

Universidade do Minho

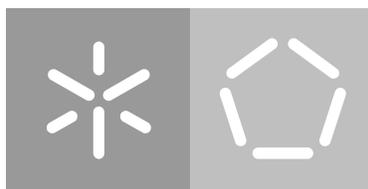
Escola de Engenharia

Departamento de Informática

Luís Miguel Moreira Fernandes

**Mecanismos de Profiling para
Sistemas de Avaliação de Conhecimento**

Outubro 2019



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Luís Miguel Moreira Fernandes

**Mecanismos de Profiling para
Sistemas de Avaliação de Conhecimento**

Dissertação de Mestrado

Mestrado Integrado em Engenharia Informática

Dissertação orientada por

Professor Doutor Orlando Manuel de Oliveira Belo

Outubro 2019

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos. Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



Atribuição-Compartilha Igual
CC BY-SA

<https://creativecommons.org/licenses/by-sa/4.0/>

Agradecimentos

Em primeiro lugar quero agradecer ao meu orientador, o Professor Orlando Belo, por ter aceitado orientar-me, por todo o apoio e disponibilidade, e ainda, pelos contributos para o avanço deste trabalho.

Seguidamente quero agradecer a todos os professores que conheci no decorrer destes cinco anos pelo conhecimento que me transmitiram, pois sem ele não seria possível a realização deste projeto e desta dissertação.

Agradeço aos *Nerds*, os meus colegas de curso e amigos, que sempre me acompanharam e apoiaram em todos os momentos. Juntos vivemos e partilhamos emoções que seria difícil escrever neste pequeno pedaço de papel, mas também é necessário dizer, não seria o local apropriado para as partilhar.

Seria impossível não agradecer, particularmente, ao meu parceiro de investigação e amigo João Coelho, que me acompanhou ao longo desta jornada de muitos altos e baixos e de muitas horas de luta.

Quero agradecer a toda a minha família que sempre esteve presente. Dando um especial e particular obrigado aos meus pais por me terem ensinado a ser resiliente e também por toda a compreensão e paciência nos dias mais difíceis.

Por último, mas definitivamente não menos importante, quero agradecer à Isabel, minha namorada, por toda a calma e paciência que teve para comigo ao longo destes meses, por todo o carinho que me transmitiu e por estar sempre presente quando eu mais precisava.

A todos o meu muito obrigado, sem cada um de vocês nada disto teria sido possível.

Declaração de Integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática do plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à elaboração.

Mais declaro que conheço e respeitei o Código de Conduta e Ética da Universidade do Minho.

Resumo

Mecanismos de *Profiling* para Sistemas de Avaliação de Conhecimento

Nos dias que correm, é cada vez mais frequente o recurso à tecnologia para a resolução de problemas nas mais diversas atividades, nos mais variados setores. O setor da educação não é, pois, uma exceção. Nesse setor, um dos aspetos que tem vindo a obter alguma relevância aborda a temática dos ITS – *Intelligent Tutoring Systems*, apesar da timidez das diversas aproximações realizadas, em especial ao nível do ensino de nível académico. Atualmente, está em curso na Universidade do Minho o desenvolvimento de um ITS, denominado Leonardo, cujo objetivo principal é o auxílio dos alunos no seu processo de aprendizagem e formação. Este sistema pretende fornecer um acompanhamento personalizado ao utilizador, tanto em termos do seu processo de formação, como, posteriormente, em termos do processo de avaliação correspondente. Nesta dissertação apresenta-se o desenvolvimento de um sistema de *profiling* para apoio ao processo de avaliação, que possibilita a construção do perfil do utilizador (do estudante) à medida que este interage com o sistema ao longo do tempo. Este sistema de *profiling* representa um dos módulos do sistema Leonardo e, através das devidas interações com os restantes módulos, serve de base de apoio a todas as decisões que são tomadas no sistema, uma vez que estabelece o perfil do utilizador pelo qual o sistema se deve reger ao adaptar os conteúdos a disponibilizar.

Palavras-chave: Base de conhecimento, Modelo de estudante, Processo de avaliação, Profiling, Sistemas de Tutoria Inteligentes.

Abstract

Profiling Mechanisms for Knowledge Evaluation Systems

Nowadays, technology is increasingly used to solve problems in a wide range of activities, in a wide range of sectors. The education sector is therefore not an exception. In this sector, one of the aspects that has come to some relevance is the subject of ITS – Intelligent Tutoring Systems, in spite of the timidity of the different approaches made, especially at the level of academic level teaching. Currently, the University of Minho is developing an ITS, called Leonardo, whose main objective is to help students in their learning and training process. This system intends to provide a personalized follow-up to the user, both in terms of their training process and, subsequently, in terms of the corresponding evaluation process. This dissertation presents the development of a profiling system to support the evaluation process, which allows the construction of the user's (student) profile as it interacts with the system over time. This profiling system represents one of the modules of the Leonardo system and, through appropriate interactions with the other modules, serves as the basis for all the decisions that are made in the system, as it establishes the user profile by which the system should govern itself to adapt the content to be made available.

Keywords: Evaluation process, Intelligent Tutoring Systems, Knowledge base, Profiling, Student model.

Índice

Introdução	1
1.1 Contextualização.....	1
1.2 Motivação e Objetivos	4
1.3 Apresentação do Trabalho Realizado	4
1.4 Organização da Dissertação	5
Trabalho Relacionado	8
2.1 <i>Background</i>	8
2.1.1 <i>Profiling</i>	8
2.1.2 Setores de Aplicação.....	9
2.2 Diferentes abordagens ao <i>Student Model</i> de um ITS.....	10
2.3 Análise Bibliográfica.....	13
2.3.1 (Chopra <i>et al.</i> , 2010)	14
2.3.2 (Morales-Rodríguez <i>et al.</i> , 2011)	16
2.3.3 (Niesler and Wydmuch, 2009)	19
2.3.4 (Schiaffino and Amandi, 2009).....	22
2.3.5 Comparação de Características.....	24
Um Modelo de <i>Profiling</i> para Avaliação de Conhecimento.....	28
3.1 Apresentação do Modelo.....	28

3.2	Elementos Base do Modelo.....	32
3.2.1	Estruturas Base do Modelo	32
3.2.2	Adição à Proposta de Valor do Modelo.....	38
3.2.3	Algoritmo de Alteração de Nível de Utilizador	44
	Um Caso de Aplicação.....	55
4.1	Leonardo	55
4.2	Arquitetura do Sistema	56
4.3	Ferramentas utilizadas.....	58
4.4	Enquadramento do Modelo de Avaliação	60
	Conclusões e Trabalho Futuro.....	76
5.1	Análise ao Trabalho Realizado.....	76
5.2	Trabalho Futuro	77
	Bibliografia	79
	Anexos	83

Índice de Figuras

Figura 1 - Arquitetura geral de um ITS apresentada por Butz, citada por (Morales-Rodríguez <i>et al.</i> , 2011), adaptada para PT-pt.	3
Figura 2 - Abordagens ao <i>Student Model</i> ao longo dos anos – imagem extraída de (Chrysafiadi and Virvou, 2013).	13
Figura 3 - Matriz de tutores orientados à preferência por (Niesler and Wydmuch, 2009) traduzida para PT-pt.	20
Figura 4 - Modelo do arquétipo do utilizador apresentado por (Niesler and Wydmuch, 2009) traduzida para PT-pt.	22
Figura 5 – Diagrama de atividades "Obter Padrão".....	30
Figura 6 – Estrutura do padrão base de um aluno.....	31
Figura 7 – Diagrama de atividades "Criar e atualizar perfis de utilizadores".	31
Figura 8 - Exemplo de uma questão preparada para o domínio 'Matemática'.	35
Figura 9 - Exemplo de um possível perfil de um aluno no sistema.....	43
Figura 10 - Exemplo de uma possível configuração de domínio.....	47
Figura 11 – Excerto do algoritmo de alteração de nível de utilizador em pseudocódigo.....	51
Figura 12 - Arquitetura do sistema Leonardo.	56
Figura 13 - Arquitetura do sistema Leonardo, contemplando o <i>Student Model</i>	58
Figura 14 – A API do <i>Profiler</i>	62
Figura 15 - Diagrama de Sequência relativo ao processo de avaliação.	64
Figura 16 - Interface do modelo de edição de questões de <i>quiz</i>	65

Figura 17 - Exemplo de uma questão do sistema em JSON.	66
Figura 18 - Escolha do domínio de estudo.	67
Figura 19 - Exemplo de uma possível questão.	67
Figura 20 - Exemplo de uma possível questão com a vista de um <i>smartphone</i>	68
Figura 21 - <i>Query</i> MongoDB para obter perfis no sistema.	69
Figura 22 - Exemplo de um possível perfil de um aluno no sistema.	71
Figura 23 - Página de resultados de um <i>quiz</i>	72
Figura 24 – Esboço de uma possível <i>dashboard</i> com dados relativos ao aluno.	73
Figura 25 - (Anexos) Diagrama de atividades "Criar e atualizar perfis de utilizadores"	84

Índice de Tabelas

Tabela 1 - Comparação das principais características entre os trabalhos estudados..... 25

Capítulo 1

Introdução

1.1 Contextualização

Nos últimos anos tem-se verificado uma maior adoção de soluções tecnológicas na área do ensino, algo fortuito de um ensino mais completo e eficaz que não pode ser proporcionado, exclusivamente, por um tutor humano. Esta lacuna tem vindo a ser gradualmente explorada no domínio dos sistemas de informação, com particular ênfase na área dos ITS e no desenvolvimento de tutores artificiais.

Guardia (1993), citado por (Morales-Rodríguez *et al.*, 2011), define tutores inteligentes como “(...) um sistema de instruções assistidas por computador, que utiliza técnicas de Inteligência Artificial, principalmente, para representar conhecimento e direcionar estratégias de ensino; é capaz de se comportar como um especialista, tanto no domínio do conhecimento que ensina (mostrando ao aluno como aplicar esse conhecimento), como no domínio pedagógico, onde é capaz de diagnosticar a situação em que o aluno se encontra e oferecer uma solução que permita ao mesmo progredir na aprendizagem”.

Um tutor artificial inteligente pode, então, ser definido como um conjunto de componentes de software que visam imitar o comportamento de um tutor humano (Anohina, 2007), visando fornecer um feedback personalizado aos alunos, durante a execução de tarefas, sem a intervenção de seres humanos. Estes sistemas têm por base a Teoria da Aprendizagem Eletrónica, que descreve os princípios da ciência cognitiva para uma aprendizagem suportada por meios multimédia, recorrendo a várias vertentes aplicacionais da tecnologia educacional. Os tutores artificiais inteligentes vêm, assim, elevar o conceito dos sistemas de *e-Learning* para um novo patamar, uma vez que permitem a disponibilização de conteúdos educacionais adequados ao perfil do utilizador, refletindo o seu conhecimento e personalizando o processo de aprendizagem, através da criação de modelos descritivos baseados no conhecimento e nas capacidades que o aluno vai revelando nas suas ações e decisões aquando da interação com o sistema.

Durante os últimos dois anos tem vindo a ser idealizado e desenvolvido um projeto interno na Universidade do Minho para o desenvolvimento de um ITS, atualmente reconhecido como o sistema Leonardo¹, que visa fornecer um tutor artificial para apoio ao estudo das várias matérias num conjunto de unidades curriculares específico. Este sistema foi concebido com o propósito de, inicialmente, servir de apoio ao estudo na Unidade Curricular (UC) de Bases de Dados do 3º ano do curso Mestrado Integrado em Engenharia Informática. No entanto, uma das principais características deste sistema é a possibilidade de ser usado em qualquer outra área de conhecimento, desde que se prepare o correspondente corpo de conhecimento relativo ao domínio em causa.

Na Figura 1, que se segue, é possível vislumbrar a arquitetura base de um ITS, bem como os seus módulos e componentes essenciais.

¹ O 'Leonardo' é uma plataforma de ensino on-line, especialmente orientada para suportar processos de aferição de conhecimento em vários domínios de estudo. O sistema foi idealizado pelo Professor Orlando Belo, do Departamento de Informática da Universidade do Minho, que desde 2017 tem promovido e suportado diversas ações de investigação e desenvolvimento relacionadas com os vários componentes do sistema que tem vindo a arquitetar.

