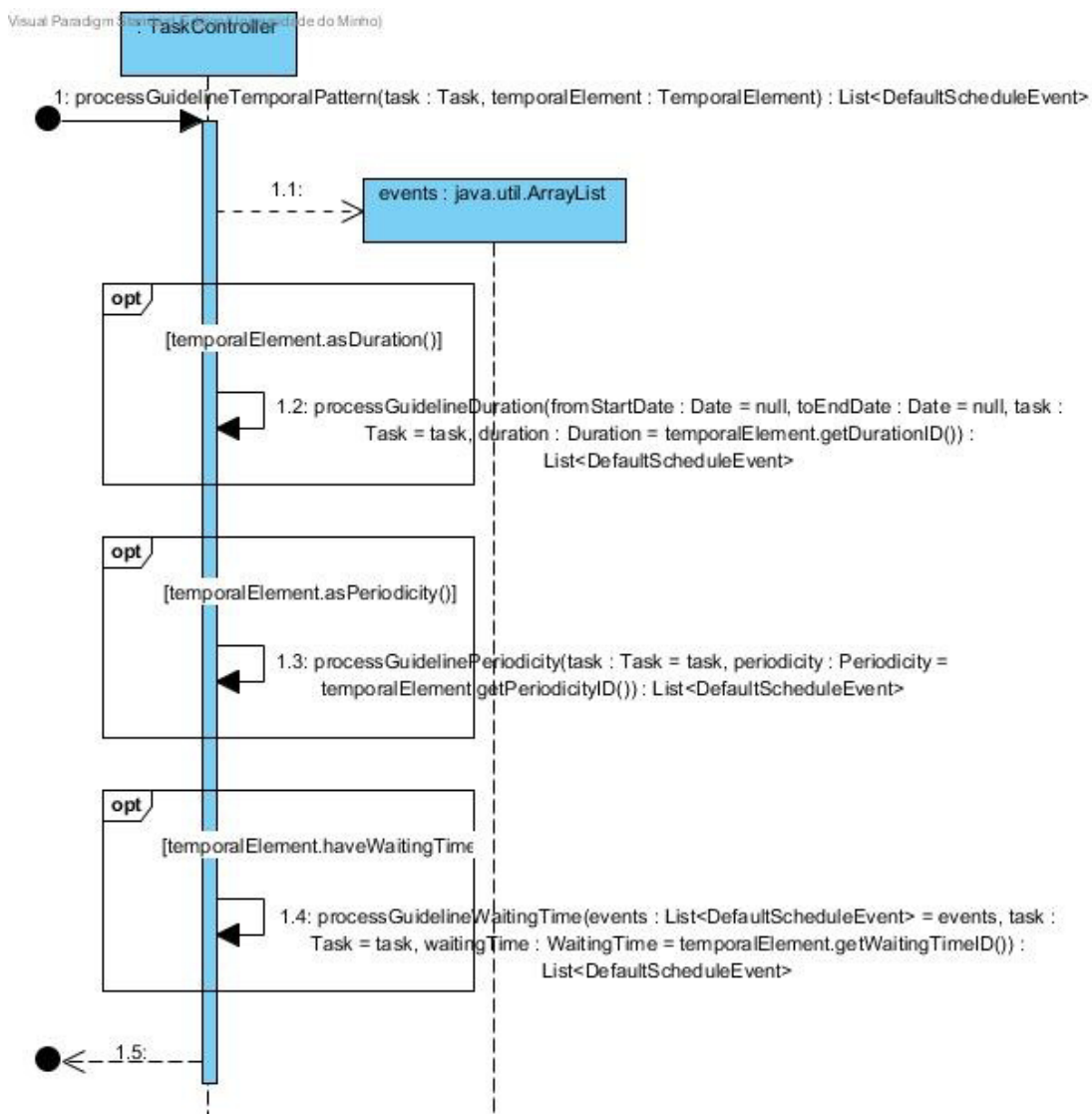


Figura 21 - Diagrama de sequência Processar Aspectos Temporais.

O diagrama de Sequência da Figura 21 representa o algoritmo que procura processar os aspetos temporais das tarefas, existentes nos protocolos clínicos.

Este algoritmo pretende processar todas as restrições temporais dos protocolos clínicos que podem ser durações, periodicidades, tempos de espera e restrições temporais sobre o estado do paciente. Desta forma, para uma determinada tarefa é possível determinar as suas sub-tarefas, o seu tempo de execução e finalização, uma vez que o algoritmo permite calcular estes atributos, tendo em conta os aspetos temporais presentes nos protocolos clínicos.

O processamento do padrão temporal tempo de espera (*WaitingTime*) encontra-se ilustrado no diagrama de sequência da Figura 22. Este algoritmo permite calcular, para todas as tarefas, os seus tempos de espera, retornando a lista de tarefas, tendo em conta esta restrição.

Os restantes algoritmos para o processamento dos aspetos temporais encontram-se no Anexo C.

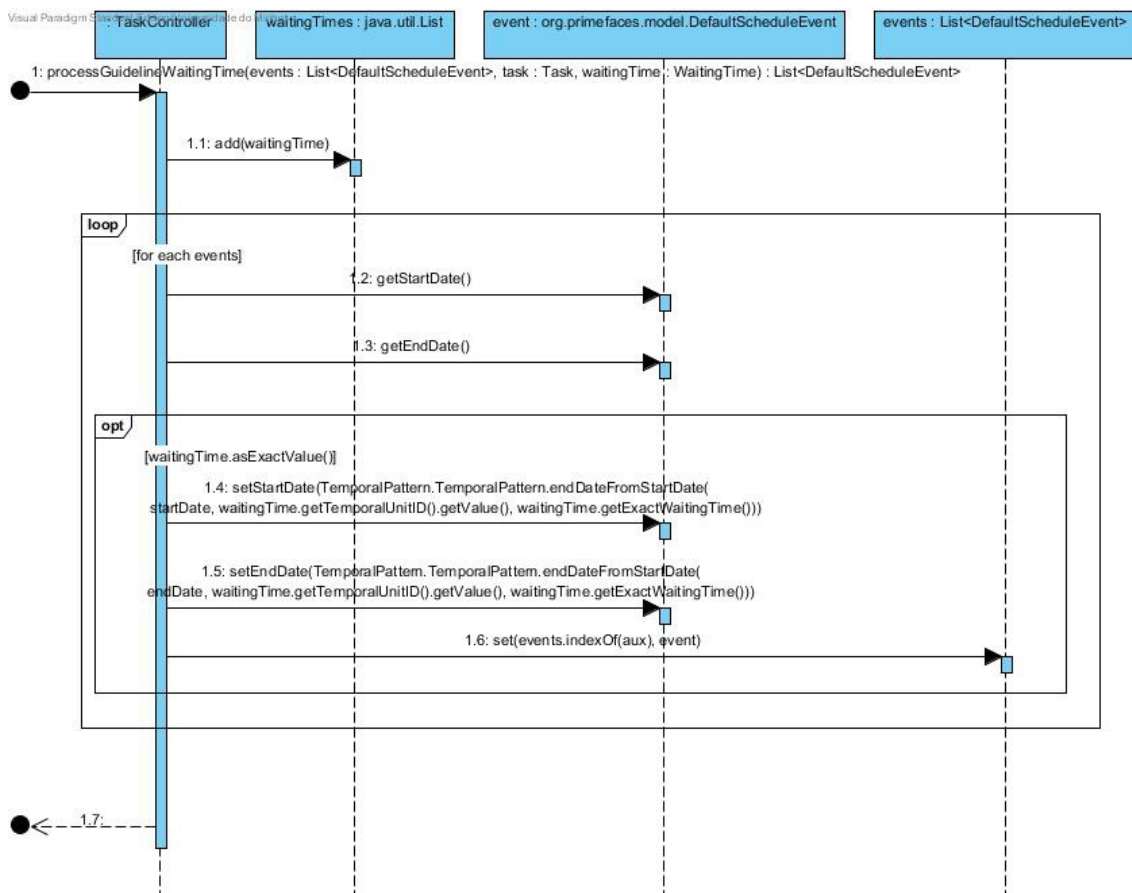


Figura 22 - Diagrama de sequência Processar Tempo de Espera.

## 5.6 Desenvolvimento Baseado em Padrões de Software

O desenvolvimento do *software* foi baseado na utilização de padrões. Um padrão de *design* é um conjunto de boas práticas para solucionar problemas parecidos que acontecem com regularidade. Desta forma, o uso de padrões neste trabalho foi crucial. Alguns dos padrões utilizados são referidos como padrões GoF – Gang of Four – porque foram inicialmente descritos por quatro autores: Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides [64].

Os padrões de *software* utilizados foram:

- Padrão *Factory*. é um padrão GoF, do tipo criacional que assenta no encapsulamento da criação de um objeto. Este padrão permite centralizar num único ponto a criação de objetos de um dado tipo (ou subclasses desse tipo), abstraindo os detalhes da criação de um objeto;
- Padrão *Adapter*. o padrão *adapter* funciona como uma ponte entre duas interfaces incompatíveis. Este tipo de padrão de *software* é um padrão do tipo estrutural uma vez que combina a capacidade de duas interfaces independentes e envolve uma única classe que é responsável por juntar as funcionalidades das interfaces independentes ou incompatíveis;
- Padrão *Singleton*. este padrão GoF do tipo criacional garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto;
- Padrão *Facade*. fornece uma interface unificada de um conjunto de interfaces num subsistema. *Facade* define uma interface de alto nível que torna o subsistema mais simples de usar, escondendo a complexidade do sistema.

Tal como ilustrado na Figura 23, o uso dos padrões *factory*, *adapter* e *singleton* prende-se com o facto de haver necessidade de se criar uma ponte entre as interfaces já existentes no projeto e as novas interfaces criadas. A classe *GuidelineHandler*, já existente no projeto, permite fazer o *parse* do ficheiro *owl* e retornar a informação que diz respeito a um protocolo clínico, como por exemplo, o tipo de tarefa, as restrições temporais, entre outras informações. *TemporalElement* representa a classe de mapeamento destas restrições dos protocolos clínicos na base de dados, pelo que existe a necessidade de criar uma ponte entre estas duas classes.

Assim sendo, por um lado, para se criar a ponte entre as duas interfaces incompatíveis, *GuidelineHandler* e a classe *TemporalElement*, o padrão *adapter* permite juntar as funcionalidades das interfaces independentes ou incompatíveis. Por outro, para se manter um ponto único de criação do objeto da classe *TemporalElementAdapter*, recorre-se ao padrão *singleton*. e para devolver o objeto do tipo *GuidelineInterface* recorre-se ao padrão *factory*.

Nesta perspetiva, facilmente se pode chamar um método de uma classe já existente com as novas funcionalidades integradas no projeto.

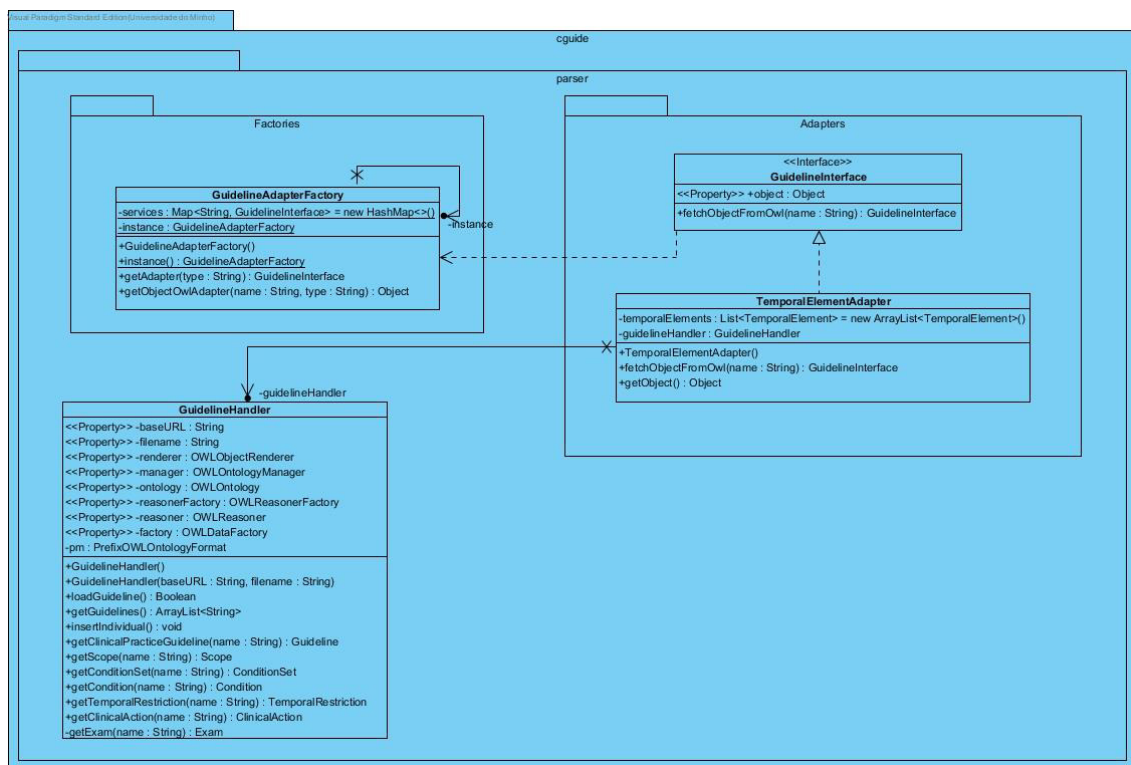


Figura 23 - Diagrama de classes Integração do Padrão Adapter.