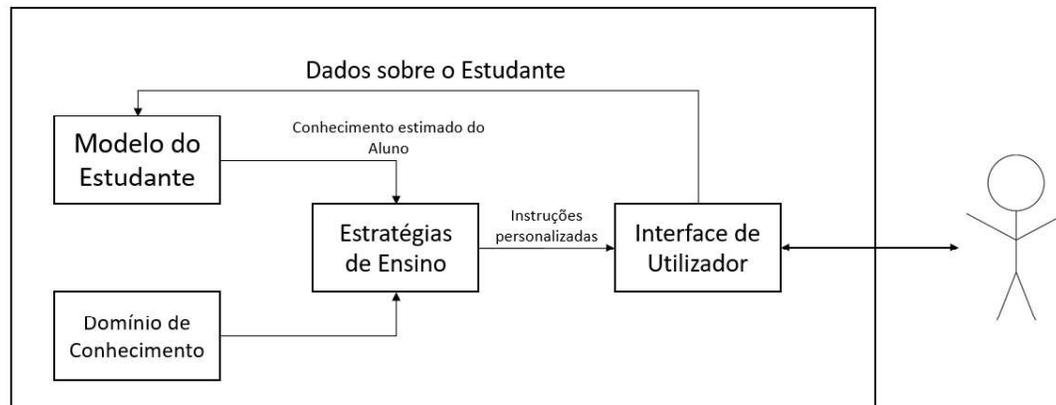


Figura 1 - Arquitetura geral de um ITS apresentada por Butz, citada por (Morales-Rodríguez *et al.*, 2011), adaptada para PT-pt.

Neste trabalho de dissertação pretendeu-se conceber, implementar e trabalhar sobre um dos módulos deste sistema, em concreto o Modelo do Estudante (*Student Model*), que é responsável por definir e atualizar os perfis dos utilizadores do sistema em termos de conhecimento nos domínios explorados, com base na informação que vai recebendo sobre os utilizadores ao longo do tempo. “Construir um modelo de estudante envolve definir quem o aluno é, incluindo a história e limitações físicas do mesmo; o que deve ser modelado, por exemplo, os objetivos, planos, atitudes, capacidades, conhecimentos e crenças do aluno; como o modelo deve ser adquirido e mantido; e porque é que a informação está a ser procurada por/sobre o aluno, seja para dar assistência ou fornecer feedback ao aluno, ou para interpretar o comportamento do mesmo” ((Stauffer, 1996), citado por (Alqahtani, 2011)).

Será através do uso deste módulo que o sistema terá a capacidade de delinear os perfis de conhecimento do utilizador e fazer a adaptação e personalização de conteúdos à medida dos utilizadores, dos alunos.

1.2 Motivação e Objetivos

Desde a viragem do século que é notória a evolução da tecnologia ao longo do tempo e a sua gradual utilização pelas pessoas no seu dia-a-dia. Porém, apesar de alguns esforços de louvar, se olharmos especificamente para o domínio da educação, é possível verificar uma clara estagnação da aplicação da tecnologia mais mundana nos processos de aprendizagem, nos processos educativos. Esta realidade começou a tentar ser combatida com o aparecimento dos ambientes de aprendizagem assistidos por computador, e, mais tarde, pelos tutores artificiais, mais completos e ditos inteligentes.

Este trabalho de dissertação integra-se no domínio deste tipo de sistemas de ensino. Em termos gerais, pretende-se conceber e desenvolver um módulo de *profiling* que possa ser integrado na arquitetura base de um tutor artificial específico, de forma a possibilitar-lhe o acompanhamento do processo de ensino e aprendizagem de um aluno de forma individualizada e ajustada ao seu conhecimento. Através do acompanhamento do aluno durante os seus processos de avaliação, processos estes usados usualmente para se conhecer o estado atual do aluno nas várias matérias em avaliação, este módulo permitirá fazer a definição do perfil do aluno ao longo do tempo, com o objetivo de inferir os aspetos mais críticos do processo de aprendizagem desse aluno. Dessa forma será possível fazer a recomendação do estudo de uma dada matéria para suprir as falhas de formação identificadas.

1.3 Apresentação do Trabalho Realizado

Após uma primeira fase de pesquisa e estudo sobre trabalhos relacionados no âmbito dos ITS, que expusessem a criação e modelação de um *Student Model*, foi altura de efetuar um levantamento de requisitos, com o intuito de estruturar o que deveria ser desenvolvido.

Neste trabalho de dissertação foi, assim, elaborado um *Student Model*, denominado *Profiler*, capaz de integrar o caso de estudo em mãos, o sistema Leonardo e baseado nos requisitos anteriormente levantados.

Este módulo de estudante será o responsável pela criação e atualização dos perfis dos utilizadores (*profiling*) através das interações que os mesmos vão efetuando com o sistema durante o seu processo de avaliação. Do mesmo modo, apresenta-se também como responsável pela distribuição da informação recolhida, de forma organizada e criteriosa, para os diversos módulos que se vejam necessitados desta, dando azos, assim, à sua característica arquitetural e comportamental como um micro-serviço.

1.4 Organização da Dissertação

Além deste primeiro capítulo, no qual foi sucintamente descrita uma introdução à dissertação e aos objetivos e motivação por detrás da mesma, este documento conta, ainda, com mais quatro capítulos:

- **Capítulo 2** – apresenta uma definição do que é o *profiling*, bem como dá a conhecer setores em que o mesmo é, ou pode ser, aplicado. São ainda apresentadas várias possíveis abordagens a um modelo de *profiling* passível de integrar num ITS e expostos e explicados alguns exemplos concretos de trabalhos nesta área.
- **Capítulo 3** – em que expõe uma definição geral do modelo desenhado e implementado nesta dissertação, dando a conhecer os seus elementos base, bem como a sua orgânica. No final é, ainda, demonstrada, graficamente, a aplicação deste modelo.
- **Capítulo 4** – aqui é introduzido o caso de estudo que serviu de base para esta dissertação, a iniciativa Leonardo. É neste capítulo que é possível verificar a integração

deste modelo no ITS ainda em desenvolvimento, bem como apurar a sua interligação com os restantes módulos existentes no sistema. No final apresenta-se ainda uma validação do modelo no enquadramento deste sistema, no que diz respeito ao processo de avaliação.

- **Capítulo 5** – Neste último capítulo são expostas algumas considerações finais, passando por um resumo das principais ideias abordadas no restante documento, seguindo-se uma avaliação crítica do trabalho realizado, manifestando pontos que considerámos positivos e negativos durante a sua realização. Finaliza-se o capítulo apontando algumas linhas de trabalho futuro, como meio para uma melhoria deste modelo e aprimoramento do sistema como um todo.

Capítulo 2

Trabalho Relacionado

2.1 *Background*

Numa primeira fase é importante adquirir noções do que é o conceito de *profiling*, tendo em consideração algumas bases sobre o qual o mesmo assenta. Esta primeira secção apresenta exatamente o mencionado, expondo uma definição de *profiling*, e abordando alguns conceitos básicos, bem como apresentando alguns setores nos quais o mesmo é, atualmente, aplicado.

2.1.1 *Profiling*

Segundo a *Wikipedia*, na ciência da informação, *profiling* refere-se ao processo de construção e aplicação de perfis de utilizadores gerados através da análise de dados computadorizados (Wikipedia, 2018). Esta construção ou criação de perfis pode ser definida como a arte de recolher informações sobre algo e usar as mesmas para fornecer uma descrição (ISProf, 2017).

A um nível mais geral, *profiling* existe no dia-a-dia de cada ser humano, seja quando conhecem alguém novo e, por vezes inconscientemente, tentam detetar certas características da personalidade que tornem essa pessoa alguém mais provável a um bom relacionamento, ou, até, quando os mesmos se dirigem à mercearia para comprar fruta e a apalpam como forma de decidir se a mesma corresponde às suas expectativas e/ou padrões. Estes exemplos não são mais que um *profiling* inconsciente realizado por qualquer ser humano, nas mais variadas ocasiões que a vida proporciona.

2.1.2 Setores de Aplicação

O *profiling*, como referido anteriormente, faz parte da vida de todos, e, por isso mesmo, encontra-se presente e pode ser aplicado numa grande variedade de domínios e para uma grande panóplia de objetivos (Kanoje, Girase and Mukhopadhyay, 2014).

Abordando, por exemplo, o setor comercial, o conhecimento sobre os comportamentos e preferências do cliente é de grande importância. A essência do *profiling*, neste domínio em específico, passa pela previsão dos comportamentos de diferentes grupos de clientes perante certas situações. Exemplo disto são as estratégias de marketing talhadas especificamente para atingir um grupo específico de clientes, como cartões de fidelização ou publicidade personalizada (Epic.org, 2019).

No setor financeiro, as técnicas de *profiling* são, maioritariamente, utilizadas na prevenção de fraude e no *credit scoring* (método para análise do risco ao conceder crédito), facilmente explicável pela necessidade dos bancos em tentar minimizar ao máximo o risco quando cedem crédito aos clientes, conseguindo, recorrendo a técnicas de *profiling*, detetar, com maior precisão, potenciais clientes com impossibilidade de pagar a dívida (Canhoto, 2007).

Se o foco recair sobre o contexto criminal e da ciência forense, o *profiling* refere-se ao processo de identificação de traços de personalidade, tendências comportamentais, localizações

geográficas e descritores demográficos ou biológicos de um infrator, com base nas características de um crime. É, essencialmente, usado para estabelecer perfis de risco de potenciais suspeitos, tendo em conta os aspetos apresentados, comparando com casos e suspeitos passados, procurando estabelecer padrões comuns (Kocsis, 2006).

Já no contexto da aprendizagem, área já abordada na introdução deste documento, o processo de *profiling* pode ser usado como ferramenta que possibilite a apresentação de conteúdos multimédia personalizados para cada utilizador, bem como para deteção de padrões e comportamentos de atenção durante a aprendizagem, de forma a permitir um melhor suporte e gestão da mesma (Nabeth, 2008).

2.2 Diferentes abordagens ao *Student Model* de um ITS

Nesta segunda secção do capítulo, e para um *background* completo dos temas que irão ser abordados ao longo deste documento, é importante apresentar diferentes abordagens ao que é um *Student Model* de um ITS.

Segundo Chrysafiadi e Virvou (2013) é possível distinguir 9 diferentes abordagens distintas, cada uma delas com os seus prós e contras, para a implementação do modelo do estudante, sendo elas:

- **Sobreposição (*Overlay*)** – O conhecimento do estudante, num certo domínio, é considerado um subconjunto do conhecimento de um especialista nesse mesmo domínio, sendo que o objetivo deste modelo é apresentar conteúdos ao aluno de forma a que o subconjunto do seu conhecimento seja cada vez mais aproximado do conjunto total de conhecimento do especialista.
- **Estereótipos (*Stereotypes*)** – Tipo de modelação bastante comum, onde são criados *clusters* de utilizadores tendo por base pontos comuns que os mesmos demonstrem. As vantagens do uso deste tipo de modelação aparecem da inferência do conhecimento de um

utilizador com base no grupo estereótipo em que se insere. Por outro lado, os grupos serão estáticos até que, por mão humana, sejam alterados, e desta forma impossibilitam flexibilidade para melhor representação dos *clusters* de utilizadores (Sani, Bichi and Ayuba, 2016).

- **Perturbação (*Perturbation*)** – Extensão do tipo de modelação *overlay*, em que, além do que foi mencionado sobre o mesmo, é acrescentado um outro subconjunto representativo de possíveis equívocos de conhecimento nesse domínio específico. Este modelo é útil como forma de diagnóstico de pensamento do utilizador.
- **Técnicas de *Machine Learning*** – Modelo baseado na observação do comportamento dos utilizadores como exemplos de treino para que, posteriormente, sistemas de *machine learning* possam usar para criar modelos desenhados para prever futuras ações do utilizador (Webb, Pazzani and Billsus, 2001).
- **Teorias Cognitivas** – Uma teoria cognitiva tenta explicar o comportamento humano durante o seu processo de aprendizagem, detetando e analisando os processos de pensamento e raciocínio do utilizador. Modelo usado recorrentemente para reconhecer o estado emocional do utilizador durante o seu processo de aprendizagem.
- **Modelo Baseado em Restrições (*Constraint-based*)** – Modelo proposto por (Ohlsson, 1994), que acreditava na aprendizagem através dos erros. Este modelo caracteriza-se por representar o conhecimento de um domínio como um conjunto de condições e o *Student Model* como um conjunto de condições que não foram cumpridas pelo utilizador (Martin, 1999). Ohlsson, citado por (Chrysafiadi and Virvou, 2013), afirma que é perigoso assumir que as condições que o utilizador não erra se considerem como conhecimento adquirido, uma vez que essa condição pode ter sido atingida inadvertidamente, no entanto, uma condição errada irá sempre demonstrar um erro de conhecimento do utilizador ou um deslize no processo de aprendizagem.
- **Fuzzy modeling** – Este modelo aparece como uma metodologia para incorporar a incerteza de um qualquer problema do quotidiano causado por informação incompleta ou imprecisa, ou ainda por subjetividade humana. Esta capacidade foi verificada como prometedora para incorporar num ITS, com vista a aumentar a performance de um ambiente

educacional, no qual decisões sobre que material de aprendizagem ou conselhos e notas a fornecer, personalizadas, ao aluno, devem ter em conta um conjunto de critérios e especificações que englobem um valor de incerteza a qualquer decisão.