A integração do padrão *facade* prende-se com o facto de haver a necessidade de abstrair a complexidade no mapeamento das tabelas da base de dados para objetos em Java *Beans*, tal como apresentado na Figura 24.

De facto, toda esta complexidade encontra-se escondida através da interface *AbstractFacade*, uma classe que fornece uma série de funcionalidades e métodos de negócio comuns a todas as interfaces. Além disso, para cada classe que representa uma determinada tabela da base de dados existe também uma interface que abstrai a complexidade dos métodos de negócio específicos dessa entidade.

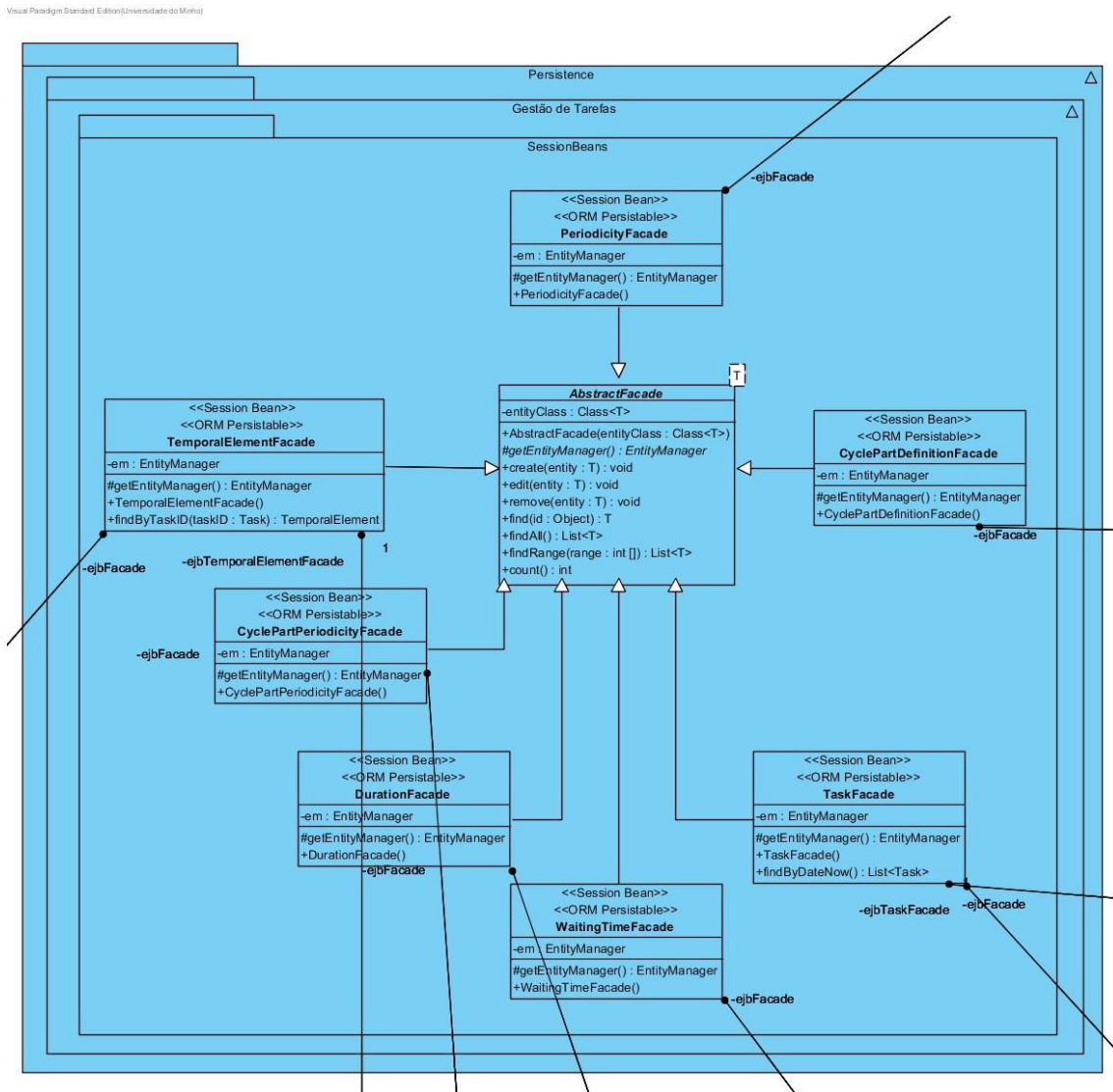


Figura 24 - Diagrama de classes Integração do Padrão Facade.

## 5.7 Apresentação da Aplicação Web

Nesta secção pretende-se apresentar o portal *web* de forma a esclarecer as funcionalidades implementadas.

A página *web* da Figura 25 permite aos utilizadores fazer o login na aplicação. Para tal, é necessário fazer o registo na aplicação ou o administrador adicionar o utilizador.



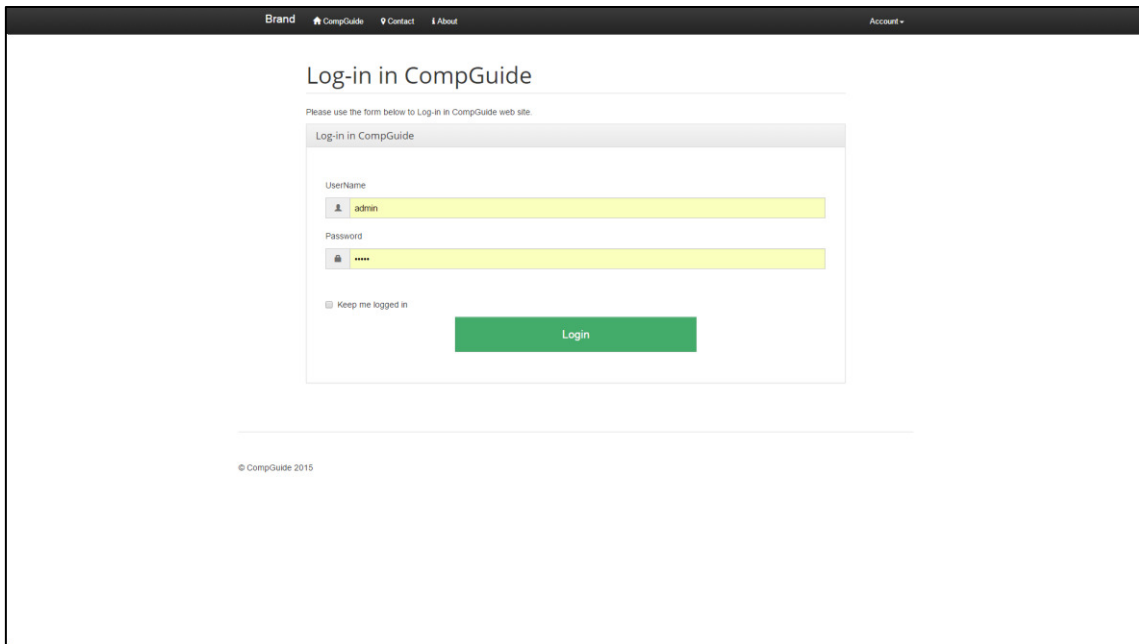


Figura 25 - Página de login.

Na Figura 26, encontra-se ilustrada a página de registo que permite aos seus utilizadores efetuar o registo na aplicação. Para efetuar o registo com sucesso os utilizadores devem facultar a informações solicitadas, sendo que, para segurança do sistema, esta informação é validada sempre que alterado o valor do campo a preencher. Após o preenchimento destes campos e selecionada a opção de registo, o utilizador será reencaminhado para a página de tarefas.

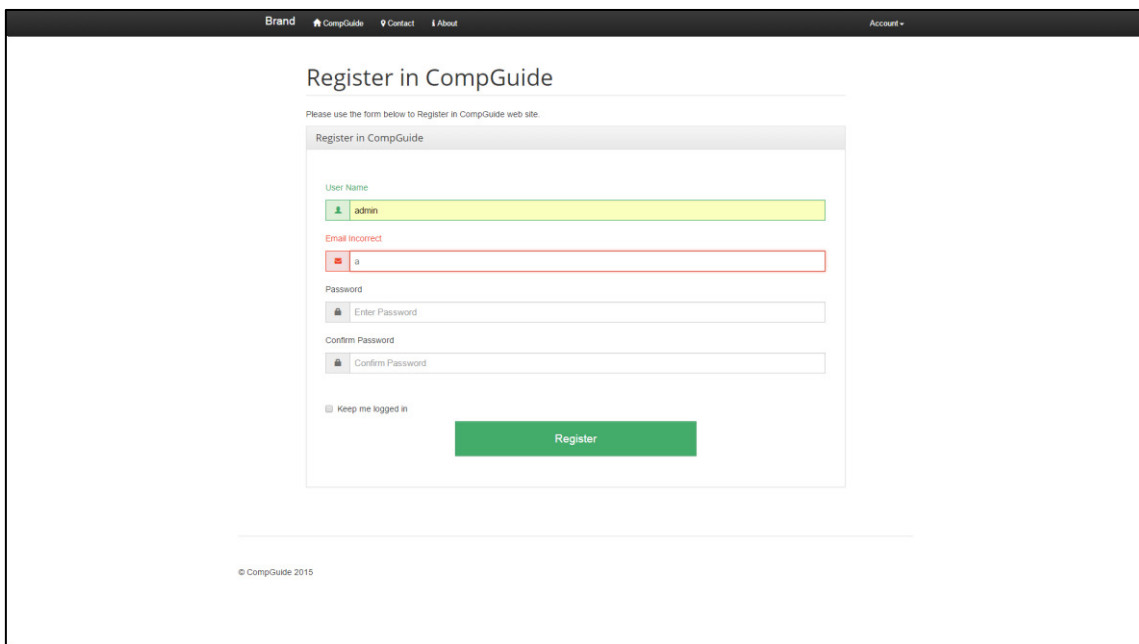


Figura 26 - Página de registo.

Na página *web* da Figura 27, o profissional de saúde poderá ver as tarefas que deve começar e terminar. Poderá visualizá-las sob a forma de um cronograma ou através de um calendário tal como ilustrado na Figura 28.

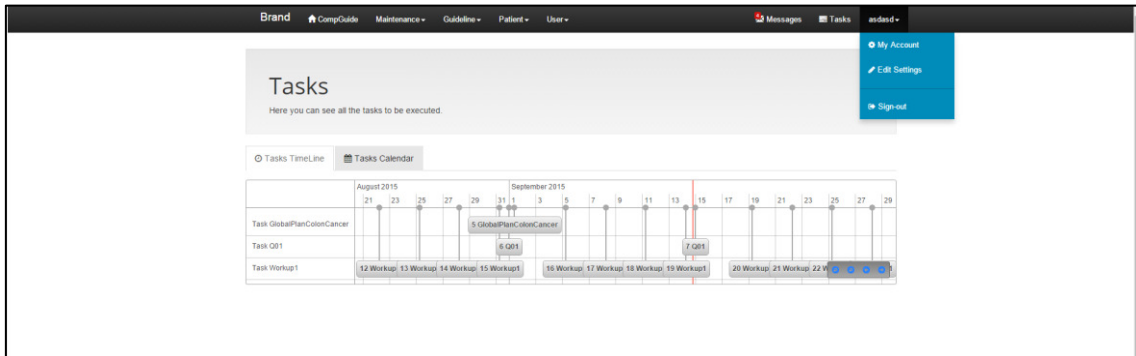


Figura 27 - Página cronograma com as tarefas

Além da visualização temporal das tarefas, o utilizador será notificado quando uma tarefa deve terminar ou começar. A visualização destas mensagens pode ser feita através da caixa de mensagens, representada na Figura 28, bem como através das mensagens instantâneas que alertam o profissional de saúde sobre determinado evento numa tarefa.

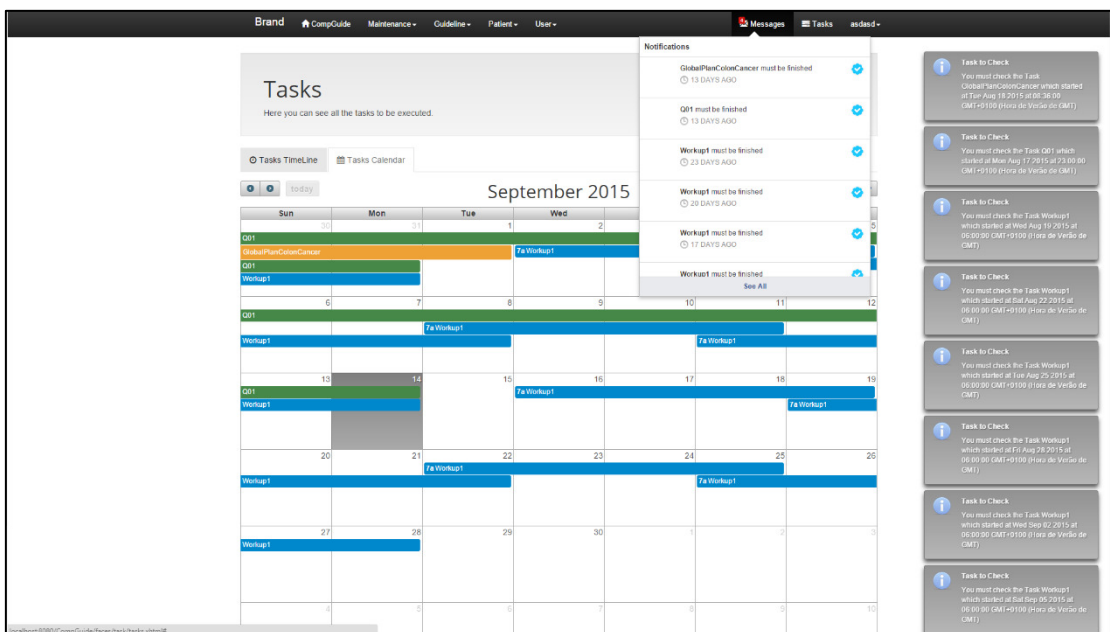


Figura 28 - Página calendário com tarefas.

Ao clicar sobre uma determinada tarefa no *widget* calendário ou no *widget* cronograma, o profissional de saúde poderá visualizar a informação detalhada da tarefa. Ao clicar na tarefa

abrir-se-á uma nova janela, tal como ilustrado na Figura 29, que contém o tempo que falta para terminar ou para começar a tarefa, o identificador e a descrição da tarefa, o número de vezes que ela foi executada, o número de vezes que deve repetir-se e ainda a data de finalização ou início da tarefa.

As restantes páginas *web* são apresentadas no Anexo D.

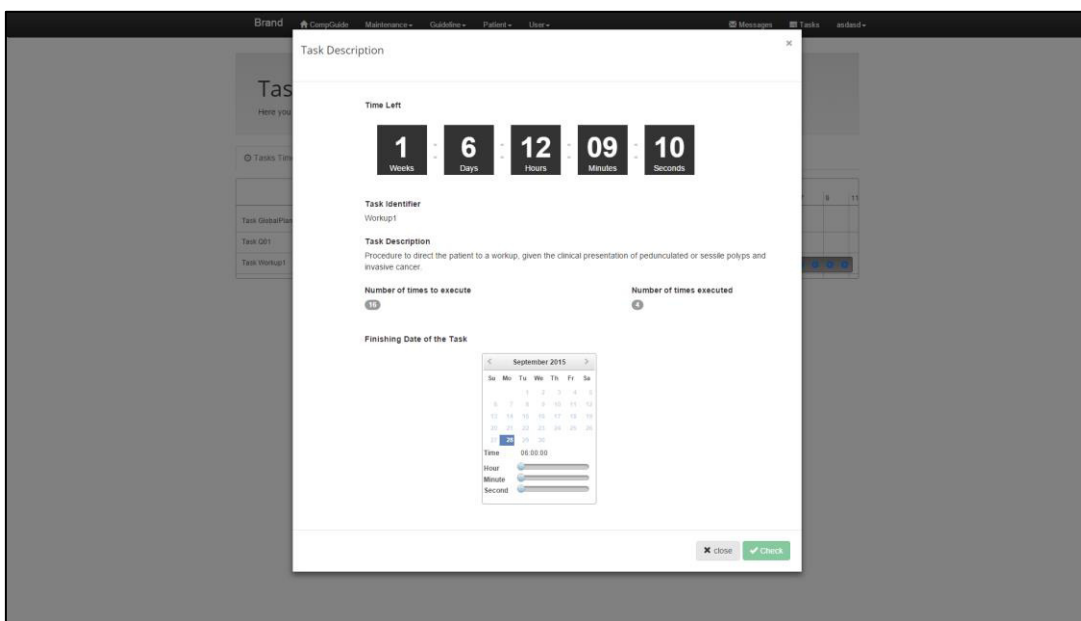


Figura 29 - Página de informação detalhada da tarefa.

## 5.8 Discussão e Análise da Solução

Relativamente às funcionalidades do sistema, este respeita o conjunto de requisitos definidos, pelo que é possível gerir a informação dos protocolos clínicos, gerir a informação dos Profissionais de Saúde e dos seus pacientes. Desta forma, é possível criar, editar, eliminar e atualizar toda esta informação, permitindo ao Administrador do sistema gerir a informação de forma simples e segura.

O motor de execução temporal foi integrado com sucesso na aplicação, e além disso, foi capaz de interpretar todos aspetos temporais do protocolo clínico para tratamento do cancro do colon. Além deste motor, os *widgets* calendário e cronograma mostraram ser ferramentas muito importantes no apoio clínico, consistindo num assistente pessoal que permite aos profissionais de saúde gerir o tempo de execução das tarefas. Para a execução das tarefas nas datas estipuladas, e de acordo com as restrições temporais existentes nos protocolos clínicos, foi

bastante importante a gestão de notificações, pelo que se adotou uma caixa de mensagens e um sistema de notificação cíclico.

Relativamente à caixa de mensagens, é possível visualizar as mensagens com as tarefas que devem ser executadas, permitindo nessa mesma caixa escolher se pretende executar ou não a tarefa, tal como ilustrado na Figura 30.

O sistema de notificações cíclico permite de forma periódica alertar o Profissional de Saúde sobre as tarefas que devem ser executadas, isto é, de 30 em 30 segundos a aplicação mostra um *pop-up* de notificação, tal como ilustrado na Figura 30.

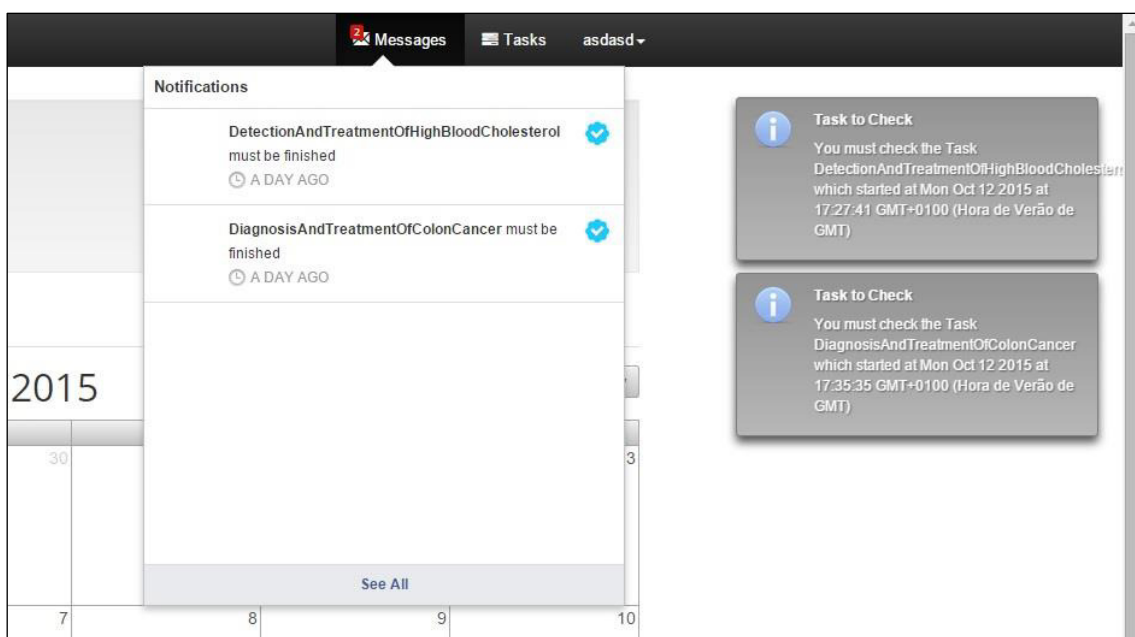


Figura 30 - Caixa de mensagens e pop-ups de notificação.

A partir dos *widets* calendário (Figura 31) e cronograma (Figura 32) é possível visualizar temporalmente as tarefas a serem executadas, ou seja, é possível visualizar a data em que uma determinada tarefa deve ser executada bem como quanto tempo dura uma tarefa. Ainda assim, cada tarefa pode conter vários eventos ou sub-tarefas, uma vez que as restrições temporais dos protocolos clínicos ditam um conjunto de mecanismos de controlo para os construtores temporais, de acordo com a secção 3.2.

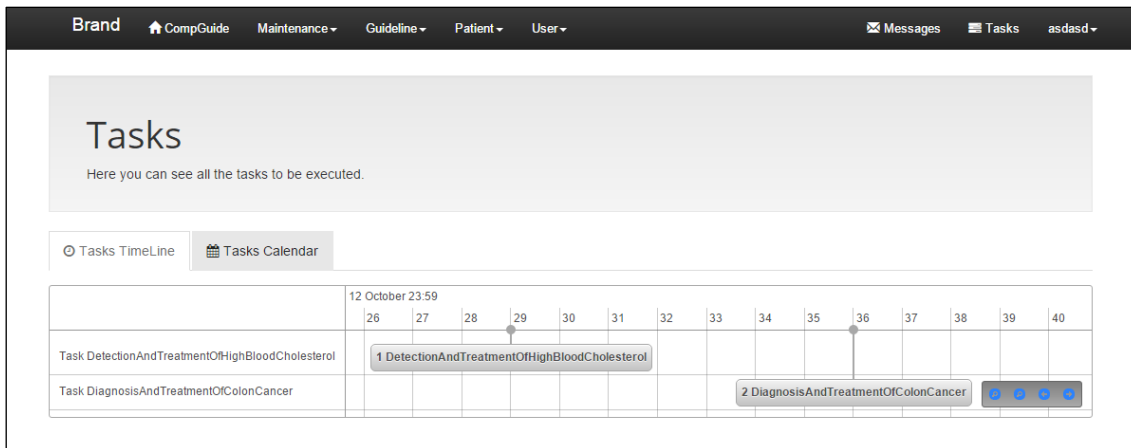


Figura 31 - Cronograma de tarefas.

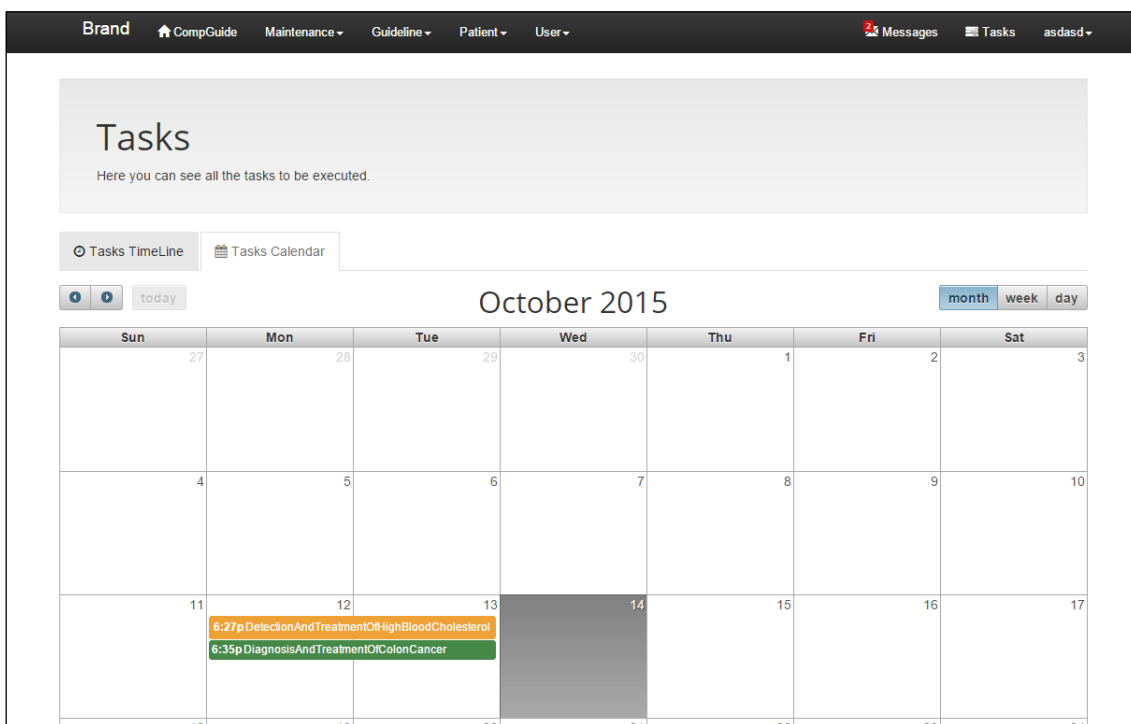


Figura 32 - Calendário de tarefas

Assim sendo, existem vários tipos de tarefas como *Actions*, *Questions*, *Decisions* e *Plans* que possuem vários mecanismos de controlo para os construtores temporais e que, em última análise, definem a disposição temporal das tarefas e sub-tarefas nos *widgets* calendário e no cronograma.

Em *Action* o construtor temporal *repetitionValue* define que a execução total da tarefa só é finalizada quando esta for executada o número de vezes especificado, pelo que o Profissional de Saúde é notificado sobre o número de repetições da tarefa, quantas vezes foi repetida e quantas

repetições faltam. Para o construtor temporal *Duration*, as tarefas do tipo *Action* apenas são finalizadas após a ação de clique no botão *execute* e após o tempo especificado em *durationValue*. Caso o construtor *Duration* esteja definido num intervalo o, Profissional de Saúde é notificado quando se alcançar os tempos mínimo e máximo de duração da tarefa.

Relativamente ao construtor *Periodicity*, é apresentado ao Profissional de Saúde o conjunto de sub-tarefas que corresponde ao número de vezes que a tarefa se repete, como por exemplo, a tarefa *x* repete-se de 2 em 2 meses. O tempo de execução total da tarefa é determinado pela definição dos construtores *Duration* ou de um *repetitionValue*. Assim sendo, é possível determinar quanto tempo dura o evento periódico ou o número de vezes que deve ser executado até terminar a execução do protocolo clínico ou, ainda, passar para a próxima tarefa.