- **Redes Bayesianas (*Bayesian networks*)** – Ferramenta para representar incerteza no *Student Model*, caracterizada por um grafo acíclico e direcionado, no qual cada nodo representa uma variável e os arcos a dependência probabilística ou relação causal entre estas variáveis. Esta ferramenta é útil para desenhar modelos capazes de analisar sequências de ações e consequências das mesmas, bem como o resultado esperado.

- **Modelação Orientada à Ontologia (*Ontology-based*)** – O facto de uma ontologia suportar a representação de conceitos e propriedades abstratas com o intuito de serem reutilizadas e, em caso de necessidade, estendidas para diferentes contextos aplicativos, torna este modelo apelativo para a representação do conhecimento do aluno ao longo do tempo pela forma volátil como este tipo de dados são caracterizados.

Na Figura 2 é possível ver o resultado de um estudo sobre vários ITS e a modelação utilizada na conceção do *Student Model* (Chrysafiadi and Virvou, 2013).

	Overlay	Stereotypes	Perturbation	Machine learning	Cognitive theories	Constraint-based	Fuzzy	Bayesian networks	Ontology-based
Web-EasyMath		X		X					
Andes						X		X	
Millán & Perez de la Cruz, 2002					X			X	
Xu et al., 2002							X		
Conati & Zhou, 2002					X				
F-SMILE		X			X				
INSPIRE	X	*					X		
Web-PTV		X		X					
Surjono & Maltby, 2003	X	X	X						
SQTL-Web						X			
ACE								X	
F-CBR-DHTC		*					X		
TADV	X						X		
LeCo-EAD			X						
WILEDS	X								
MEDEA	X								
InterMediActor	X						X		
The personal reader	X								X
Baker et al., 2004				X					
KERMIT	X					X			
ICICLE	X	X							
Web-IT		X			X				
VIRGE					X				
Stathakopoulou, Magoulas, Grigoriadou & Samaracou, 2005							X		
COLLECT-UML						X			
InfoMap	X		X						
Kumar, 2006a; Kumar, 2006b	X								
Salim & Haron, 2006		*					X		
English ABLE								X	
Baker, 2007				X					
DeLC	X	X							
AUTO-COLLEAGUE		X	X						
Kofod-Petersen et al., 2008		X							
Web_Tutor_Pas		X							
AMPLIA					X			X	
Carmona et al., 2008		*						X	
Alepis et al., 2008					X				
DesignFirstITS								X	
ADAPTAPlan				X			X		
e-Teacher								X	

* Stereotype-like.

Figura 2 - Abordagens ao *Student Model* ao longo dos anos – imagem extraída de (Chrysafiadi and Virvou, 2013).

2.3 Análise Bibliográfica

Durante a pesquisa pela descoberta de trabalhos relacionados, o foco recaiu sobre artigos e estudos que explorassem diferentes abordagens, em específico, ao *Student Model* e à aplicação de *profiling* das mais variadas formas e perspectivas em, na sua maioria, sistemas de ensino (inteligentes) computacionais e tutores artificiais, tentando sempre explorar os seus principais componentes e entender a sua orgânica.

Os estudos apresentados de seguida não se focam numa abordagem concreta, entre as apresentadas no ponto anterior, ao *Student Model*, mas sim em tentativas de exploração de diferentes temáticas que pudessem ser incluídas num *Student Model* como complemento ao que foi anteriormente enumerado.

No final do capítulo pode-se encontrar uma grelha na qual, de forma resumida, se encontra explícita a disparidade entre abordagens a esta problemática, bem como a forma como se enquadra o módulo em desenvolvimento nesta dissertação no meio destas várias abordagens e que pontos poderão ou não ter em comum com algumas delas.

2.3.1 (Chopra *et al.*, 2010)

Chopra *et al.* (2010) residem o seu foco na preparação de um ITS *web-based* capaz de apresentar *hypermedia* adaptada ao perfil de cada estudante. Neste sistema, cada domínio é representado por um conjunto adaptável de atributos, sendo os atributos base: a necessidade de pensamento analítico; de memorização; de aptidão técnica; necessidade de pré-requisitos e de pensamento criativo. Cada estudante é, também, representado por um conjunto adaptável de atributos, no qual se incluem os atributos de domínio, enumerados anteriormente, uma vez que denotam a aptidão do estudante para diferentes tipos de raciocínio, e se adicionam outros como a resistência de concentração e as preferências sobre o tipo de *hypermedia* apresentado (áudio, vídeo, texto, imagem). Cada atributo possui um peso associado. No entanto, enquanto que os pesos atribuídos aos atributos de um domínio são da responsabilidade do *expert* com base em estimativas pessoais, os pesos dos atributos do estudante serão, ao longo da interação com o sistema, adaptados, reescritos e reinterpretados. Para além dos mencionados, há ainda que ter em consideração os pesos que provêm do nível de dificuldade das questões, inicialmente fornecidos pelo *expert* e que se encontram armazenado na configuração das questões. Um ponto interessante a reter deste estudo, é a flexibilidade no nível de dificuldade das questões, que podem sofrer alterações tendo como base o estudo do número de alunos que responde correta e incorretamente

às mesmas, permitindo, assim, que o nível de dificuldade possa aumentar ou diminuir caso a taxa de acerto a uma dada questão seja muito elevada ou reduzida.

Este estudo apresenta uma *frame network* que denota crescimento horizontal e vertical. Progredir de um *frame* para o seguinte, verticalmente, significa que será apresentado o mesmo conteúdo, mas com mais informação ou com informação mais profunda e mais complexa. Se a progressão se verificar horizontalmente, ocorre mudança para um tópico mais complexo dentro do mesmo domínio.

Chopra *et al.* (2010) aliam-se a um modelo de *quizzes* e *weighted performance* dos alunos, apresentando um *workflow* de sistema simples: apresentação de um *frame* com conteúdos adaptados ao *ranking* do aluno – este *ranking* é calculado através da *performance* do estudante ao longo do tempo de interação com o sistema, com base num modelo matemático que tem em conta todos os pesos mencionados no primeiro parágrafo; de seguida é efetuado um *quiz* como forma de avaliação, em que são registadas as respostas do aluno; com base nas respostas e nos dados obtidos, são atualizados os pesos dos atributos e mostrado um novo *frame* com base no crescimento demonstrado pelo *ranking* atual do aluno.

O novo *ranking* do estudante será, então, calculado com base nos pesos dos atributos do mesmo e denotará um crescimento, diga-se positivo ou negativo, em relação ao *ranking* imediatamente anterior, a que Chopra *et al.* (2010) chamam *growth factor* (Gf). Este fator de crescimento é comparado a um valor T (*threshold* para assegurar o rácio de crescimento do estudante) previamente definido e um de três casos é possível: **Gf > T; -T < Gf < T; Gf < -T.**

Num primeiro caso, em que o crescimento do estudante é positivo e significativo, o *frame* apresentado de seguida irá, necessariamente, denotar um desenvolvimento positivo, pelo que será apresentado um *frame* hierarquicamente acima, seja horizontal ou vertical. Esta escolha em termos de crescimento horizontal ou vertical é dado por uma *beta-distribution* – que representa a distribuição de probabilidade de, por exemplo, num próximo lançamento de uma moeda obter cara e coroa, tendo em conta o conhecimento prévio de lançamentos anteriores e crenças nos possíveis estados da moeda (normal, enviesada, ...) (Lu, 2019).

No segundo caso, uma vez que é denotado um desenvolvimento consistente e dentro dos parâmetros expectáveis, é apresentado ao aluno um *frame* verticalmente acima, abordando o mesmo tópico, mas acrescentando conceitos mais complexos.

O terceiro caso é similar ao primeiro, diferindo apenas no sentido do crescimento, que, neste caso, passará pela apresentação de um *frame* hierarquicamente abaixo, seja horizontal ou vertical.

Outro ponto de interesse no estudo em análise observa-se na construção de clusters, utilizando *Possibilistic Clustering Algorithm* (Krishnapuram and Keller, 1993). Chopra *et al.* (2010) defendem que, uma vez se tratando de um sistema para humanos, é seguro afirmar que nenhum dos atributos dos estudantes deverá mudar drasticamente num pequeno período de tempo, e, desta forma, grandes quantidades de dados, ou dados drasticamente diferentes dos registados anteriormente, não devem ser tidos em conta. Exemplos como, um aluno ter uma prestação consideravelmente má, sendo que a mesma se pode dever a, por exemplo, problemas de saúde, não deve, de maneira alguma, afetar consideravelmente, de forma negativa, uma *performance* geral realizada ao longo do tempo. O mesmo se aplica para um caso contrário, uma execução exceccionalmente boa pode dever-se a, por exemplo, efetuar aquela interação com a presença e auxílio de um *expert*, pelo que não deve afetar consideravelmente, de forma positiva, a *performance* geral do aluno.

2.3.2 (Morales-Rodríguez *et al.*, 2011)

Os autores Morales-Rodríguez *et al.* (2011) apresentam uma abordagem diferente à típica arquitetura de um ITS, refinando a mesma com o intuito de incluir preferências dos alunos no que concerne ao estilo de aprendizagem e, conseqüentemente, estratégias de ensino que possam ir de encontro a estas preferências mencionadas.

Gomez (2004), citado por (Morales-Rodríguez *et al.*, 2011), afirma que quando os estudantes são ensinados de acordo com o seu estilo preferido de aprendizagem tendem a

apresentar maior eficiência na mesma. Gomez (2004) avisa, no entanto, que atribuir etiquetas aos alunos é errado, uma vez que os estilos de aprendizagem preferidos, apesar de relativamente estáveis, podem mudar sob certas circunstâncias e estão sujeitos a melhorias.

Morales-Rodríguez *et al.* (2011) referem que entre os tipos de aprendizagem mais comuns e mais usados, se encontram os modelos de Felder and Silverman, de Kolb e modelo VARK. De seguida é possível encontrar uma breve introdução e explicação destes modelos, ditos mais comuns.

- **Modelo de Felder e Silverman** apresentado por Richard Felder e Linda Silverman, em 1988, este modelo classifica um indivíduo, através de um curto questionário contendo entre 13 e 16 questões, em quatro dimensões distintas, – **sensing/intuitive** (determina como prefere absorver a informação); **visual/verbal** (determina como prefere que a informação lhe seja apresentada); **active/reflective** (determina como prefere processar a informação); **sequential/global** (determina como prefere organizar e, progressivamente, direcionar-se até ao entendimento e retenção da informação) – sendo este mesmo indivíduo posteriormente, com base nas suas respostas, inserido no espectro do tipo de aprendizagem preferida (Ciloglugil, 2016).
- **Modelo de Kolb** apresentado por David Kolb, em 1984, cuja teoria da aprendizagem baseada em experiências é exposta em dois níveis: o primeiro trata-se de um ciclo de aprendizagem de quatro estados e o segundo, quatro distintos estilos de aprendizagem. “Aprender é o processo onde conhecimento é criado através da transformação de experiência” (Kolb, 1984).
No primeiro nível, Kolb apresenta a sua teoria da aprendizagem baseada em experiências como sendo um ciclo de quatro estados que, inevitavelmente, são presenciados. Este ciclo começa com a ocorrência de uma nova experiência ou situação (experiência concreta), passa pela fase de reflexão e revisão sobre a experiência observada (observação refletiva), de seguida encontra-se o estado onde a reflexão anterior dá origem a uma nova ideia, conceito ou alteração de um destes pré-existentes e conseqüente aprendizagem com a experiência presenciada

(conceituação abstrata) e, por fim, aplicação da(s) nova(s) ideia(s) ao ambiente que rodeia o indivíduo (experimentação ativa).

Tendo em conta a sua teoria, Kolb acredita que os estilos de aprendizagem podem ser divididos em quatro categorias: **Accomodating** – pessoas que, tendencialmente, regem-se pela intuição em detrimento da lógica, preferem uma abordagem mais prática; **Diverging** – pessoas capazes de olhar para as situações sobre diferentes perspetivas, preferem observar e recolher informação ao invés de experimentar; **Converging** – perfis tendencialmente menos preocupados com as pessoas e aspetos interpessoais (menor sensibilidade), nats em descobrir uso prático para ideias e teorias; **Assimilating** – pessoas de excelência em perceber informação em larga escala e em organizar a mesma de uma forma clara e lógica (McLeod, 2017).