Em cada execução das sub-tarefas, o Profissional de Saúde é informado da restante duração do evento periódico ou do número de vezes que ainda é necessário repeti-lo.

No caso de haver uma *StopCondition* a cada execução do evento, o profissional de Saúde é questionado se a *StopCondition* se verifica. Se assim for, a execução periódica do evento termina e avança-se para a tarefa seguinte. Caso a *stopCondition* não se verifique, a execução periódica do evento prossegue.

Contudo, se houver o construtor temporal *CyclePartDefinition*, é necessário definir um conjunto de sub-tarefas que correspondem ao número de vezes que cada sub-tarefa se repete (caso exista o construtor *repetitionValue*), ou ao tempo que deve durar (caso exista o construtor *Duration*).

Pode ainda existir o construtor *WaitingTime* que restringe o início de uma tarefa quando se confirma um determinado tempo de espera. Assim sendo, uma determinada tarefa apenas pode ser iniciada após se verificar o tempo definido neste construtor.

Um *Plan* apresenta as mesmas restrições da tarefa *Action*. Como tal, os mecanismos de controlo e notificação são os mesmos.

No entanto, há alguns elementos adicionais que são considerados, e que auxiliam o controlo temporal de um protocolo clínico. Assim sendo, o somatório das durações, durações de eventos periódicos e tempos de espera de tarefas que estão dentro de um *Plan* não deve ultrapassar a duração total desse *Plan*. Esta verificação realiza-se quando se adiciona um protocolo clínico à lista de protocolos disponíveis.

A utilização do construtor temporal *TemporalRestriction* permite validar as restrições temporais sobre os parâmetros do estado do paciente.

Desta forma, é possível validar há quanto tempo um determinado paciente evidencia um determinado sintoma, através do operador *within\_the\_last*.

No caso do padrão temporal *within\_the\_following*, este é utilizado apenas para expressar *outcomes* de tarefas. Sempre que uma tarefa apresenta um *outcome* com este padrão temporal, o motor de execução questiona o utilizador se o *outcome* esperado tiver ocorrido após o tempo que foi especificado.

Esta informação é guardada e, posteriormente, utilizada quando for necessário comparar com uma condição. Assim, a avaliação das condições permite então, determinar se uma tarefa pode ser ou não executada.

## 6 Conclusões e Trabalho Futuro

Este capítulo descreve as conclusões finais sobre o trabalho realizado ao longo do projeto, evidencia também os objetivos cumpridos. Serão identificadas as limitações e definido o trabalho futuro, bem como as dificuldades que foram surgindo ao longo do projeto.

### 6.1 Objetivos Realizados

Relativamente aos objetivos, considera-se que foram alcançados satisfatoriamente, tendo em conta que foi elaborada uma análise extensa ao problema, o que por si só, permite encarar a implementação como uma tarefa muito complexa. Foram consignadas várias abordagens ao mesmo tema, de forma a abranger o maior número de hipóteses no que toca à posterior implementação de funcionalidades. Todos os requisitos do projeto foram implementados e, como resultado, a solução final entregue na data de conclusão da dissertação satisfaz os objetivos impostos no seu início.

Relativamente ao primeiro objetivo deste trabalho que consiste na caracterização e identificação dos aspetos temporais na representação de protocolos clínicos, verificou-se que estes se podem dividir em restrições temporais à execução de tarefas e restrições temporais sobre condições do estado do paciente. Este objetivo foi concretizado no capítulo 2.

O segundo objetivo, a identificação das lacunas dos modelos CIG abordados, apurou-se que nenhum destes modelos apresenta uma representação temporal que cubra na totalidade estes dois grupos de restrições anteriormente descritos, identificando-se lacunas, sobretudo, no segundo aspeto. De realçar a comparação dos modelos existentes, realizada no capítulo 2.

O terceiro objetivo, o desenvolvimento de um modelo de representação geral capaz de preencher as lacunas dos modelos existentes, foi concretizado no capítulo 3, com a apresentação de um modelo temporal validado através de um exemplo de um protocolo clínico. Ao nível de durações, tempos de espera, periodicidades e restrições sobre condições do estado do paciente, foi possível representar o protocolo na sua totalidade.

Relativamente ao quarto objetivo deste trabalho de dissertação, a conceção de um motor execução para o modelo, este foi capaz de interpretar de forma satisfatória os aspetos temporais, para o caso do protocolo clínico do cancro do cólon, pelo que se pode afirmar, devido



à complexidade deste protocolo, que o motor de execução abrange a maioria dos aspetos temporais. O desenvolvimento do motor de execução, viável e completo, foi um processo extremamente moroso, durante o qual foi necessário fazer um minucioso estudo dos projetos já existentes, para assim identificar possíveis lacunas e possíveis pontos que poderiam integrar o projeto. Este objetivo foi concretizado no capítulo 5.

Além disso, o quinto objetivo que visa a conceção de uma interface que possibilite a um profissional de saúde visualizar as recomendações clínicas num contexto temporal, optou-se por desenvolver uma aplicação *web* que possibilita visualizar toda a informação referente à ocorrência de eventos temporais e disponibiliza um sistema de notificações e avisos periódicos ao profissional de saúde. Este objetivo foi também concretizado no capítulo 5.

Assim sendo, a proposta de valor deste projeto consiste num modelo de representação de restrições temporais dos protocolos clínicos completo, uma vez que as diferentes abordagens não apresentam alguns dos aspetos temporais tidos como cruciais.

## 6.2 Limitações e Trabalho Futuro

Inicialmente, surgiram algumas dificuldades pelo facto do projeto se basear num conjunto de conceitos complexos e, à data, desconhecidos. O domínio do projeto foi a maior dificuldade uma vez que a área clínica é complexa e de difícil compreensão, pelo que foi necessário fazer um trabalho de pesquisa para perceber não só o domínio do projeto, mas também perceber até que ponto os protocolos clínicos podem afetar a prática clínica.

Percebido o domínio do projeto, facilmente se passou para a elaboração do modelo de representação em OWL. Considera-se que a representação do modelo em OWL é uma mais-valia, uma vez que é possível representar relações complexas através de uma linguagem estruturada para essas relações. Contudo, seria ainda interessante a criação de um *plugin* que permitisse o mapeamento das classes em OWL para tabelas da base de dados, integrando este *plugin* na aplicação. Este ponto traria uma enorme vantagem, permitindo na aplicação em Java facilmente fazer novas alterações, acrescentando novas funcionalidades. É possível através da JPA e do *Hibernate* atualizar as classes que representam as tabelas da base de dados, pelo que a integração deste *plugin* facilitaria a manutenção da aplicação.

### **6.3 Divulgação de Resultados**

Deste trabalho de dissertação resultou a publicação de um artigo científico no *2015 Imperial College Computing Student Workshop*, um *workshop* de referência organizado pelo Departamento de Ciências da Computação do *Imperial College London*, onde todos os artigos aceites passaram por um processo de revisão por pares. A referência do artigo é:

A. Silva, T. Oliveira, P. Novais, and J. Neves, “Representing Temporal Patterns in Computer-Interpretable Clinical Guidelines,” in *2015 Imperial College Computing Student Workshop (ICCSW15)*, 2015, pp. 62–69.

## Referências

- [1] V. Della Mea, "What is e-health (2): The death of telemedicine?," *J. Med. Internet Res.*, vol. 3, no. 2, pp. 6–7, 2001.
- [2] G. Eysenbach and T. L. Diepgen, "The role of e-health and consumer health informatics for evidence-based patient choice in the 21st century," *Clin. Dermatol.*, vol. 19, no. 1, pp. 11–17, 2001.
- [3] J.-C. J. Healy, "Implementing e-health in developing countries: Guidance and principles," *ICT Appl. Cybersecurity Div. CYB, ...*, no. September, pp. 1–53, 2008.
- [4] E. Rich, K. Knight, and M. C. S. R. Ratto, *Inteligência artificial*, 2nd ed. Makron Books, 1988.
- [5] G. F. Luger, *Inteligência Artificial - Estruturas e estratégias para a solução de problemas complexos*, 4th ed. Bookman, 2004.
- [6] E. S. Berner and T. J. L. Lande, *Clinical Decision Support Systems: Theory and Practice*. Springer Science & Business Media, 2007.
- [7] M. A. Musen, Y. Shahar, and E. H. Shortliffe, "Clinical decision-support systems," in *Biomedical Informatics Computer Applications in Health Care and Biomedicine*, E. Shortliffe and J. Cimino, Eds. Springer, 2006, pp. 698–736.
- [8] E. Coiera, *Guide to health informatics*, 3rd ed. CRC Press, 2015.
- [9] R. S. Ledley and L. B. Lusted, "Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason.," *Science*, vol. 130, no. 3366, pp. 9–21, 1959.
- [10] L. Perreault and J. Metzger, "A pragmatic framework for understanding clinical decision support.," *J. Healthc. Inf. Manag.*, vol. 13, no. 2, pp. 5–21, 1999.
- [11] G. Kong, D. Xu, J. Yang, and M. Business, "Clinical Decision Support Systems: A Review on Knowledge Representation and Inference Under Uncertainties," *Int. J. Comput. Intell. Syst.*, vol. 1, no. 2, pp. 159–167, 2008.
- [12] S. Vachhrajani, A. V. Kulkarni, and J. R. W. Kestle, "Clinical practice guidelines," *J. Neurosurg. Pediatr.*, vol. 3, no. 4, pp. 249–256, 2009.
- [13] M. Samwald, K. Fehre, J. de Bruin, and K. P. Adlassnig, "The Arden Syntax standard for clinical decision support: Experiences and directions," *J. Biomed. Inform.*, vol. 45, no. 4, pp. 711–718, 2012.
- [14] M. Peleg, a a Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E. H. Shortliffe, and R. a Greenes, "GLIF3: the evolution of a guideline representation format.," in *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, 2000, pp. 645–649.
- [15] M. Balsler, C. Duelli, and W. Reif, "Formal semantics of Asbru-an overview," in *Proceedings of the International Conference on Integrated Design and Process Technology*, 2002.

- [16] E. Vier, J. Fox, N. Johns, C. Lyons, A. Rahmzadeh, and P. Wilson, "PROforma: systems," *Comput. Methods Programs Biomed.*, vol. 2607, no. 97, 1997.
- [17] P. Ram, D. Berg, S. Tu, G. Mansfield, Q. Ye, R. M. Abarbanel, and N. Beard, "Executing clinical practice guidelines using the SAGE execution engine.," *Stud. Health Technol. Inform.*, vol. 107, no. Pt 1, pp. 251–5, 2004.
- [18] T. Oliveira, P. Leão, P. Novais, and J. Neves, "Webifying the Computerized Execution of Clinical Practice Guidelines," *Trends Pract. Appl. Heterog. Multi-Agent Syst. PAAMS Collect. SE - 18*, vol. 293, pp. 149–156, 2014.
- [19] T. Oliveira, P. Novais, and J. Neves, "Representation of clinical practice Guideline components in OWL," *Adv. Intell. Syst. Comput.*, vol. 221, pp. 77–85, 2013.
- [20] B. Somekh, *Action Research : a Methodology for Change and Development.* ., 2005.
- [21] K. Schwaber and M. Beedle, *Agile Software Development with Scrum.* Prentice Hall, 2002.
- [22] P. A. de Clercq, J. A. Blom, H. H. M. Korsten, and A. Hasman, "Approaches for creating computer-interpretable guidelines that facilitate decision support.," *Artif. Intell. Med.*, vol. 31, no. 1, pp. 1–27, 2004.
- [23] K. Kaiser and S. Miksch, "Versioning computer-interpretable guidelines: Semi-automatic modeling of 'Living Guidelines' using an information extraction method," *Artif. Intell. Med.*, vol. 46, no. 1, pp. 55–66, 2009.
- [24] M. N. Hadley, "Clinical practice guidelines.," *J. Neurosurg. Pediatr.*, vol. 3, no. 4, pp. 247–8; author reply 248, 2009.
- [25] S. H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw, "Potential benefits, limitations, and harms of clinical guidelines," *Bmj*, vol. 318, no. 7182, pp. 527–530, 1999.
- [26] A. Latoszek-Berendsen, H. Tange, H. J. van den Herik, and A. Hasman, "From Clinical Practice Guidelines to Computer-interpretable Guidelines," *Methods Inf. Med.*, vol. 49, no. 6, pp. 550–570, 2010.
- [27] C. S. Chim, N. T. Cheung, H. Fung, and K. C. Wong, "Electronic clinical practice guidelines: current status and future prospects in Hong Kong," *Hong Kong Med J*, vol. 9, pp. 299–301, 2003.
- [28] M. Peleg, S. Keren, and Y. Denekamp, "Mapping computerized clinical guidelines to electronic medical records: Knowledge-data ontological mapper (KDOM)," *J. Biomed. Inform.*, vol. 41, no. 1, pp. 180–201, 2008.
- [29] R. M. Gardner, T. A. Pryor, and H. R. Warner, "The HELP hospital information system: update 1998," *Int. J. Med. Inform.*, vol. 54, no. 3, pp. 169–182, 1999.
- [30] M. A. Musen, S. W. Tu, A. K. Das, and Y. Shahar, "EON: A Component-Based Approach to Automation of Protocol-Directed Therapy," *Emerg. Infect. Dis.*, vol. 3, no. 6, pp. 367–388, 1996.
- [31] A. Bottrighi, P. Terenziani, S. Montani, M. Torchio, and G. Molino, "Clinical guidelines contextualization in GLARE.," in *AMIA ... Annual Symposium proceedings / AMIA*

*Symposium. AMIA Symposium*, 2006, vol. 2006, p. 860.

- [32] P. D. Clayton, T. A. Pryor, O. B. Wigertz, and G. Hripcsak, "Issues and Structures for Sharing Medical Knowledge among Decision-Making Systems: The 1989 Arden Homestead Retreat," in *Annual Symposium on Computer Application in Medical Care*, 1989, pp. 116–121.
- [33] G. Hripcsak, P. Ludemann, T. A. Pryor, O. B. Wigertz, and P. D. Clayton, "Rationale for the Arden Syntax," *Comput. Biomed. Res.*, vol. 27, no. 4, pp. 291–324, 1994.
- [34] "HL7 Standards - Master Grid," 2015. [Online]. Available: [http://www.hl7.org/implement/standards/product\\_matrix.cfm?Family=V2](http://www.hl7.org/implement/standards/product_matrix.cfm?Family=V2). [Accessed: 15-Jan-2015].
- [35] G. Hripcsak, "Writing Arden Syntax medical logic modules," *Comput. Biol. Med.*, vol. 24, no. 5, pp. 331–363, 1994.
- [36] L. Ohno-Machado, J. H. Gennari, S. N. Murphy, N. L. Jain, S. W. Tu, D. E. Oliver, E. Pattison-Gordon, R. a Greenes, E. H. Shortliffe, and G. O. Barnett, "The guideline interchange format: a model for representing guidelines.," *J. Am. Med. Inform. Assoc.*, vol. 5, no. 4, pp. 357–372, 1998.
- [37] P. E. Stoufflet, L. Ohno-Machado, S. R. A. Deibel, D. Lee, and R. A. Greenes, "GEODE-CM: a state-transition framework for clinical management," in *AMIA Annual Symposium Proceedings*, 1996, vol. 9, no. 4, p. 924.
- [38] M. Barnes and G. O. Barnett, "An architecture for a distributed guideline server.," in *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, 1995, vol. 19, pp. 233–7.
- [39] M. Peleg, a a Boxwala, E. Bernstam, S. Tu, R. a Greenes, and E. H. Shortliffe, "Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax.," *J. Biomed. Inform.*, vol. 34, no. 3, pp. 170–181, 2001.
- [40] S. Miksch, Y. Shahar, and P. Johnson, "Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans," *Proc. 7th Work. Knowl. Eng. Methods Lang.*, pp. 1–25, 1997.
- [41] Y. Shahar, S. Miksch, and P. Johnson, "The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines," *Artif. Intell. Med.*, vol. 14, no. 1–2, pp. 29–51, 1998.
- [42] Y. Shahar, O. Young, and E. Shalom, "DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines," in *Artificial Intelligence in ...*, Springer, 2003, pp. 122–131.
- [43] S. W. Tu, J. R. Campbell, J. Glasgow, M. a. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, J. G. Mansfield, M. a. Musen, and R. M. Abarbanel, "The SAGE Guideline Model: Achievements and Overview," *J. Am. Med. Informatics Assoc.*, vol. 14, no. 5, pp. 589–598, 2007.
- [44] J. Fox, N. Johns, and A. Rahmanzadeh, "Disseminating medical knowledge : the PRO forma approach," *Lect. Notes Artif. Intell.*, vol. 14, no. 1, pp. 157–182, 1997.

- [45] A. Vollebregt, A. ten Teije, F. van Harmelen, J. van Der Lei, and M. Mosseveld, "A study of PROforma, a development methodology for clinical procedures," *Artif. Intell. Med.*, vol. 17, no. 2, pp. 195–221, 1999.
- [46] P. Terenziani, S. Montani, A. Bottrighi, G. Molino, and M. Torchio, "Applying Artificial Intelligence to clinical guidelines: The GLARE approach," in *Studies in Health Technology and Informatics*, vol. 139, Springer, 2008, pp. 273–282.
- [47] K. P. Adlassnig, C. Combi, A. K. Das, E. T. Keravnou, and G. Pozzi, "Temporal representation and reasoning in medicine: Research directions and challenges," *Artif. Intell. Med.*, vol. 38, no. 2, pp. 101–113, 2006.
- [48] Y. Shahar and M. a Musen, *Knowledge-based temporal abstraction in clinical domains.*, vol. 8, no. 3. 1996.
- [49] M. Van der Heijden and P. J. F. Lucas, "Describing disease processes using a probabilistic logic of qualitative time," *Artif. Intell. Med.*, vol. 59, no. 3, pp. 143–155, 2013.
- [50] M. Ben-Ari, A. Pnueli, and Z. Manna, "The temporal logic of branching time," *Acta Inform.*, vol. 20, no. 3, pp. 207–226, 1983.
- [51] L. Anselma, P. Terenziani, S. Montani, and A. Bottrighi, "Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines," *Artif. Intell. Med.*, vol. 38, no. 2, pp. 171–195, 2006.
- [52] A. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. A. Greenes, and E. H. Shortliffe, "GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines.," *J. Biomed. Inform.*, vol. 37, no. 3, pp. 147–61, 2004.
- [53] Y. Shahar, S. Miksch, and P. Johnson, "A task-specific ontology for design and execution of time-oriented skeletal plans," in *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, KAW.*, 1996, vol. 96, pp. 1–20.
- [54] D. Wang, M. Peleg, S. W. Tu, E. H. Shortliffe, and R. a. Greenes, "Representation of clinical practice guidelines for computer-based implementations," *Stud. Health Technol. Inform.*, vol. 84, pp. 285–289, 2001.
- [55] P. Terenziani, C. Carlini, S. Montani, P. Orientale, A. Avogadro, and C. Borsalino, "Towards a Comprehensive Treatment of Temporal Constraints in Clinical Guidelines," in *Temporal Representation and Reasoning, 2002. TIME 2002. Proceedings. Ninth International Symposium on*, 2002, pp. 20–27.
- [56] E. H. Sherman, G. Hripcsak, J. Starren, R. a Jenders, and P. Clayton, "Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge.," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 1995, pp. 238–242.
- [57] R. Dechter, I. Meiri, and J. Pearl, "Temporal constraint networks," *Artif. Intell.*, vol. 49, no. 1–3, pp. 61–95, 1991.
- [58] A. Benson, T. Bekaii-Saab, E. Chan, Y.-J. Chen, M. Choti, H. Cooper, and P. Engstrom, "NCCN Clinical Practice Guideline in Oncology Colon Cancer," 2013.

- [59] A. Seyfang, S. Miksch, and M. Marcos, "Combining diagnosis and treatment using ASBRU.," *Int. J. Med. Inform.*, vol. 68, no. 1–3, pp. 49–57, 2002.
- [60] J. Robertson and S. Robertson, *Volere*, 16th ed. Atlantic Systems Guild Limited, 2000.
- [61] I. Jacobson, G. Booch, and J. Rumbaugh, *The unified software development process*. Addison-Wesley Longman Ltd, 1999.
- [62] D. Distante, P. Pedone, G. Rossi, and G. Canfora, "Model-driven development of Web applications with UWA, MVC and JavaServer faces," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4607, pp. 457–472, 2007.
- [63] "The NetBeans E-commerce Tutorial - Designing the Application Preparing Mockups Use-Case," 2010. [Online]. Available: <https://netbeans.org/kb/docs/javaee/ecommerce/design.html>. [Accessed: 25-Mar-2015].
- [64] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1994.

## **Anexos**



## A. Descrição dos Casos de Uso

O presente anexo fornece uma descrição dos diferentes casos de uso do sistema de execução de protocolos clínicos na forma das Tabelas Tabela 7 -Tabela 24.

Tabela 7 - Descrição caso de uso Pesquisar Paciente.

<b>Nome</b>	<b>Pesquisar Paciente</b>
<b>Propósito</b>	Permitir pesquisar um Paciente que esteja registrado no sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Pesquisar Paciente”
<b>OUT</b>	2. O Sistema disponibiliza um campo para introdução do nome do Paciente.
<b>INP</b>	3. O Administrador introduz o nome do Paciente.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>OUT</b>	5. O Sistema apresenta a lista de Pacientes que correspondem o valor introduzido.
<b>UD</b>	6. O Administrador seleciona o Paciente
<b>OUT</b>	7. O Sistema fornece as várias opções de gestão do Paciente.
<b>ALT</b>	<b>Alternativa 4a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são inválidos.

	2. Volta a 2.
<b>ALT</b>	<b>Alternativa 5a [Pacientes não encontradas]:</b>
	1. Sistema informa que não foram encontrados Pacientes.
	2. Volta a 2.

Tabela 8 - Descrição caso de uso Adicionar Paciente.

<b>Nome</b>	<b>Adicionar Paciente</b>
<b>Propósito</b>	Permitir adicionar um Paciente no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador adiciona o Paciente.
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Adicionar Paciente”
<b>OUT</b>	2. O Sistema disponibiliza um formulário para introdução dos dados do Paciente.
<b>INP</b>	3. O Administrador introduz os dados.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>SR</b>	5. O Sistema regista o Paciente na aplicação.
<b>ALT</b>	<b>Alternativa 3a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são inválidos.
	2. Volta a 2.

<b>EXEC</b>	<b>Exceção 5ª [Erro ao gravar os dados]</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados
	2. Volta a 2.

Tabela 9 - Descrição caso de uso Criar Protocolos clínicos

<b>Nome</b>	<b>Criar Protocolos Clínicos</b>
<b>Propósito</b>	Permitir adicionar um protocolo clínico no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador adiciona a Protocolos clínicos.
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Criar Protocolos clínicos”
<b>OUT</b>	2. O Sistema disponibiliza um formulário para introdução dos dados do protocolo clínico.
<b>INP</b>	3. O Administrador introduz os dados.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>SR</b>	5. O Sistema regista a protocolos clínicos na aplicação.
<b>ALT</b>	<b>Alternativa 3a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são inválidos.
	2. Volta a 2.
<b>ALT</b>	<b>Alternativa 5a [Protocolos clínicos não encontradas]:</b>

<b>OUT</b>	1. Sistema informa que não foram encontrados protocolos clínicos.
	2. Volta a 2.
<b>EXEC</b>	<b>Exceção 5ª [Erro ao gravar os dados]:</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados
	2. Volta a 2.

Tabela 10 - Descrição caso de uso Adicionar Profissional de Saúde

<b>Nome</b>	<b>Adicionar Profissional de Saúde</b>
<b>Propósito</b>	Permitir adicionar um Profissional de Saúde no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador adiciona o Profissional de Saúde.
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Adicionar Profissional de Saúde”
<b>OUT</b>	2. O Sistema disponibiliza um formulário para introdução dos dados do Profissional de Saúde.
<b>INP</b>	3. O Administrador introduz os dados.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>SR</b>	5. O Sistema regista o Profissional de Saúde na aplicação.
<b>ALT</b>	<b>Alternativa 3a [Dados inválidos]:</b>
<b>OUT</b>	1. Sistema informa que os dados introduzidos são

	inválidos.
	2. Volta a 2.
<b>ALT</b>	<b>Alternativa 5a [Profissionais de Saúde não encontrados]:</b>
<b>OUT</b>	1. Sistema informa que não foram encontrados Profissionais de Saúde.
	2. Volta a 2.
<b>EXEC</b>	<b>Exceção 5ª [Erro ao gravar os dados]:</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados
	2. Volta a 2.

Tabela 11 - Descrição caso de uso Editar Profissional de Saúde.

<b>Nome</b>	<b>Editar Profissional de Saúde</b>
<b>Propósito</b>	Permitir adicionar um Profissional de Saúde no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação. O Administrador ter feito o caso de uso “Pesquisar Profissional de Saúde”.
<b>Pós-Condição</b>	O Administrador edita a informação do Profissional de Saúde.
<b>Super Use Case</b>	Pesquisar Profissional de Saúde
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Editar Profissional de Saúde”.

<b>OUT</b>	2. O Sistema fornece um formulário com os dados do profissional de saúde.
<b>INP</b>	3. O Administrador altera os dados e clica em guardar.
<b>IVAL</b>	4. O Sistema valida os dados alterados.
<b>SR</b>	5. O Sistema guarda os dados alterados.
<b>ALT</b>	<b>Alternativa 4a [Dados alterados inválidos]:</b>
<b>OUT</b>	1. O Sistema informa que os dados alterados são inválidos.
	2. Volta a 5.
<b>EXEC</b>	<b>Exceção 5a [Erro ao gravar os dados]:</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados
	2. Volta a 2.

Tabela 12 - Descrição do caso de uso Editar Protocolos clínicos.

<b>Nome</b>	<b>Editar Protocolos clínicos</b>
<b>Propósito</b>	Permitir editar um +rotocolos clínicos no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação. O Administrador ter feito o caso de uso “Pesquisar Protocolos Clínicos”.
<b>Pós-Condição</b>	O Administrador edita a informação da Protocolos clínicos.
<b>Super Use Case</b>	Pesquisar Protocolos clínicos
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Editar

	Protocolos clínicos”
<b>OUT</b>	2. O Sistema fornece um formulário com os dados da Protocolos clínicos.
<b>INP</b>	3. O Administrados altera os dados e clica em guardar.
<b>IVAL</b>	4. O Sistema valida os dados alterados.
<b>SR</b>	5. O Sistema guarda os dados alterados.
<b>ALT</b>	<b>Alternativa 4a [Dados alterados inválidos]:</b>
<b>OUT</b>	1. O Sistema informa que os dados alterados são inválidos.
	2. Volta a 5.
<b>EXEC</b>	<b>Exceção 5a [Erro a guardar os dados]:</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados.
	2. Volta a 2.