- **Modelo VARK** apresentado por Neil Fleming, em 1987, cuja sigla na nomenclatura apresenta uma letra para cada um dos quatro tipos de aprendizagem referidos pelo autor. **Visual** – preferência sobre informação apresentada sob a forma de imagens, vídeos, diagramas ou gráficos; **Aural** – preferência por informação que pode ser ouvida ou falada, como palestras, discussões em grupo, rádio ou *web-chats*; **Reading/Writing** – preferência por informação que se apresenta por escrito, baseada em texto, tanto como *input* (leitura), como *output* (escrita); **Kinesthetics** – preferência pela recolha de informação através de experiências, como demonstrações ou simulações. À semelhança do que acontece com o modelo de Felder e Silverman, é através da análise das respostas a um questionário que os indivíduos são enquadrados num tipo específico de aprendizagem. De salientar que um indivíduo não necessita de possuir exclusivamente um tipo de aprendizagem predileto (Cherry, 2019) (VARK Learn Limited, 2019).

No seu estudo, Morales-Rodríguez *et al.* (2011) utilizam o último modelo apresentado, *VARK model*, para incorporar na sua arquitetura adaptada de ITS e definir o tipo preferido de

aprendizagem dos alunos, demonstrando, ao longo do documento, a abordagem de implementação do modelo referido.

2.3.3 (Niesler and Wydmuch, 2009)

Niesler and Wydmuch (2009) apresentam uma abordagem diferenciada à temática do *profiling* nos ITS, demonstrando um foco e especial atenção pelos fundamentos da psicologia por trás da criação de um perfil de utilizador. Os autores propõem, assim, um *profiling* baseado na predisposição para a aprendizagem do utilizador, bem como das suas preferências, claro está, no que à aprendizagem concerne.

A base do modelo de Niesler e Wydmuch assenta sob o método de definição do tipo de personalidade de Myers-Briggs, denominado Myers-Briggs Type Indicator (MBTI). O MBTI é baseado nos trabalhos de Carl Jung, que mostram como o temperamento afeta o comportamento dos humanos durante as suas ações e descreve as funcionalidades mentais baseadas em quatro critérios: atitude para com o mundo exterior; processamento de informação; tomada de decisões; organização da vida. Suportado por estes quatro critérios, o MBTI apresenta oito características de personalidade, organizadas duas a duas, sendo que um indivíduo tem, obrigatória e exclusivamente, um dos traços de personalidade de cada um dos pares. São eles:

- ***Extraversion*** que lida com situações e factos do mundo exterior;
Intraversion, que lida com ideias, pensamentos, informação, crenças e reflexões.
- ***Sensing***, que lida com factos, experiências, dados claros e tangíveis;
Intuition, que lida com ideias, com o desconhecido, com informação mais especulativa, conceptual e abstrata.
- ***Thinking***, que decide com base na lógica, usando abordagens analíticas e dando ênfase às tarefas e resultados que devem ser cumpridos;

Feeling, que respeita valores e crenças pessoais.

- **Judging**, que estável e organizado, focado em completar, organizar e planejar;

Perception, que mantém a flexibilidade, lida com os problemas à medida que os mesmos aparecem.

Niesler e Wydmuch (2009) referem que, no que aos tutores inteligentes diz respeito, a classificação nestes tipos de personalidade é insuficiente, pelo que apresentam uma proposta mais detalhada, em que é esmiuçado o conteúdo de cada par de características e são criados atributos que devem ser enquadrados através da análise das respostas dadas pelos utilizadores a um questionário. Na Figura 3 encontra-se a proposta dos autores para a aquisição das preferências de aprendizagem dos utilizadores.

Task presentation Program frame – tutoring organization		(S-N) Sensing – Intuition	(T-F) Thinking – Feeling
		(E-I) Extravert – Introvert	empirical – reasoning varied – thorough group – individual
(J-P) Judging – Perceiving	scheduled – adaptive result – method decision – analysis	detailed – general step-by-step – creative	prescriptive – descriptive calculation – contribution

Figura 3 - Matriz de tutores orientados à preferência por (Niesler and Wydmuch, 2009) traduzida para PT-pt.

Niesler e Wydmuch (2009) vão ainda mais longe e defendem que o perfil de utilizador deve seguir a divisão da realidade de Rene Decartes, que a particiona em duas áreas: *res cogitan* – do latim, mente – e *res extensa* – do latim, tudo além da mente.

Com foco neste ideal de divisão segundo Decartes os autores propõem uma arquitetura em que, em primeiro lugar, pertencente à área *res extensa*, se encontram as preferências de aprendizagem, também estas divididas em dois campos: preferências de processamento e preferências de orientação. A segunda área da teoria de Decartes, *res cogitan*, irá ser correspondida pela predisposição para a aprendizagem.

Mesmo com a clara divisão anteriormente apresentada, existem áreas que se interligam para criar novos conceitos: a junção da predisposição para a aprendizagem (correspondente a toda a área *res cogitan*) com as preferências de processamento (uma das divisões dentro da área *res extensa*) dará origem ao arquétipo do utilizador; por outro lado, as preferências de orientação aparecem sem interligação com outras áreas, uma vez que estão diretamente ligadas à camada de interface do ITS e apresentam, por exemplo, preferências de organização e calendarização do estudo.

Na Figura 4 é possível ver o modelo do arquétipo do utilizador proposto por (Niesler and Wydmuch, 2009) que, no restante estudo, apresentam ainda um diagrama de estado que demonstra o fluxo que o ITS deverá apresentar ao implementar este sistema de *profiling* diferenciado e com profundas bases da área da psicologia.

Learning determinants	Learning quality indicator	Measurement description
PREDISPOSITIONS		
MEMORY	high / low repetition	exclusive assign
	short, intensive / long, extensive training	
CONTENT	visualization	by percentage
	language: keywords, word relations	
	logic: causal-result relationship	
	formula representation	
UNDERSTANDING	analytics: deduction	
	synthetics: induction	
PREFERENCES		
INFORMATION PROCESSING	facts / ideas	exclusive assign
	practical / imaginative	
	detailed / general	
	step-by-step / creative	
PROBLEM SOLVING	logic / principles	
	solution / attitude	
	prescriptive / descriptive	
	calculation / contribution	

Figura 4 - Modelo do arquétipo do utilizador apresentado por (Niesler and Wydmuch, 2009) traduzida para PT-pt.

2.3.4 (Schiaffino and Amandi, 2009)

Schiaffino e Amandi (2009) apresentam um estudo sobre os principais conceitos no que concerne ao perfil dos utilizadores e do *User Model* para sistemas de *software* que se baseiam no conhecimento dos utilizadores.

Para este efeito, os autores denotam os conteúdos mais comuns de um perfil base de um utilizador: interesses; conhecimento; *background* e *skills*; objetivos; comportamentos; preferências de interação; características individuais e contexto dos utilizadores. Diferentes sistemas de *software*

utilizam conjuntos diferentes de conteúdos para representar o utilizador, sendo que os conteúdos devem ser adaptados às necessidades do *software* em questão. Neste trabalho foram exploradas formas de obtenção dos perfis de utilizador, que podem ser observados de forma explícita, através dos dados que os utilizadores inserem no sistema por via de questionários ou algum tipo de interface, ou de forma implícita, através da observação das ações dos utilizadores aquando da interação com o sistema, registo destas mesmas ações e posterior aplicação de técnicas de *machine learning* ou de *data mining* para detetar padrões e extrair o conhecimento.

Informação relativa a idade, género, ocupação, estado civil ou *hobbies* são exemplos de dados que podem ser obtidos de forma explícita. É, no entanto, na obtenção de dados de forma implícita que encontramos técnicas de *profiling* interessantes, com recurso a mecanismos inteligentes.

Estes autores depositam especial atenção em três mecanismos inteligentes de *profiling*: ***Bayesian networks***, ***association rules*** e, ainda, ***case-based reasoning***.

Schiaffino e Amandi (2009) definem uma *bayesian network* como uma representação compacta e expressiva de relações incertas entre variáveis de interesse de um domínio. (Jensen, 2001), citado pelos autores deste estudo em análise, refere que uma *bayesian network* é um grafo acíclico e orientado, onde os nodos representam variáveis aleatórias e os arcos representam correlações probabilísticas entre variáveis. Um exemplo prático de implementação de *bayesian networks* pode ser encontrado no trabalho (García *et al.*, 2007), no qual estas redes são usadas para modelar o comportamento dos estudantes com um sistema de *e-Learning*, de maneira a detetar o seu estilo de aprendizagem.

Association rules, em português regras de associação, são definidas, por Schiaffino e Amandi (2009), como regras que implicam relações de associação entre um conjunto de objetos de um certo domínio, sendo que estes ocorrem juntos ou então um implica outro. Em (Schiaffino and Amandi, 2006), são usadas regras de associação para descobrir preferências de interação do utilizador com um *interface agent*. Cada interação guarda a situação ou contexto que a originou, a assistência que o agente providenciou, a tarefa que o utilizador estava a efetuar na altura em que a interação aconteceu, o tipo da assistência fornecida e o *feedback* do utilizador como avaliação da

interação (sucesso, falha ou indefinido). Em (Chen, 2005), os autores usam regras de associação para efetuar o diagnóstico de erros comuns entre os estudantes durante os seus processos de aprendizagem, criando regras que ditam que se um erro A ocorre, então é provável que o erro B ocorra também.

Case-based reasoning é uma técnica que resolve novos problemas através da memorização de experiências anteriores similares (Kolodner, 1993, citado por (Schiaffino and Amandi, 2009)). O fluxo da resolução de problemas utilizando *case-based reasoning* passa por, aquando da chegada de uma nova situação, recuperar casos passados que apresentem similaridades com o novo caso em estudo, adaptar as soluções até que o novo problema esteja resolvido e, por fim, guardar este caso como mais uma experiência para poder ser recuperada no futuro. Em (González Serrano *et al.*, 2013) é usada uma abordagem *case-based reasoning* para modelar o perfil dos estudantes com maior rapidez e eficácia, com base em modelações anteriormente experienciadas. Da mesma forma, dando uso às competências desta abordagem em recuperar e adaptar experiências passadas para solucionar novos problemas, os autores utilizam um modelo de *case-based reasoning* para, segundo o perfil de um utilizador, recuperar conteúdos anteriormente apresentados a utilizadores com perfis semelhantes, e expô-los, adaptando-os em caso de necessidade.

2.3.5 Comparação de Características

Na Tabela 1 podemos ver, de forma resumida e simplificada, uma comparação das características que representam os vários trabalhos apresentados ao longo desta última secção, bem como o trabalho desta dissertação, no que diz respeito às várias possibilidades na abordagem à problemática do *profiling*.

Tabela 1 - Comparação das principais características entre os trabalhos estudados.

	Trabalho na secção 2.2.1	Trabalho na secção 2.2.2	Trabalho na secção 2.2.3	Trabalho na secção 2.2.4	<i>Profiler</i> desta dissertação
Atributos pesados	X				X
Estilos de Aprendizagem		X	X		X
Hipermédia adaptada	X		X		X
Técnicas de <i>Machine Learning</i>				X	

Na Tabela 1 é possível visualizar a influência que os vários estudos demonstraram para o trabalho apresentado nesta dissertação, uma vez que foram obtidas ideias e metodologias dos mesmos, possíveis de integrar no *Student Model* aqui apresentado, adaptadas para o contexto deste.

É, ainda, demonstrado na tabela que o único trabalho que, à primeira vista, não apresentou aparentes referências e ideais para o desenvolvimento do trabalho desta dissertação se encontra na secção 2.2.4 ((Schiaffino and Amandi, 2009)). Apesar do facto de, durante a execução do desenvolvimento do *Profiler* ao longo deste documento, não existirem referências ao trabalho de (Schiaffino and Amandi, 2009), o mesmo serviu, de facto, para estudo de possíveis integrações neste módulo, à semelhança dos restantes trabalhos estudados. No entanto a decisão passou pela construção de uma estrutura sólida, baseada em atributos indispensáveis à caracterização de um estudante, deixando espaço para eventuais melhorias e progressos numa área mais abstrata e com maior possibilidade de divergência de metodologias, como a área de *machine learning*, para o futuro.

No último capítulo deste documento serão abordadas algumas ideias de possíveis integrações no modelo construído, como referências para trabalho futuro, que se baseiam, essencialmente, em técnicas de *machine learning* ou possíveis abordagens ao *Profiler* que incorporem, de uma forma ou outra, técnicas desta referida temática.

Capítulo 3

Um Modelo de *Profiling* para Avaliação de Conhecimento

3.1 Apresentação do Modelo

O módulo de *Profiling* do sistema Leonardo, representando o *Student Model* deste ITS, apresenta-se como responsável pela criação e gestão dos perfis dos utilizadores deste sistema em termos de conhecimento nos domínios explorados, com base nas informações que vai recebendo sobre os mesmos ao longo do tempo de interação com o sistema, com o intuito de possibilitar a adaptação dos conteúdos apresentados a estes mesmos utilizadores, numa procura pelo aumento da sua produtividade e eficiência no que ao seu processo de aprendizagem diz respeito.