Tabela 13 - Descrição do caso de uso Editar Paciente.

<b>Nome</b>	<b>Editar Paciente</b>
<b>Propósito</b>	Permitir adicionar um Paciente no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação. O Administrador ter feito o caso de uso Pesquisar Paciente.
<b>Pós-Condição</b>	O Administrador edita a informação do Paciente.
<b>Super Use Case</b>	Pesquisar Paciente
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Editar

	Paciente”
<b>OUT</b>	2. O Sistema fornece um formulário com os dados da Paciente.
<b>INP</b>	3. O Administrados altera os dados e clica em guardar.
<b>IVAL</b>	4. O Sistema valida os dados alterados.
<b>SR</b>	5. O Sistema guarda os dados alterados.
<b>ALT</b>	<b>Alternativa 4a [Dados alterados inválidos]:</b>
<b>OUT</b>	1. O Sistema informa que os dados alterados são inválidos.
	2. Volta a 5.
<b>EXEC</b>	<b>Exceção 5a [Erro a guardar os dados]:</b>
<b>OUT</b>	1. O Sistema informa que os dados não foram gravados
	2. Volta a 2.

Tabela 14 - Descrição do caso de uso Listar Profissionais de Saúde.

<b>Nome</b>	<b>Listar Profissionais de Saúde</b>
<b>Propósito</b>	Permitir listar os Profissionais de Saúde do Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Listar Profissionais de Saúde”



<b>OUT</b>	2. O Sistema apresenta a lista de profissionais de saúde.
------------	---

Tabela 15 - Descrição do caso de uso Listar Protocolos clínicos.

<b>Nome</b>	<b>Listar Protocolos clínicos</b>
<b>Propósito</b>	Permitir listar os protocolos clínicos do Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Listar Protocolos clínicos”
<b>OUT</b>	2. O Sistema apresenta a lista de protocolos clínicos.

Tabela 16 - Descrição do caso de uso Listar Pacientes.

<b>Nome</b>	<b>Listar Pacientes</b>
<b>Propósito</b>	Permitir pesquisar um Paciente que esteja registrado no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Listar Pacientes”
<b>OUT</b>	2. O Sistema apresenta a lista de Pacientes.

Tabela 17 - Descrição do caso de uso Remover Profissional de Saúde.

<b>Nome</b>	<b>Remover Profissional de Saúde</b>
<b>Propósito</b>	Permitir remover um profissional de saúde que esteja registrado no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação. O Administrador ter feito o caso de uso Pesquisar Profissional de Saúde.
<b>Pós-Condição</b>	O Administrador remove o profissional de saúde.
<b>Super Use Case</b>	Pesquisar Profissional de Saúde
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
	<b>UD</b> 1. O Administrador seleciona a opção “Remover Profissional de Saúde”
	<b>SR</b> 2. O Sistema remove o profissional de saúde.

Tabela 18 - Descrição do caso de uso Remover Protocolos clínicos.

<b>Nome</b>	<b>Remover Protocolos clínicos</b>
<b>Propósito</b>	Permitir remover um Protocolos clínicos que esteja registrado no Sistema. O Administrador ter feito o caso de uso “Pesquisar Protocolos clínicos”.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador remove a Protocolos clínicos.
<b>Super Use Case</b>	Pesquisar Protocolos clínicos
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
	<b>UD</b> 1. O Administrador seleciona a opção “Remover

Protocolos clínicos”	
<b>SR</b>	2. O Sistema remove o Profissional de Saúde.

Tabela 19 - Descrição do caso de uso Remover Paciente.

<b>Nome</b>	<b>Remover Paciente</b>
<b>Propósito</b>	Permitir remover um Paciente que esteja registado no Sistema. O Administrador ter feito o caso de uso Pesquisar Paciente.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador remove o Paciente.
<b>Super Use Case</b>	Pesquisar Paciente.
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Pesquisar Paciente”
<b>SR</b>	2. O Sistema remove o Paciente.

Tabela 20 - Descrição do caso de uso Visualizar Tarefas.

<b>Nome</b>	<b>Visualizar Tarefas</b>
<b>Propósito</b>	Permitir visualizar a lista de tarefas.
<b>Pré-Condição</b>	O Profissional de Saúde ter feito login na aplicação.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Visualizar Tarefas”

<b>OUT</b>	2. O Sistema apresenta a lista de Tarefas.
<b>&lt;&lt;extend&gt;&gt;</b>	3. <<extend>> Gerir Informação Tarefa.

Tabela 21 - Descrição do caso de uso Visualizar Informação Tarefa.

<b>Nome</b>	<b>Gerir Informação Tarefa</b>
<b>Propósito</b>	Permitir gerir a informação de uma tarefa.
<b>Pré-Condição</b>	O Profissional de Saúde ter feito login na aplicação. O Profissional de Saúde ter efetuado o caso de uso Visualizar Tarefas.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a Tarefa.
<b>OUT</b>	2. O Sistema apresenta a opção de Gerir Informação Tarefa.
<b>UD</b>	3. O Administrador seleciona a opção “Gerir Informação Tarefa”
<b>OUT</b>	4. O Sistema fornece as várias opções de gestão das Tarefas.

Tabela 22 - Descrição do caso de uso Verificar Tarefa.

<b>Nome</b>	<b>Verificar Tarefa</b>
<b>Propósito</b>	Permitir confirmar a execução de uma tarefa.
<b>Pré-Condição</b>	O Profissional de Saúde ter feito login na aplicação. O

	Profissional de Saúde ter efetuado o caso de uso “Visualizar Tarefas”.
<b>Pós-Condição</b>	Nenhuma
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Verificar Tarefa”.
<b>OUT</b>	2. O Sistema apresenta o formulário com a informação da Tarefa.
<b>UD</b>	3. O Profissional de Saúde seleciona a opção de “Verificar Tarefa”
<b>IVAL</b>	4. O Sistema calcula se a Tarefa pode ser verificada.
<b>SR</b>	5. O Sistema guarda a informação da Tarefa
<b>ALT</b>	<b>Alternativa 3a [Cancelar]:</b>
<b>UD</b>	1. O Profissional de Saúde seleciona a opção “Cancelar”
	2. Volta a 2.
<b>EXEC</b>	<b>Exceção 4a [A Tarefa não pode ser verificada]:</b>
<b>OUT</b>	1. O Sistema informa que a Tarefa não pode ser verificada

Tabela 23 - Descrição do caso de uso Editar Tarefa

<b>Nome</b>	<b>Editar Tarefa</b>
<b>Propósito</b>	Permitir editar a informação de uma Tarefa no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação. O Administrador ter feito o caso de uso Gerir Informação

	Tarefa.
<b>Pós-Condição</b>	O Administrador edita a informação da Tarefa.
<b>Super Use Case</b>	Gerir Informação Tarefa
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	6. O Administrador seleciona a opção “Editar Tarefa”
<b>OUT</b>	7. O Sistema fornece um formulário com os dados da Tarefa.
<b>INP</b>	8. O Administrados altera os dados e clica em guardar.
<b>IVAL</b>	9. O Sistema valida os dados alterados.
<b>SR</b>	10. O Sistema guarda os dados alterados.
<b>ALT</b>	<b>Alternativa 4a [Dados alterados inválidos]:</b>
<b>OUT</b>	3. O Sistema informa que os dados alterados são inválidos.
	4. Volta a 5.
<b>EXEC</b>	<b>Exceção 5a [Erro a guardar os dados]:</b>
<b>OUT</b>	3. O Sistema informa que os dados não foram gravados
	4. Volta a 2.

Tabela 24 - Descrição do caso de uso Criar Tarefa

<b>Nome</b>	<b>Criar Tarefa</b>
<b>Propósito</b>	Permitir adicionar uma Tarefa no Sistema.
<b>Pré-Condição</b>	O Administrador ter feito login na aplicação.
<b>Pós-Condição</b>	O Administrador adiciona a Tarefa.
<b>Super Use Case</b>	Nenhum
<b>Fluxo de Eventos</b>	<b>Comportamento Normal</b>
<b>UD</b>	1. O Administrador seleciona a opção “Criar Tarefa”
<b>OUT</b>	2. O Sistema disponibiliza um formulário para introdução dos dados do Tarefa.
<b>INP</b>	3. O Administrador introduz os dados.
<b>IVAL</b>	4. O Sistema valida os dados introduzidos.
<b>SR</b>	5. O Sistema regista o Tarefa na aplicação.
<b>ALT</b>	<b>Alternativa 3a [Dados inválidos]:</b>
<b>OUT</b>	3. Sistema informa que os dados introduzidos são inválidos.
	4. Volta a 2.
<b>EXEC</b>	<b>Exceção 5ª [Erro ao gravar os dados]:</b>
<b>OUT</b>	3. O Sistema informa que os dados não foram gravados
	4. Volta a 2.

## B. Modelo Relacional

O presente anexo fornece o modelo relacional da aplicação na Figura 33.

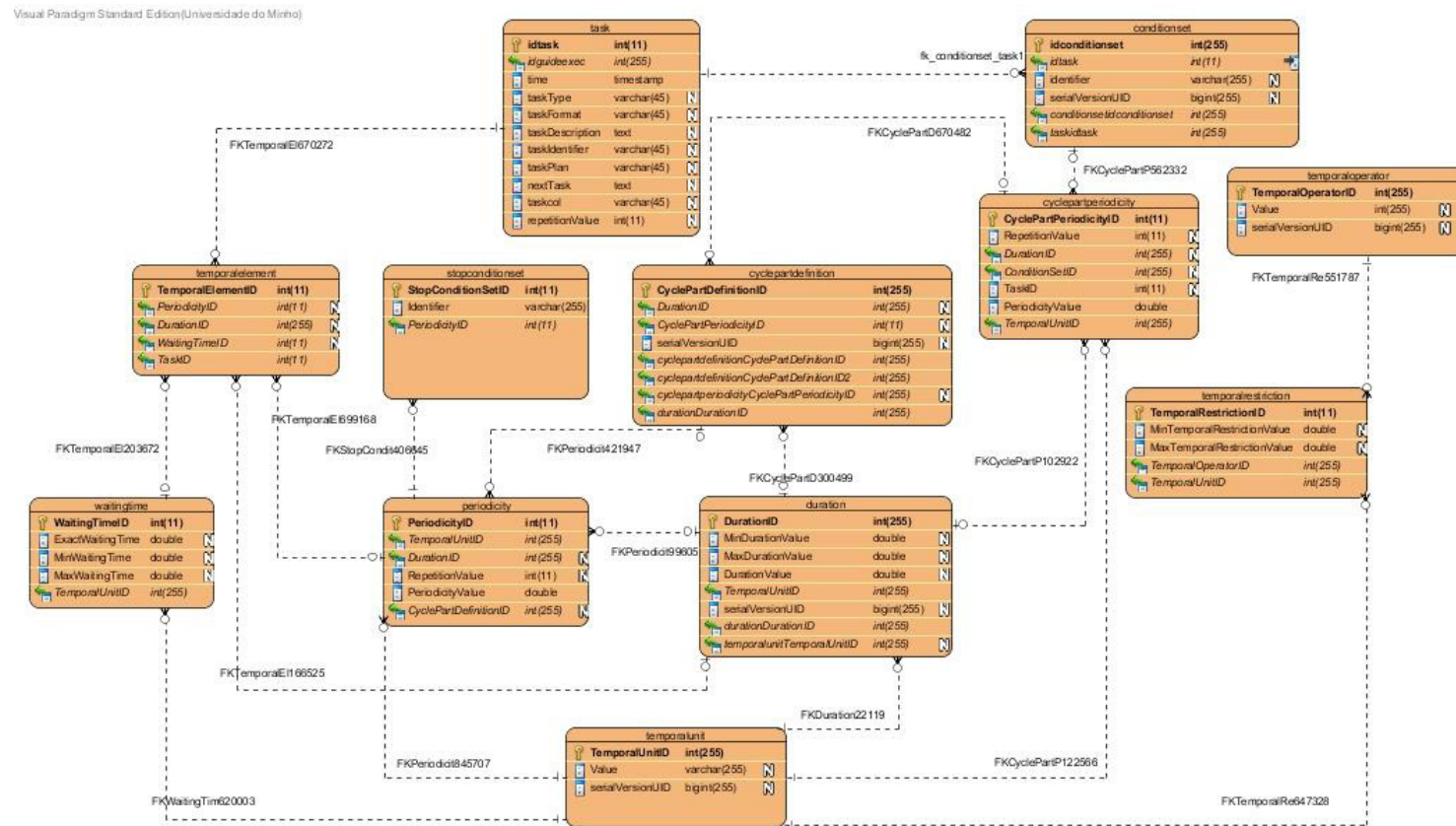


Figura 33 - Modelo relacional.



### C. Diagramas de Sequência

O presente anexo trata dos diagramas de sequência da aplicação.

O Diagrama de Sequência da Figura 34 representa o algoritmo que procura processar a duração das tarefas, existentes nos protocolos clínicos. Assim sendo, este algoritmo permite determinar quanto tempo deve durar uma tarefa, bem como calcular a data em que a tarefa deve começar e terminar.

Caso a classe *Duration* contenha um conjunto de intervalos, o algoritmo retorna uma lista de sub-tarefas com as datas de início e finalização, dentro desse intervalo.