Sabemos que cada estudante possui uma própria maneira de aprendizagem, com ritmos e métodos característicos da sua personalidade. Sabemos também que a aprendizagem numa sala de aula, seja pelo seu ritmo, métodos, ou ambos, dificilmente se enquadrará e servirá como modelo de aprendizagem ideal para cada perfil de estudante. É, assim, importante que existam sistemas capazes de suportar as necessidades únicas de cada estudante, auxiliando-o sempre que o mesmo

precise, em todo e qualquer passo do seu autônomo e independente processo de aprendizagem de um determinado assunto.

Com o que foi anteriormente referido em vista, foi inicialmente formulado um conjunto de requisitos funcionais que este módulo deveria cobrir para que, em conjunto com os restantes módulos do sistema, demonstrasse a capacidade de apoio ao aluno necessária em todas as fases do seu processo de aprendizagem. Assim, foram estipuladas duas grandes principais ações para este módulo:

1) **criação de perfis de utilizador**, através da obtenção do padrão que será explicado mais à frente no documento;

2) **atualização do perfil dos utilizadores.**

Com base nisto, definiram-se os seguintes requisitos:

- distinção do conhecimento do aluno por domínio;
- distinção do conhecimento do aluno por subdomínio (dentro de cada domínio) e subsubdomínio (dentro de cada subdomínio);
- capacidade de retenção de dados do aluno, tanto gerais, como por domínio, subdomínio e subsubdomínio estudados;
- definição de um nível para caracterizar o aluno, a nível geral, mas também por domínio estudado;
- capacidade de obtenção de informação sobre o tipo de aprendizagem que melhor se adequa ao aluno, a nível geral, mas também por domínio estudado.

A partir deste conjunto de requisitos o módulo começou a ser delineado e estruturado, bem como a base de dados que iria acumular toda a informação necessária para lhe fornecer o devido suporte.

As ações mencionadas anteriormente - criação de perfis de utilizadores e atualização do perfil dos utilizadores - poderão ser vistas de seguida, nas Figura 5 e Figura 7, respetivamente, sob

a forma de diagramas de atividade, para um melhor entendimento da sua mecânica e do algoritmo simplificado por trás destas mesmas ações.

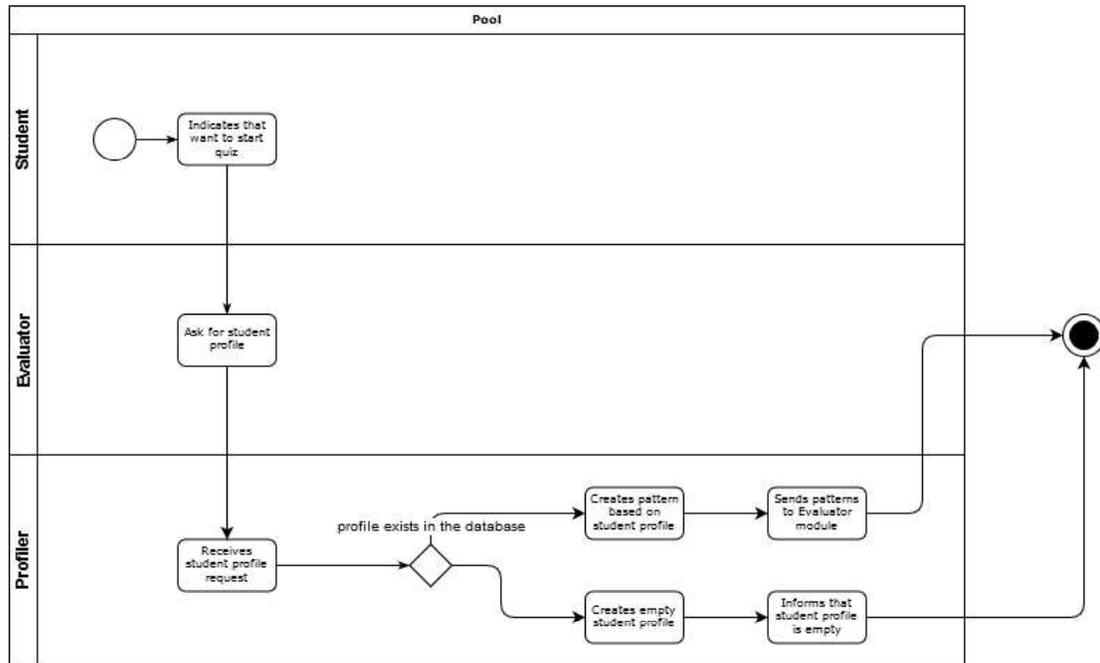


Figura 5 – Diagrama de atividades "Obter Padrão".

No diagrama de atividades apresentado (Figura 5) é possível ver o início de um processo de ensino, no qual um aluno indica que deseja começar a estudar num certo domínio, o *Evaluator* pede, então, ao *Profiler* o padrão base do aluno para aquele domínio - sendo este padrão um documento JSON (Figura 6) -, criado para troca de informação interna no sistema. De seguida o *Profiler* irá averiguar se o aluno em causa existe na base de dados e possui ou não perfil naquele domínio em específico e, em caso de existência do mesmo, cria o mencionado padrão para enviar de volta ao *Evaluator*. No caso de o aluno não ter perfil na base de dados ou para o domínio em estudo, é criado um perfil vazio e enviado um documento JSON, também ele, vazio, terminando o *flow* deste diagrama.

```

{
  "performance"      : "...",
  "skill"            : "...",
  "user_level"       : "...",
  "question_ids_in_session" : "..."
}
    
```

Figura 6 – Estrutura do padrão base de um aluno.

Para o segundo fluxo, apresentado no diagrama de atividades da Figura 7, vê-se o processo de criação e de atualização do perfil do aluno, que é iniciado quando um aluno responde a uma questão. Este processo consiste no *Evaluator* recolher informações provenientes dessa resposta e enviar um pedido de atualização de perfil ao *Profiler*.

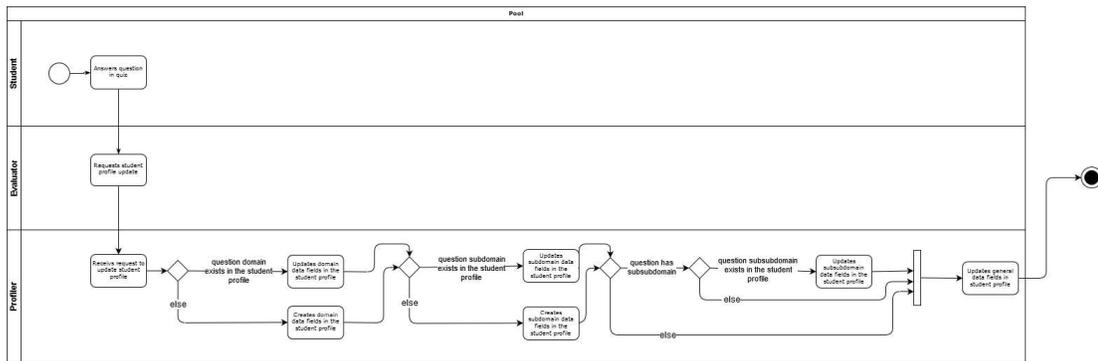


Figura 7 – Diagrama de atividades "Criar e atualizar perfis de utilizadores".

É, então, no *Profiler* que é avaliado se o aluno já tem perfil no domínio da questão respondida e, em caso afirmativo, são atualizados os parâmetros do perfil relativos a esse domínio. O processo repete-se para o subdomínio da questão respondida. Quanto ao subsubdomínio da questão, uma vez sendo um campo facultativo, é efetuada uma verificação com o intuito de averiguar se este campo existe na estrutura da questão respondida, sendo o processo repetido,

analogamente ao acontecido anteriormente relativo ao domínio e subdomínio, no caso do campo existir. Caso contrário, esta parte do processo não é executada e o fluxo segue para a sua última etapa, que se centra em atualizar os dados gerais do aluno com as informações que recebeu.

O diagrama apresentado na Figura 7, constará dos anexos desta dissertação, para poder ser visto num tamanho consideravelmente maior.

3.2 Elementos Base do Modelo

3.2.1 Estruturas Base do Modelo

Começando por uma vertente mais tecnológica, todo o tipo de dados a guardar e tratar neste sistema são documentos JSON, que são guardados numa base de dados NoSQL, orientada a documentos: **MongoDB** (MongoDB, 2019). Esta opção sustentou-se na característica *schemaless* desta base de dados, que permite, como o nome indica, não ter uma estrutura rígida e previamente estabelecida, suportando alterações de documento para documento, do perfil de um aluno para o perfil de um outro aluno.

No que diz respeito à vertente mais funcional, este módulo foi desenhado sobre uma metodologia *ontology-based* (Razmerita, Angehrn and Maedche, 2003), uma vez que reúne um conjunto de atributos para definir o estudante, representando tanto as suas características mais gerais, como outras mais específicas para o(s) domínio(s) em que o mesmo se encontra inserido.

Como base para a organização dos atributos e construção do perfil do estudante esteve a estrutura base da questão do sistema. Esta estrutura apresenta uma complexidade considerável em termos do número de atributos e da sua abrangência. De seguida, apresenta-se os vários atributos que integram a estrutura da questão, bem como a descrição de cada um deles para um entendimento mais simplificado.

- **id** – Identificação do número da questão no domínio
- **language** – Idioma no qual a questão está escrita
- **domain** – Domínio de conhecimento da questão, que se subdivide em 3 campos:
 - **study_cycle** – Descrição do ciclo de estudos em que está inserido o domínio de conhecimento
 - **scholarity** – Descrição da escolaridade do domínio de conhecimento dentro do ciclo de estudos
 - **description** – Descrição concreta do domínio
- **subdomain** – Designação do subdomínio da questão, dentro do domínio da mesma. Especificação mais concreta do tema abordado na questão
- **difficulty_level** – Nível de dificuldade da questão, num ranking de 1 .. 5
- **answering_time** – Tempo máximo disponibilizado para responder à questão
- **header** – Cabeçalho da questão
- **body** – Lista de possíveis respostas para a questão, decomposto em 3 subcampos para cada resposta:
 - **answer** – Corpo da resposta
 - **correction** – Grau de correção da resposta, onde 0 significa uma resposta totalmente errada e 1 uma resposta totalmente correta
 - **mandatory** – Valor indicativo na obrigatoriedade ou falta da mesma desta resposta aparecer como uma das opções apresentadas
- **solution** – Solução da resposta onde é possível descrever o método para a obtenção da mesma
- **source** – Fontes de informação da resposta, quer a nível de fonte de onde uma questão foi retirada, como fontes onde será possível obter informações para ajuda ao estudo
- **notes** – Campo para notas, opcionais, relativas à questão
- **inserted_by** – Identificador do utilizador que inseriu a questão na base de conhecimento, dividido em 2 campos:
 - **id** – Identificador único do utilizador no sistema

- **name** – Nome do utilizador
- **inserted_at** – Data de inserção da questão na base de conhecimento
- **validated_by** – Identificador do utilizador que validou a questão, disponibilizando-a como passível de ser apresentada a alunos. Uma vez mais, campo subdividido em 2 campos:
 - **id** – Identificador único do utilizador no sistema
 - **name** – Nome do utilizador
- **validated_at** – Data de validação da questão e consequente disponibilização no sistema
- **status** – Estado na questão (com dois possíveis estados – V e E – representativos dos estados “Validada e pronta a ser lançada ao aluno” e “Editada e a aguardar validação”, respetivamente)
- **repetitions** – Número máximo de vezes que a questão pode ser apresentada ao mesmo aluno
- **precedence** – Lista de *id's* das questões que devem, estritamente, ser lançadas previamente à questão em causa
- **display_mode** – Processo como as várias respostas serão apresentadas ao aluno (com dois possíveis métodos – “G” ou “I” – representativos de um processo simultâneo, em que todas as respostas são disponibilizadas simultaneamente ao aluno, e iterativo, onde vão aparecendo gradualmente, respetivamente)

Aproveitando a característica da base de dados escolhida, foi possível serem definidos atributos não obrigatórios na estrutura da questão, desta forma, campos como *precedence* ou *notes* podem, simplesmente, não constar da estrutura de uma certa questão.

Utilizando um exemplo prático que poderia constar de uma dada base de conhecimento do sistema para o domínio 'Matemática', do 7º ano de escolaridade do ensino básico, podemos ver na Figura 8 como pode ser definida em JSON uma questão para o sistema.

```

{
  {
    "id": "EB7MAT001",
    "language": "pt",
    "domain": {
      "study_cycle": "Ensino Básico",
      "scholarity": "7º Ano de Escolaridade ",
      "description": "Matemática"
    },
    "subdomain": "Números Racionais",
    "difficulty_level": "2",
    "display_mode": "G",
    "answering_time": "90",
    "header": "Considera a seguinte relação de números: 6, 12, 14,
15, 20 e 28. Quais destes números são múltiplos de 3?",
    "body": [
      {
        "answer": "6, 12 e 15.",
        "correction": 1,
        "mandatory": true
      },
      {
        "answer": "6, 12, 14 e 20.",
        "correction": 0,
        "mandatory": false
      },
      {
        "answer": "Nenhum dos números.",
        "correction": 0,
        "mandatory": false
      },
      {
        "answer": "6, 12 e 18.",
        "correction": 0,
        "mandatory": false
      },
      {
        "answer": "Nenhuma das anteriores.",
        "correction": 0,
        "mandatory": true
      }
    ],
    "source": "Costa, B., Rodrigues, E., Novo Espaço,
Matemática 7º ano, Parte 1, 1ª Edição, Porto Editora,
2018. (p9) (Adaptado)",
    "status": "E",
    "insertedBy": {
      "id": 1,
      "name": "admin"
    },
    "insertedAt": "30-10-2019"
  }
}

```

Figura 8 - Exemplo de uma questão preparada para o domínio 'Matemática'.

Como foi anteriormente referido, com base na estrutura da questão apresentada, foi esboçada a estrutura base do perfil do aluno que era pretendida face às necessidades levantadas nos requisitos, resultando daí o esquema que se encontra apresentado de seguida.

- **username** – identificação do utilizador (aluno) no sistema;
- **user_level** – nível de conhecimento do aluno, numa escala de 1 a 5;
- **total_questions** – número total de questões apresentadas ao aluno;
- **questions_right** – número de respostas certas dadas pelo aluno;
- **questions_wrong** – número de respostas erradas dadas pelo aluno (uma questão não respondida, ou uma questão em que o tempo esgote antes de ser dada uma resposta, é considerada errada);
- **answering_time** – tempo total despendido pelo aluno a responder as questões;
- **total_time** – tempo total disponível para responder ao total de questões apresentadas;
- **performance** – percentagem de acerto do aluno (número de respostas certas / número total de questões apresentadas);
- **skill** – percentagem de eficiência do aluno (tempo de resposta / tempo total disponível para responder às questões apresentadas);
- **profile** – perfil do aluno que consiste numa lista de domínios, com os seus atributos caracterizadores, em que o aluno está inserido, que é constituído por:
 - **domain** – domínio de conhecimento estudado pelo aluno, subdividido em 3 campos, tal como na estrutura da questão apresentada anteriormente, nomeadamente:
 - **study_cycle** – ver estrutura da questão;
 - **scholarity** – ver estrutura da questão;
 - **description** – ver estrutura da questão;
 - **user_level** – nível de conhecimento do aluno dentro do domínio especificado;

- ***q_in_current_level*** – número de questões respondidas no *user_level* atual (atributo importante para os algoritmos de alteração de nível que serão apresentados mais à frente no documento);
- ***answering_time*** – tempo total despendido pelo aluno a responder às questões deste domínio especificado;
- ***total_time*** – tempo total disponível para responder ao total de questões deste domínio especificado;
- ***total_questions*** – número total de questões apresentadas ao aluno sobre este domínio;
- ***hitted*** – número de respostas certas dadas pelo aluno a questões deste domínio;
- ***rbacklog*** – número de questões certas em sequência (uma vez mais, atributo importante para o algoritmo de alteração de nível que será apresentado ainda neste capítulo da dissertação);
- ***wbacklog*** – número de questões erradas em sequência (análogo ao anterior);
- ***skill*** – percentagem de eficiência do aluno no domínio especificado;
- ***subdomains*** – lista de subdomínios no qual é possível subdividir o domínio estudado, é constituído por:
 - ***subdomain*** – identificador do subdomínio dentro do domínio;
 - ***total_questions*** – número total de questões apresentadas ao aluno sobre este subdomínio, dentro do domínio;
 - ***hitted*** – número de respostas certas dadas pelo aluno a questões deste subdomínio, dentro do domínio;
 - ***answering_time*** – tempo total despendido pelo aluno a responder a questões sobre este subdomínio, dentro do domínio;
 - ***total_time*** – tempo total disponível para responder ao total de questões deste subdomínio especificado, dentro do domínio;
- ***last_question*** – identificação da última questão apresentada ao aluno;
- ***last_answer*** – identificação da última resposta dada pelo aluno (certa ou errada);

- **language** – preferência de idioma demonstrado pelo aluno;

Nesta altura é importante salientar alguns aspetos relativos a esta estrutura. Como já foi anteriormente referido, nomeadamente ao nível da questão, o atributo *subdomain* trata-se de uma especificação do domínio no que ao tema que aborda diz respeito. Desta forma, como é possível verificar na estrutura apresentada, os domínios mostram-se mais completos, no que diz respeito ao número de atributos, comparado aos seus subdomínios. Será, ainda assim, possível efetuar uma análise ao nível do subdomínio para um dado aluno, esta apenas constará menos completa em comparação com uma possível análise ao nível de um qualquer domínio.

Importa ainda referir que a grande maioria dos atributos pertencentes ao perfil do aluno irá, de forma independente ou conjunta com outros atributos, ter um papel relevante no processo de alteração de nível de utilizador que será explicado na Secção 3.2.3. Não obstante, atributos como *last_question* ou *last_answer* são, até à data, meramente informativos.

3.2.2 Adição à Proposta de Valor do Modelo

Tendo em consideração a lista de requisitos previamente levantada, esta estrutura especificada para o perfil do aluno, baseada na estrutura definida anteriormente para as questões do sistema, aparentava uma cobertura satisfatória destes mesmos requisitos, no entanto não atingia as exigências da totalidade dos mesmos.

Ao atentarmos à lista de requisitos é possível verificar que, numa primeira fase, foi indicado que deveria existir uma nova divisão de temas, como forma de atingir os temas o mais atómicamente possível, suscitando a criação de um subsubdomínio para cada subdomínio, sem uma estrita obrigatoriedade de existência do mesmo. Com este fim em vista foi necessário a introdução deste atributo, primeiro, ao nível da estrutura da questão, para que a mesma fosse, assim, capaz de suportar esta nova divisão de temas.

No entanto, observando a lista relativa ao levantamento de requisitos, constará que um dos mais importantes requisitos do sistema não se encontra coberto por esta estrutura de perfil de aluno, uma vez que não acompanha as necessidades especiais e características a cada individualidade, como é o tipo de aprendizagem predileto para um determinado aluno.

Uma vez mais, para atingir este fim e cobrir esta importante exigência, foi necessária uma alteração à estrutura da questão que fornecesse o suporte necessário à criação do perfil do aluno capaz de responder às necessidades apresentadas. Tendo por base os estudos de (Niesler and Wydmuch, 2009) e (Morales-Rodríguez *et al.*, 2011) foram, assim, juntamente com o atributo *subsubdomain* mencionado no início desta subsecção, criados três novos atributos para incorporar na estrutura da questão: ***type***, ***videos*** e ***images***.

Ora, estes novos campos podem ser vistos como:

- ***subsubdomain*** – designação do subsubdomínio da questão, dentro do domínio e subdomínio da mesma. Especificação mais atómica do tema abordado na questão;
- ***type*** – definição do tipo da questão, podendo tomar, até ao momento, dois valores possíveis: *Definition* ou *Exercise*;
- ***videos*** – como o próprio nome indica, vídeos dos quais a questão se pode fazer acompanhar;
- ***images*** – uma vez mais o nome fala por si e trata-se de imagens das quais a questão se pode fazer acompanhar;

Os nomes dos atributos, juntamente com a resumida descrição fornecida, falam por si, no entanto importa realçar que estes são os campos que permitirão a criação de atributos no perfil do aluno que possibilitarão a aprendizagem, por parte do sistema, sobre as metodologias e ferramentas preferenciais para o estilo próprio de aprendizagem deste mesmo aluno.

Assim sendo, a estrutura para o perfil do aluno resultante destas adições à estrutura da questão, admitindo a continuidade de todos os atributos apresentados anteriormente, resume-se da seguinte maneira:

- **subsubdomains** – lista de subsubdomínios no qual é possível subdividir o subdomínio estudado, que se organiza em:
 - **total_questions** – número total de questões apresentadas ao aluno sobre este subsubdomínio, dentro do subdomínio;
 - **hitted** – número de respostas certas dadas pelo aluno a questões deste subsubdomínio, dentro do subdomínio;
 - **answering_time** – tempo total despendido pelo aluno a responder a questões deste subsubdomínio;
 - **total_time** – tempo total disponível para responder ao total de questões deste subsubdomínio;
- **hypermedia_images_total** – número total de questões apresentadas ao aluno contendo imagens auxiliares à questão em si;
- **hypermedia_images_hitted** – número de respostas corretas dadas pelo aluno a questões contendo imagens auxiliares à questão em si;
- **hypermedia_videos_total** – número total de questões apresentadas ao aluno contendo vídeos auxiliares à questão em si;
- **hypermedia_videos_hitted** – número de respostas corretas dadas pelo aluno a questões contendo vídeos auxiliares à questão em si;
- **definition_total** – número total de questões apresentadas ao aluno cujo tipo fosse *definition*;
- **definition_hitted** – número de respostas certas dadas pelo aluno a questões cujo tipo fosse *definition*;
- **exercise_total** – número total de questões apresentadas ao aluno cujo tipo fosse *exercise*;
- **exercise_hitted** – número de respostas certas dadas pelo aluno a questões cujo tipo fosse *exercise*;

Como foi mencionado antes, este é o conjunto de atributos que vai permitir que o sistema recolha informações e as processe com o intuito de conhecer e identificar diferentes tipos de processos de aprendizagem dos vários alunos no sistema, possibilitando, desta forma, a passagem de um conjunto de informações mais completo para que outros mecanismos do sistema possam trabalhar sobre eles.

Este conjunto de atributos permite-nos, também, tirar ilações, por exemplo, da hipermédia preferencial de um dado aluno, caso exista uma. É, recorrendo às informações obtidas e guardadas nestes campos, possível distinguir se um aluno se sente mais confortável a responder a questões quando estas recorrem a imagens auxiliares, permitindo uma abordagem mais personalizada na altura da aprendizagem, recorrendo a estas mesmas ferramentas em que o aluno demonstrou facilidade de compreensão e absorção de informação.

Tendo em consideração o estudado no trabalho (Morales-Rodríguez *et al.*, 2011), podemos afirmar que o aluno passa a ser enquadrado segundo o modelo VARK que já foi mencionado no segundo capítulo deste documento.

Com base nestas alterações e nestes novos atributos adicionados, podemos afirmar que o perfil do aluno se encontra, agora, mais robusto e, ao mesmo tempo, consegue suportar as decisões necessárias a tomar pelo sistema para que o seu processo de aprendizagem seja o mais eficiente e fluído possível.

Sendo assim, estamos agora em condições de visualizar um exemplo de como poderia ficar um perfil de um qualquer aluno na base de dados do sistema (Figura 9):

```
{
  "username" : "luismfernandes",
  "user_level" : 3,
  "total_questions" : 58,
  "questions_right" : 51,
  "questions_wrong" : 7,
  "performance" : 0.879,
  "skill" : 0.750,
  "hypermedia_images_total": 23
  "hypermedia_images_hitted": 19
  "hypermedia_videos_total": 19
  "hypermedia_videos_hitted": 18
  "definition_total": 28
  "definition_hitted": 22
  "exercise_total": 30
  "exercise_hitted": 29
  "profile" : [
    {
      "domain" : {
        "study_cycle" : "Ensino Superior",
        "scholarity" : "Engenharia Informática",
        "description" : "Sistemas de Bases de Dados"
      },
      "user_level" : 3,
      "q_in_current_level" : 58,
      "answers_time" : 33419.231431,
      "rbacklog" : 0,
      "wbacklog" : 3,
      "hitted" : 51,
      "total" : 58,
      "skill" : 0.750
      "hypermedia_images_total": 23
      "hypermedia_images_hitted": 19
      "hypermedia_videos_total": 19
      "hypermedia_videos_hitted": 18
      "definition_total": 28
      "definition_hitted": 22
    }
  ]
}
```

```
"exercise_total": 30
"exercise_hitted": 29
"subdomains": [
  {
    "subdomain" : "Bases de dados relacionais",
    "hitted" : 0,
    "total" : 3,
    "skill" : 0.750
    "subsubdomains": [
      {
        "subsubdomain" : "Modelo Lógico",
        "hitted" : 0,
        "total" : 3,
        "skill" : 0.750
      }
    ]
  },
]
},
"session_questions_ids" : [
  "UKEINSBDO156",
  "UKEINSBDO157",
  "UKEINSBDO154"
],
"sessions_time" : 0,
"questions_time" : 0,
"answers_time" : 33419.231431,
"last_question" : ...
},
"last_answer" : "wrong",
"last_difficulty_level" : 3
"language" : "pt"
}
```

Figura 9 - Exemplo de um possível perfil de um aluno no sistema.