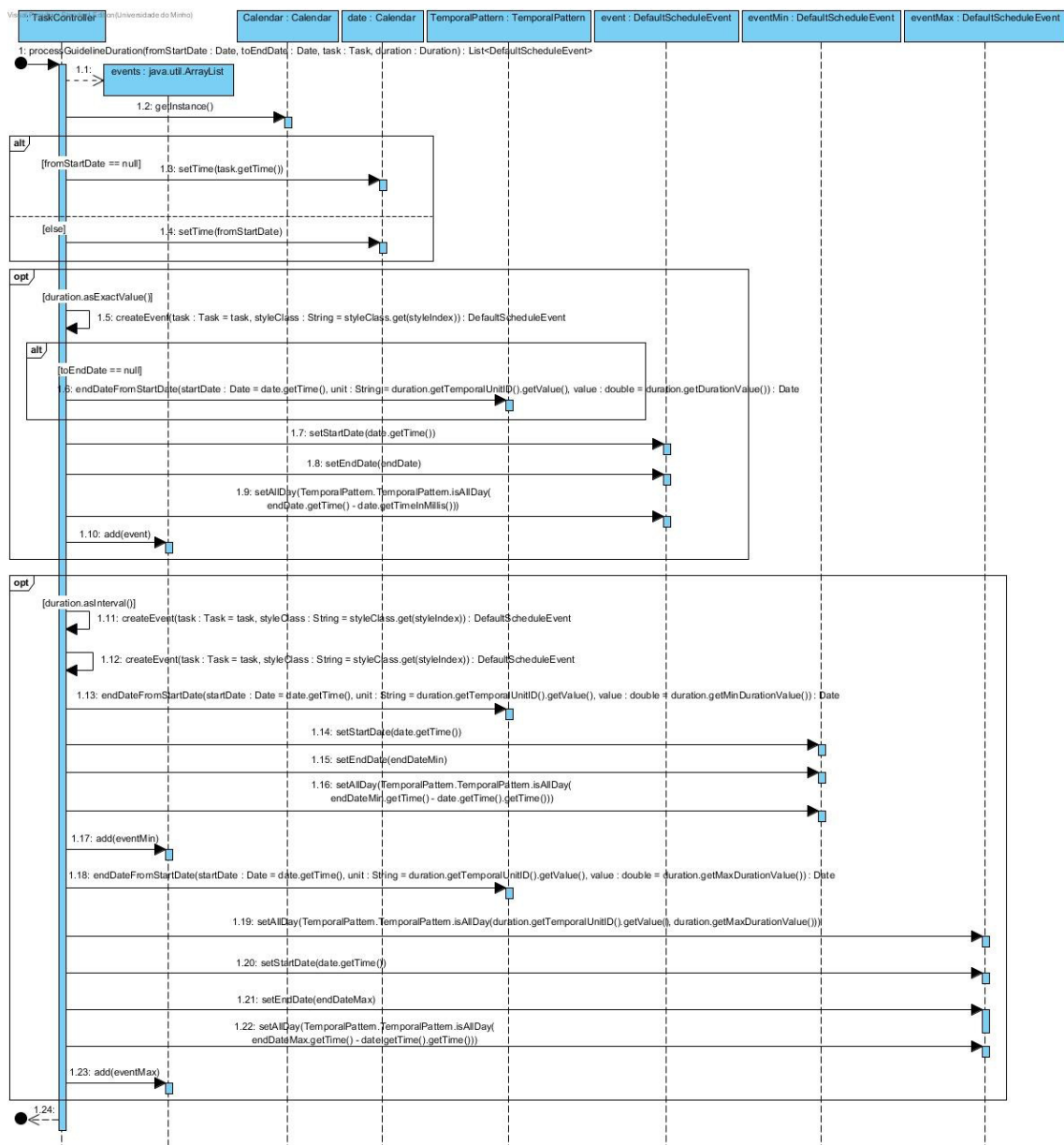


Figura 34 - Diagrama de sequência Processar Durações.

O diagrama de sequência da Figura 35 representa o algoritmo que procura processar a periodicidade das tarefas, existentes nos protocolos clínicos. Este algoritmo, é responsável por determinar o conjunto de sub-tarefas que dizem respeito à tarefa periódica.

Estas sub-tarefas são calculadas através dos atributos *repetitionValue* e *cyclePartDefinition*, uma vez que expressam quantas vezes a tarefa se deve repetir bem como especificam a duração ou periodicidade com que a sub-tarefa deve ocorrer em cada parte de um ciclo.

UML Paradigm Standard Edition (Universidade do Minho)

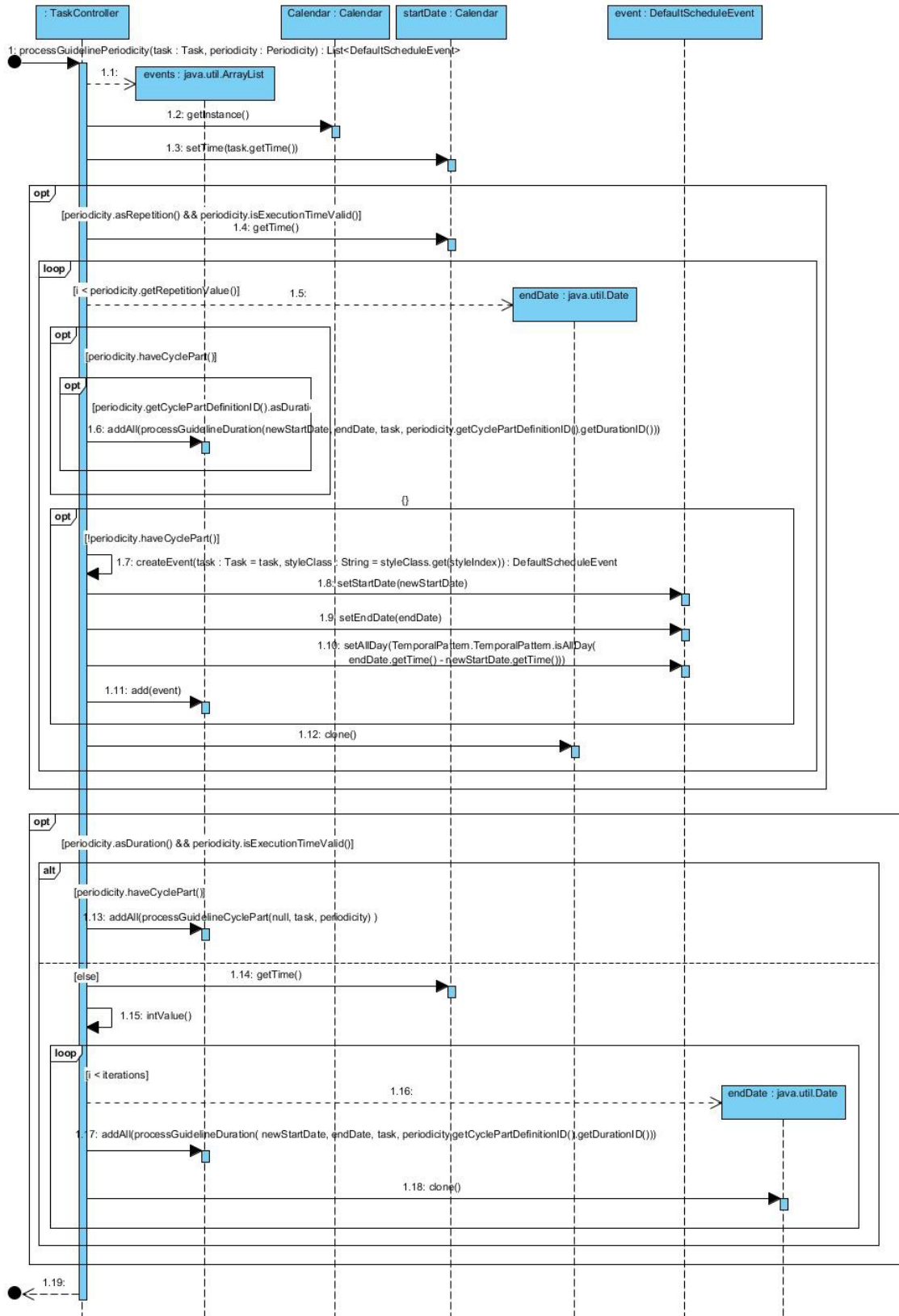


Figura 35 - Diagrama de sequência Processar Periodicidades.

O diagrama de sequência da Figura 36 representa o algoritmo que procura processar a parte de cada ciclo das tarefas, existentes nos protocolos clínicos. Cada instância da classe *CyclePartDefinition*, apenas pode conter um objeto da classe *Duration* ou da classe *CyclePartPeriodicity*, sendo então possível definir a duração ou a periodicidade de cada sub-tarefa. O algoritmo que diz respeito a este diagrama calcula a duração do conjunto de sub-tarefas, tendo em conta os atributos definidos no objeto da classe *Duration*, ou a periodicidade dessas sub-tarefas, de acordo com os atributos definidos no objeto da classe *CyclePartPeriodicity*.

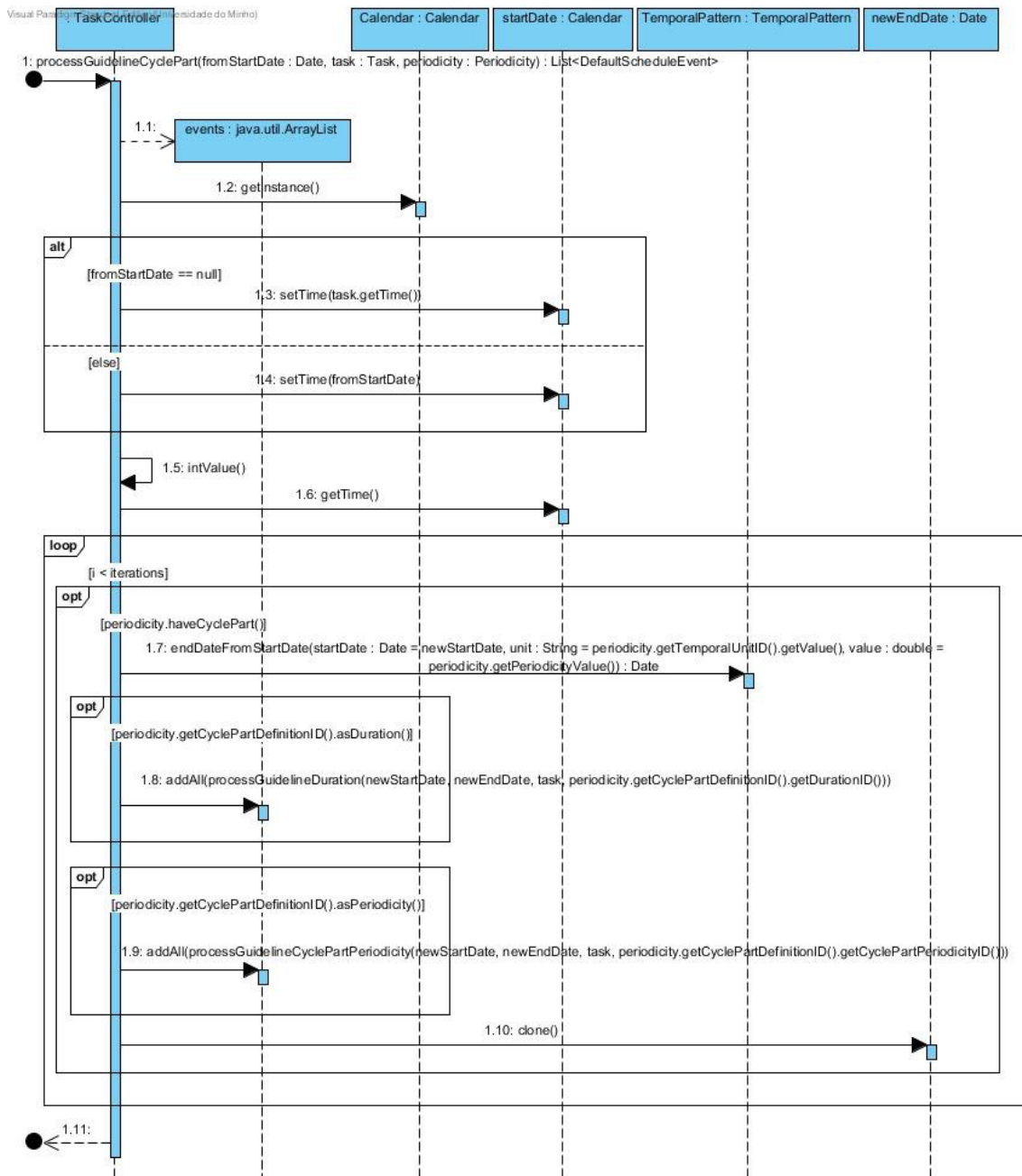


Figura 36 - Diagrama de sequência Processar Parte de Cada Ciclo.

O diagrama de sequência da Figura 37 representa o algoritmo que procura processar a periodicidade de cada parte de um ciclo das tarefas, existentes nos protocolos clínicos. Este algoritmo permite calcular o conjunto de sub-tarefas, de acordo com o valor especificado em *repetitionValue*, ou a duração dessas sub-tarefas de acordo com os atributos definidos no objeto da classe *Duration*.