3.2.3 Algoritmo de Alteração de Nível de Utilizador

O enquadramento do aluno num nível de conhecimento é parte integrante e essencial para o seu processo de *profiling*. Este enquadramento, como já foi referido, é efetuado a um nível geral, mas também a um nível mais concreto, ou seja, ao nível dos vários domínios.

Na estrutura da questão é possível verificar a existência do atributo *difficulty_level*. Ora, este atributo serve como mapeamento para o *user_level*, atributo existente no perfil do aluno e em foco nesta secção do documento. Apesar da não obrigatoriedade de que a um aluno com *user_level* X seja apresentada uma questão de *difficulty_level* X, este acaba por ser o caso mais provável de acontecer ao longo da interação existente entre o aluno e o sistema, sendo que os casos excecionais, controlados pelo *Evaluator*, se baseiam em duas vertentes:

- 1) quando não existem mais perguntas disponíveis na base de conhecimento para um certo nível e o sistema se vê obrigado a lançar uma questão de nível inferior ou superior;
- 2) quando se aplica a regra de *backlog*, que será apresentada ainda nesta secção do capítulo.

Assim sendo, o atributo *user_level* mostra-se essencial em toda a mecânica do processo de avaliação de um aluno e como tal foi necessário definir um conjunto bem estruturado de regras que contemplassem as possíveis variações neste mesmo atributo.

Começemos então por olhar para uma estrutura, também ela preponderante, para este processo de alteração de nível de utilizador, a configuração de um domínio. Os campos desta estrutura são os seguintes:

- ***id*** – identificador único do domínio no sistema;
- ***study_cycle*** – descrição do ciclo de estudos em que se encontra inserido o domínio de conhecimento;

- ***scholarity*** – descrição da escolaridade do domínio de conhecimento dentro do ciclo de estudos;
- ***description*** – descrição concreta do domínio;
- ***subdomains*** – lista de subdomínios associados ao domínio, em que cada um deles é composto por:
 - ***id*** – identificador derivado do *id* do domínio em causa (se domínio tem *id* “1”, então o subdomínio terá um *id* “1.X”);
 - ***description*** – descrição concreta do subdomínio;
 - ***subsubdomains*** – lista de subsubdomínios associados ao subdomínio, sendo estes compostos por 2 campos:
 - ***id*** – identificador derivado do *id* do subdomínio em causa (se domínio tem *id* “1” e subdomínio tem *id* “1.3”, então o subsubdomínio terá “1.3.Y”);
 - ***description*** – descrição concreta do subsubdomínio;
- ***config*** – configuração do domínio, subdividido em 8 campos, nomeadamente:
 - ***default_user_level*** – nível de utilizador pré-definido;
 - ***high_performance_factor*** – valor de referência do desempenho do aluno para intervenção no processo de subida de nível;
 - ***low_performance_factor*** – análogo ao anterior, mas para intervenção no processo de descida de nível;
 - ***high_skill_factor*** – valor de referência da aptidão do aluno, para intervenção no processo de subida de nível;
 - ***low_skill_factor*** – análogo ao anterior, mas para intervenção no processo de descida de nível;
 - ***min_questions_number*** – número mínimo predefinido de questões necessárias para um aluno ficar habilitado a subir ou a descer de nível;

- **questions_factor** – valor que age diretamente com o atributo *min_questions_number*, sendo o fator multiplicativo para que com a subida de nível, o número de questões necessárias para um aluno ficar habilitado a subir ou a descer de nível seja também superior. No caso de este fator ser 0, então o número de questões necessárias para um aluno ficar habilitado a ver o seu nível alterado, em qualquer nível, será sempre igual a *min_questions_number*,
- **backlog_factor** – valor que se relaciona diretamente com o *user_level* atual do aluno, uma vez que define a quantidade de perguntas acertadas ou erradas em sequência para que o aluno automaticamente suba ou desça de nível. Esta incorporação de um fator de *backlog* será abordado ainda nesta subsecção.

De seguida, na Figura 10, apresenta-se um caso prático de aplicação desta mesma estrutura, na qual se pode ver um exemplo de uma possível configuração de um domínio que poderia, se desejado, ser parte integrante da base de conhecimento do sistema.

```
{
  "id": "1",
  "study_cycle": "Ensino Superior",
  "scholarity": "Engenharia Informática"
  "description": "Sistemas de Bases de Dados"
  "subdomains": [
    {
      "id": "1.1",
      "description": "Bases de dados relacionais"
      "subsubdomains": [
        {
          "id": "1.1.1",
          "description": "Modelo Lógico"
        }
      ]
    }
    {
      "id": "1.2",
      "description": "Bases de dados não relacionais"
    }
  ]
  "config": {
    "default_user_level": 2,
    "high_performance_factor": 0.75,
    "low_performance_factor": 0.35,
    "high_skill_factor": 0.60,
    "low_skill_factor": 0.20,
    "min_questions_number": 5,
    "questions_factor": 2,
    "backlog_factor": 4
  }
}
```

Figura 10 - Exemplo de uma possível configuração de domínio.

Ora, como já foi mencionado antes, as configurações dos vários domínios no sistema são parte fulcral para o seu bom funcionamento e desempenho e apresentam um papel extremamente relevante no processo de alteração do nível dos utilizadores (alunos).

Para além dos atributos que definem e descrevem os domínios, subdomínios e subsubdomínios, e que são, obviamente, importantes, aqui o foco recairá no atributo *config* e seus subatributos, uma vez que, estes sim, integram de forma ativa o processo em estudo nesta subsecção do documento.

Assim sendo, comecemos por explicar a regra e mecanismo de *backlog* já mencionada. Este mecanismo foi introduzido no sistema como forma de premiar alunos com excelentes desempenhos no presente, para contrabalançar um desempenho menos conseguido que possam ter obtido num passado mais distante. Analogamente, este mecanismo irá também afetar alunos com desempenhos negativos, apesar de excelentes resultados que possam ter sido obtidos no passado.

Vejamus um pouco da orgânica deste mecanismo:

- um qualquer aluno num qualquer *user_level* poderá subir de nível assim que acertar, em sequência, em $backlog_factor * user_level$ respostas;
- da mesma forma, um qualquer aluno num qualquer *user_level* poderá descer de nível assim que errar, em sequência, em $backlog_factor * user_level$ respostas.

No entanto, e apesar do atributo *backlog_factor* poder ser tão grande quanto o *expert* do domínio o desejar, este mecanismo pode parecer um pouco drástico, uma vez que faz uma alteração do nível de utilizador apenas baseado nesta sequência de respostas, sem efetuar análise a outros atributos do perfil do aluno.

Como forma de combater este radicalismo no processo de alteração de nível de utilizador, foi introduzido sobre este mecanismo a, anteriormente definida, **regra de *backlog***.

Esta regra de *backlog* vem de encontro ao analisado no estudo (Chopra *et al.*, 2010) que refere que “uma vez que o sistema é desenhado para humanos, é seguro assumir que nenhum dos atributos do estudante deva sofrer uma drástica alteração num curto período de tempo” e, assim sendo, sequências fortuitas de acerto ou, do mesmo modo, sequências desastradas de erro devem ser postas à prova e não devem ser simplesmente aceites pelo sistema.

Apesar desta regra não ser implementada por este módulo, mas sim pelo *Evaluator*, é importante que seja referida aqui, uma vez que integra de forma importante o modo como o sistema, e o *Profiler* especificamente, denota as alterações ao nível dos utilizadores.

Assim, o processo desta regra consiste em o *Evaluator* detetar que o aluno atingiu o *threshold*, internamente definido como 1, em relação ao *backlog*. Isto significa que o aluno está, nesse momento, a 1 resposta, certa ou errada, de subir ou descer de nível, segundo o mecanismo de *backlog*. De seguida, o *Evaluator* vai lançar uma questão do nível imediatamente acima ou abaixo, como forma de testar se o aluno se encontra realmente preparado ou necessitado de subir ou descer de nível.

Obviamente este mecanismo não é ótimo, uma vez que, se pensarmos, por exemplo, num caso de um aluno estar a executar uma avaliação com um *expert* do domínio ao seu lado a fornecer apoio, o mesmo vai, certamente, conseguir fornecer o mesmo apoio a uma pergunta de nível superior e o aluno acabará por superar a regra de *backlog* e seguir para o nível seguinte, no entanto apresenta um desafio extra a esta alteração.

Uma vez esclarecido o modo como o mecanismo e regra de *backlog* foram integrados, surge agora o momento de expor o algoritmo de alteração de nível do utilizador e passar à sua explicação detalhada.

Chegado o momento da entrada em ação do algoritmo de alteração de nível, existem 3 cenários possíveis:

- o aluno encontra-se no *user_level* X e, após o algoritmo, passa a estar no *user_level* $X+1$;
- o aluno encontra-se no *user_level* X e, após o algoritmo, permanece no *user_level* X ;
- o aluno encontra-se no *user_level* X e, após o algoritmo, passa a estar no *user_level* $X-1$.

Posto isto, assumindo que seja:

- *q_in_current_level* o número de questões que o aluno efetuou no *user_level* em que se encontra;

- *r_questions* o valor proveniente do cálculo $min_questions_number + (questions_factor * user_level)$;
- *performance* o valor presente no perfil do aluno que denota a percentagem de acerto do mesmo;
- *skill* o valor presente no perfil do aluno que denota a eficiência ao responder às questões apresentada pelo mesmo;
- *rbacklog* o número de respostas certas dadas pelo aluno em sequência;
- *wbacklog* o número de respostas erradas dadas pelo aluno em sequência;
- *backlog* o valor proveniente do cálculo $backlog_factor * user_level$;
- *high_performance_level* o valor proveniente do cálculo $high_performance_factor + (user_level * 0.04)$;
- *low_performance_level* o valor proveniente do cálculo $low_performance_factor + (user_level * 0.04)$;
- *high_skill_level* o valor proveniente do cálculo $high_skill_factor + (user_level * 0.03)$;
- *low_skill_level* o valor proveniente do cálculo $low_skill_factor + (user_level * 0.03)$;

O algoritmo de avaliação pode ser expresso da forma como está apresentado na Figura 11:

```
IF
  user_level < 5
  IF
    (q_in_current_level >= r_questions AND
     performance >= high_performance_level AND
     skill >= high_skill_level) OR
    (rbacklog >= backlog_factor)
  IF
    user_level > 1
    IF
      (q_in_current_level >= r_questions AND
       performance <= low_performance_level AND
       skill <= low_skill_level) OR
      (wbacklog >= backlog_factor)
```

Figura 11 – Excerto do algoritmo de alteração de nível de utilizador em pseudocódigo.

Passemos agora à escrutinação concreta do algoritmo.

Em primeiro lugar, é avaliado se o *user_level* é inferior a 5, uma vez que este é o nível máximo que este atributo pode tomar e, como tal, uma vez neste nível, não é possível subir mais, o que em termos de código representa a não possibilidade de executar o pedaço de código relativo à tentativa de aumentar o nível de utilizador.

Uma vez que este teste é ultrapassado e o *user_level* é menor que 5, são testados 2 casos distintos:

- o primeiro teste consiste na subida de nível pelo modo conceptualmente regular, sendo aqui verificado se o utilizador já respondeu a um número maior ou igual de questões necessárias estipuladas na configuração do domínio, para o seu *user_level*, que possibilite a alteração de nível, se a sua *performance* é superior ou igual ao valor de referência para o seu *user_level* e ainda se a sua *skill* é superior ou igual ao valor de referência para o seu *user_level*; quando as 3 comparações efetuadas são

verdadeiras em simultâneo, dá-se então o aumento de nível do utilizador para o imediatamente ao seu, entrando, assim, no primeiro dos 3 cenários apresentados anteriormente; no caso de uma ou mais destas comparações não ser verdadeira, é verificado o segundo teste;

- o segundo teste compreende o mecanismo de *backlog* já bastante discutido nesta subsecção; é verificado se o número de respostas certas em sequência dadas pelo aluno é igual ou superior ao valor estipulado para o seu *user_level*; novamente, em caso afirmativo, dá-se a subida do nível do aluno, uma vez mais entrando no primeiro cenário apresentado.