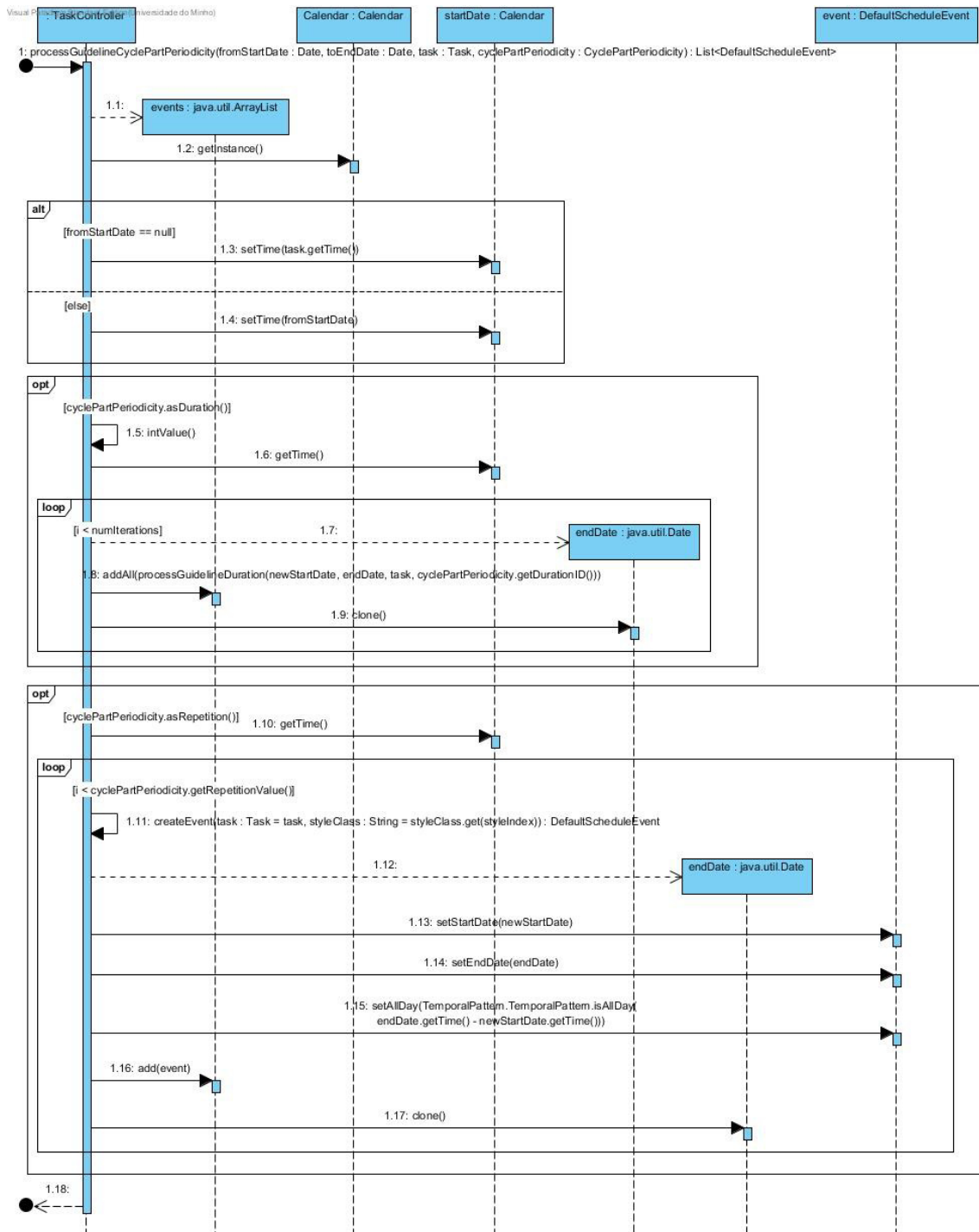
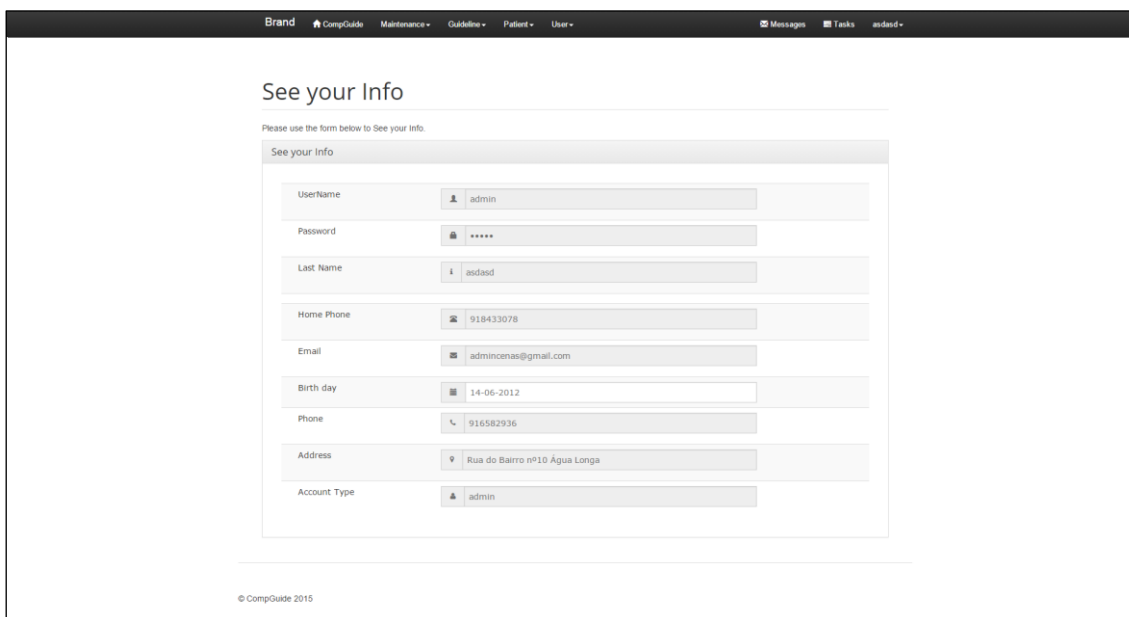


Figura 37 - Diagrama de seqüência Processar Periodicidade de Cada Parte de um Ciclo.

## D. Apresentação da Aplicação Web

Este anexo refere-se à apresentação da aplicação *web*, tendo como principal objetivo fornecer mais detalhes sobre as diferentes interfaces disponíveis.

Relativamente aos dados da conta de cada utilizador, este poderá visualizar a sua informação bem como alterar os seus dados, conforme se pode visualizar na Figura 38 e na Figura 39.



The screenshot displays a web interface for viewing and editing user information. The page title is "See your Info". Below the title, there is a prompt: "Please use the form below to See your info". The form itself is titled "See your Info" and contains the following fields:

Field	Value
UserName	admin
Password	*****
Last Name	asdasd
Home Phone	918433078
Email	adminceas@gmail.com
Birth day	14-06-2012
Phone	916582936
Address	Rua do Bairro nº10 Água Longa
Account Type	admin

At the bottom of the page, there is a copyright notice: "© CompGuide 2015".

Figura 38 - Página de visualização da informação pessoal.

The screenshot shows a web application interface for editing a user's profile. At the top, there is a navigation bar with 'Brand', 'CompGuide', 'Maintenance', 'Guideline', 'Patient', and 'User' menus, along with 'Messages', 'Tasks', and 'asad' icons. The main heading is 'See your Info'. Below it, a sub-heading reads 'Please use the form below to See your Info.' The form itself is titled 'See your Info' and contains several input fields: 'UserName' (admin), 'Password' (masked with dots), 'Last Name' (asdasd), 'Home Phone' (918433078), 'Email' (admincenas@gmail.com), 'Birth day' (14-06-2012), 'Phone' (916582936), 'Address' (Rua do Bairro nº10 Água Longa), and 'Account Type' (admin). A green 'Save Changes' button is at the bottom of the form. The footer of the page indicates '© CompGuide 2015'.

Figura 39 - Página de edição da informação pessoal.

O Administrador da aplicação poderá adicionar profissionais de saúde no sistema de acordo com a Figura 40.

The screenshot shows a web application interface for creating a new user. The navigation bar is identical to the previous page. The main heading is 'Create User'. Below it, a sub-heading reads 'Please use the form below to Create User.' The form is titled 'Create User' and contains several input fields, all of which are currently empty and display the placeholder text 'There is no Information': 'UserName', 'Password', 'Last Name', 'Home Phone', 'Email', 'Phone', 'Address', and 'Account Type'. The 'Birth day' field has a placeholder 'Enter Birth day'. A green 'Create' button is at the bottom of the form. The footer of the page indicates '© CompGuide 2015'.

Figura 40 - Página criação de Profissionais de Saúde.

Além disso poderá editar e eliminar a informação dos Profissionais de Saúde conforme ilustrado na Figura 41 e na Figura 42.



Brand | CompGuide | Maintenance | Guidelines | Patient | User | Messages | Tasks | ajuda |

visualize all information that you need.

### Search User

Please use the form below to Search User.

Search User

Search for User  
admin

UserName

Password

Last Name

Home Phone

Email

Phone

Address

Account Type

Figura 41 - Página edição de Profissionais de Saúde.

Brand | CompGuide | Maintenance | Mudou para o ecrã inteiro. | Sair do modo de ecrã inteiro (F11) | Messages | Tasks | ajuda |

### Remove User

Please use the form below to Remove User.

Remove User

Search User  
Tiago Oliveira

UserName

Password

Last Name

Home Phone

Email

Phone

Address

Account Type

© CompGuide 2015

Figura 42 - Página eliminar Profissionais de Saúde.

O Administrador da aplicação poderá ainda gerir toda a base de dados através do separador *Maintenance* que permite numa só página visualizar todos os dados de uma tabela, bem como editar, eliminar e inserir informação (Figura 43).

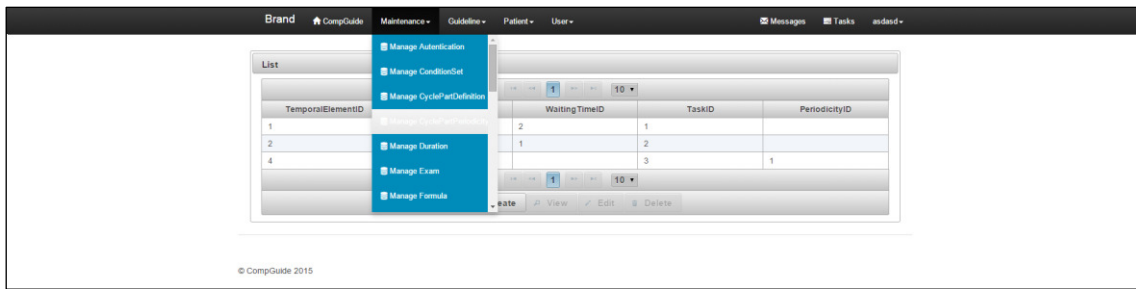


Figura 43 - Página para a gestão da informação de uma tabela da Base de Dados.

Também é possível ao Administrador gerir os protocolos clínicos e os pacientes do sistema através dos separadores *Guideline* e *Patient* que permitem criar, eliminar e editar estas entidades, tal como ilustrado na Figura 44 e na Figura 45.

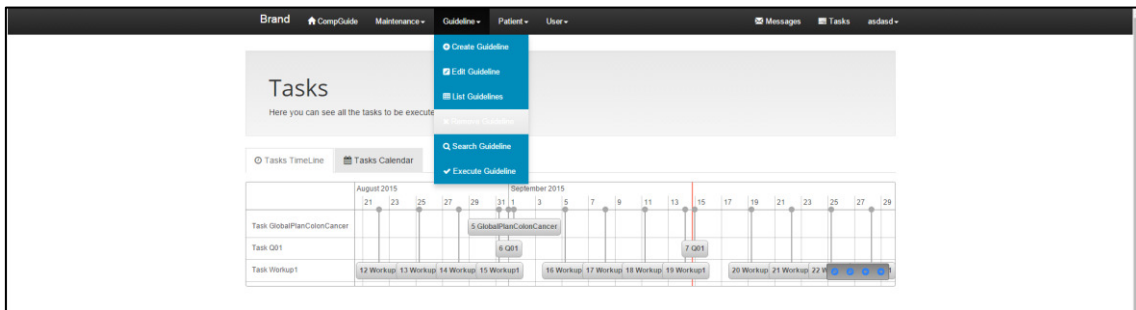


Figura 44 - Separador para gerir os protocolos clínicos.

De referir que as páginas para editar, criar, e eliminar estas entidades são idênticas às páginas para gerir os profissionais de saúde, mudando apenas a entidade que se pretende gerir.

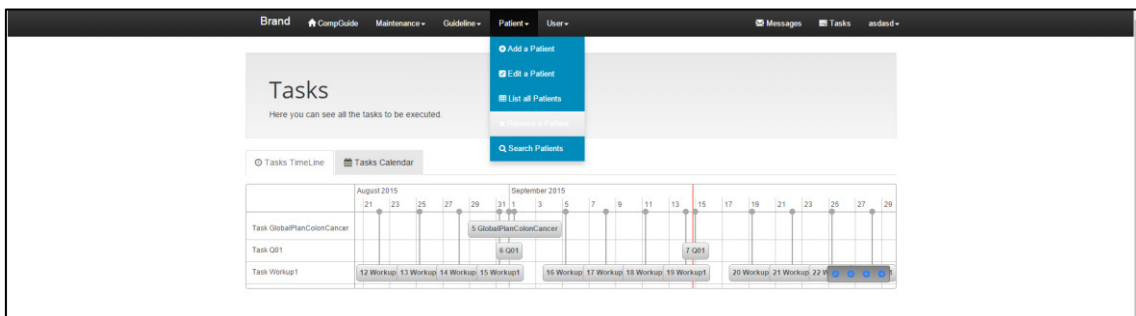


Figura 45 - Separador para gerir os Pacientes.