Segue-se então a segunda instância deste algoritmo, onde é verificado se o *user_level* é superior a 1, mais uma vez isto acontece visto que 1 é o valor mínimo possível para este atributo e como tal, uma vez aqui não mais é possível baixar de nível.

Este teste, sendo ultrapassado, denota a possibilidade de reduzir o nível do utilizador e para isso são necessárias as comparações, análogas às do primeiro caso, mas em sentido inverso:

- o primeiro teste consiste na descida de nível, também ele, pelo modo regular, sendo que é necessário para isso que o utilizador cumpra as 3 comparações efetuadas, em simultâneo; primeiro é necessário que o utilizador tenha respondido a um número maior ou igual de questões necessárias estipuladas na configuração do domínio, para o seu *user_level*, que possibilite a alteração de nível; em simultâneo, o utilizador indesejavelmente teria de possuir um valor de *performance* igual ou inferior ao valor de referência para o seu *user_level*, e, ainda, de possuir uma *skill* igual ou inferior ao valor de referência para a *skill* no seu *user_level*; como mencionado, uma vez cumpridas as 3 comparações em simultâneo, o nível do utilizador é baixado para o imediatamente abaixo, dando entrada no terceiro cenário possível entre os apresentados; por outro lado, caso uma ou mais destas comparações não se verifique, prossegue-se para o segundo teste;

- o segundo teste, naturalmente, compreende também ele o mecanismo de *backlog*, aqui é verificado se o aluno possui uma sequência de respostas erradas igual ou superior ao valor de *backlog* referenciado para o seu *user_level*, o que, em caso afirmativo, se materializará na respectiva descida de nível de utilizador para o imediatamente abaixo, uma vez mais, entrando no terceiro cenário dos apresentados.

Ultrapassada esta segunda etapa do algoritmo, caso o aluno não tenha sofrido alterações no seu *user_level* o segundo cenário prevaleceu e este aluno irá, pelo menos até responder a uma nova questão, permanecer no nível em que encontra nesse momento, estando, desta forma, todos os cenários possíveis cobertos.

Capítulo 4

Um Caso de Aplicação

4.1 Leonardo

Nascido na Universidade do Minho, proveniente de uma iniciativa interna de I&D – Investigação & Desenvolvimento, o sistema Leonardo tem como objetivo colmatar algumas deficiências no ensino tradicional, procurando mostrar-se como uma plataforma educacional online orientada especialmente para suportar processos de avaliação (Belo, Coelho and Fernandes, 2019). A iniciativa despontou com o propósito de ser integrado como ferramenta preponderante no suporte ao processo de aprendizagem e avaliação, inicialmente, da unidade curricular de Base de Dados, do 3º ano do curso Mestrado Integrado em Engenharia Informática.

Complementando, as principais motivações para a criação deste projeto basearam-se na ausência de um sistema capaz de complementar a formação e avaliação no contexto do ensino académico, fornecendo um suporte e *feedback* personalizado ao utilizador (aluno), durante a execução de tarefas, sem a intervenção de seres humanos.

O Leonardo apresenta-se como uma plataforma *web-based*, que nesta fase ainda embrionária de desenvolvimento, garante, entre outras, funcionalidades como o registo de utilizadores, criação e

manutenção de domínios de conhecimento na base de conhecimento do sistema ou ainda gestão de dados para análise.

Podemos afirmar que os principais objetivos do Leonardo passam por ser um tutor inteligente capaz de se adaptar aos perfis de conhecimento revelados pelos alunos num dado conteúdo ao longo da interação que vão tendo com o sistema, no processo designado de *profiling*; implementar uma avaliação apoiada deste conhecimento apreendido do aluno, recomendando elementos de estudo de acordo com fragilidades que possam ser reveladas por estes mesmos alunos e, a um nível mais técnico, ser uma ferramenta de estudo disponível 24h (Marques and Belo, 2010) e acessível a partir de qualquer plataforma computacional com acesso à internet (*cross-platform*).

4.2 Arquitetura do Sistema

A arquitetura do sistema Leonardo não denota diferenças significativas para aquilo que é utilizado comumente na definição de um ITS. Na Figura 12, é possível visualizar a arquitetura funcional do sistema Leonardo, onde, desde logo, se destacam três módulos principais: Módulo de Edição (*Edition Mode*), Módulo da Base de Conhecimento (*Knowledge Base Mode*) e Módulo de Raciocínio (*Reasoning Mode*), da esquerda para a direita.

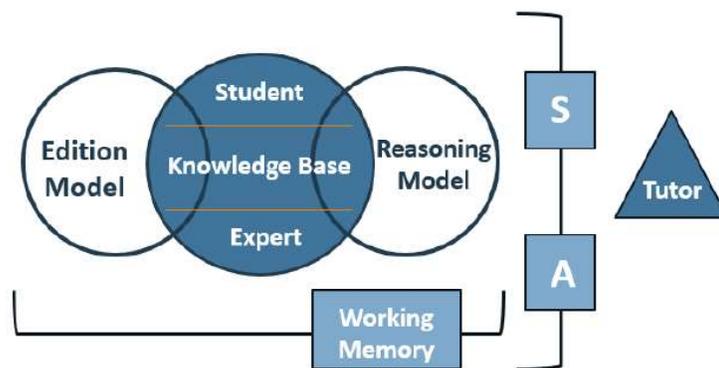


Figura 12 - Arquitetura do sistema Leonardo.

Ora, detalhando mais pormenorizadamente o conteúdo desta figura representativa da arquitetura do Leonardo, temos, como ponto central o Módulo da Base de Conhecimento. Este módulo é o ponto fulcral deste projeto, uma vez que é aqui, que toda a informação do sistema é armazenada, alimentado pelos dois agentes lá ao lado apresentados: *Student* (aluno) e *Expert* (utilizador qualificado).

Ao lado, e em contínua ligação, com este módulo anterior, encontra-se o módulo de Edição. É aqui que os *experts* poderão passar informação diretamente para a base de conhecimento, incluindo definição de domínios e configurações dos mesmos ou exercícios e questões, devidamente acompanhados de material de estudo apropriado ou referências para o mesmo, que possam ser parte integrante de um destes domínios.

Do outro lado encontra-se o Modelo de Raciocínio (*Reasoning Model*), ou como já foi denominado ao longo deste documento, o *Evaluator*. Este módulo funciona com um motor de inferência baseado em regras de produção. Este motor de inferência trabalha sobre a base de conhecimento, por exemplo aproveitando o mecanismo de *profiling*, que foi trabalhado nesta dissertação, adquirindo a capacidade de determinar o estado do conhecimento do aluno num dado momento e adaptar o lançamento da próxima questão.

É ainda possível visualizar uma *Working Memory* que, como é comum, simboliza o comportamento do sistema em atividade, podendo, por exemplo, ser carregado para memória o perfil do aluno conseguindo, desta forma, acesso mais facilitado e rápido a informação necessária para a tomada de decisões.

Ainda na arquitetura, encontram-se definidas umas formas quadrangulares com um S e um A que representam, respetivamente, os Sensores e Atuadores do ITS na interação com o utilizador, estando intrinsecamente relacionados com o a interface desenvolvida para o utilizador.

Importa ainda referir que na fase de definição da arquitetura do Leonardo, ainda não era conhecida, e, portanto, não ficou esquematizada, o modo como o *Student Model*, denominado *Profiler* e estudado ao longo do documento, integraria no ITS. Posto isto, é-nos agora possível desenhar uma arquitetura que inclua este módulo:

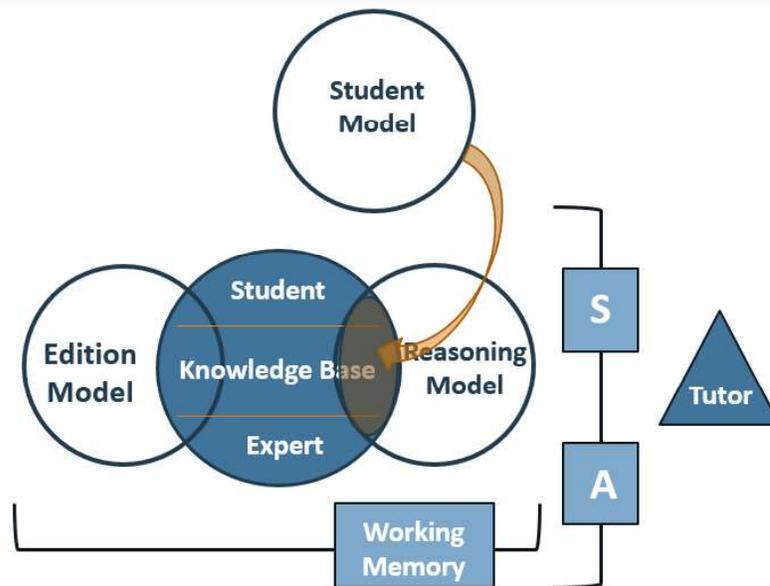


Figura 13 - Arquitetura do sistema Leonardo, contemplando o *Student Model*.

Podemos afirmar que o *Profiler – Student Model* na Figura 13 – servirá como módulo de ligação e suporte e atuará, principalmente nas zonas de união dos vários módulos que possam necessitar de informações que o *Profiler* recolhe e trata. Este módulo é, como foi discutido ao longo desta dissertação, o principal responsável pela construção do perfil de um aluno (*profiling*), tendo influência e ligação direta com o *Reasoning Model (Evaluator)* no processo de avaliação que já foi abordado, mas ainda será descrito detalhadamente um pouco mais à frente.

4.3 Ferramentas utilizadas

Uma vez definido e modelado o sistema, o passo lógico que se seguiu consistiu na escolha das ferramentas e *stack* tecnológica a utilizar.

Tendo como assumido que a principal “arma” do sistema são os dados e o modo como este é capaz de os gerir e manipular, a primeira escolha a ser efetuada recaiu no sistema de base de dados. Neste ponto em específico, a necessidade de manipular grandes volumes de dados e processá-los de forma rápida e eficaz, aliando a uma necessária capacidade de *performance* e escalabilidade, levou a uma escolha relativamente óbvia: um sistema de gestão de bases de dados não relacional. Aqui, nas várias opções disponíveis para a nossa busca já filtrada, a escolha recaiu sobre MongoDB (MongoDB, 2019). Uma vez mais, esta opção pode ser justificada pelo facto de que a MongoDB se apresenta como uma base de dados orientada a documentos, sem um esquema rígido – *schemaless*. O acesso rápido a dados para uma atempada resposta do tutor é crucial e uma base de dados orientada a documentos oferece a vantagem de um documento poder conter diversas informações, distintas de um documento para outro dentro da mesma coleção, apenas em troca de alguma redundância, evitando os cruzamentos e junções de dados de diferentes tabelas, como acontece, por exemplo, no mundo das bases de dados relacionais.

Ditada esta sentença, algumas decisões tiveram que ser tomadas, uma vez que eram necessárias para que o sistema a ser desenvolvido pudesse realmente corresponder aos objetivos delineados de consistir numa plataforma acessível a partir de qualquer lado, em qualquer altura. Assim sendo, a escolha recaiu sobre uma *webapp*, capaz de, por exemplo, se adaptar aos vários tamanhos de ecrãs que os dispositivos podem possuir (*responsive*) e estando, desta forma, disponível 24h através de qualquer sistema computacional com acesso à Internet.

O próximo e o último passo no que à escolha das ferramentas diz respeito, consistiu na escolha da *framework web* a utilizar. Esta decisão não apresentou grande ciência devido ao facto de que a escolha da linguagem de programação para o desenvolvimento foi *Python* (Python.org, 2019), muito devido às suas imensas bibliotecas de análise de dados, e por consequente, a *framework* escolhida foi *Flask* (Pallets Projects, 2019), porque, em comparação com outras como *Django* (Django Project, 2019), permite um maior controlo sobre os componentes a usar no desenvolvimento deste sistema.

4.4 Enquadramento do Modelo de Avaliação

Neste ponto, e para finalizar o capítulo e a dissertação, passaremos para a descrição e explicação da componente de avaliação do Leonardo, o principal processo dentro deste sistema, demonstrando a integração do modelo desenhado e desenvolvido ao longo deste documento na sua orgânica.

Numa primeira instância, e antes que qualquer processo de avaliação possa ocorrer, é necessário que um Administrador do sistema crie o(s) domínio(s) que pretende ver disponível(eis) e atribuir um responsável a cada um dos domínios de conhecimento que criou. São estes responsáveis que se encontrarão encarregues de configurar o domínio, bem como especificar o conhecimento do mesmo, subdividindo-o em subdomínios e subsubdomínios. Além disto, é necessário que este Responsável, ou um Operador nomeado por este, crie as questões que pretende ver inseridas na base de conhecimento relativas ao domínio pelo qual está responsabilizado. É necessário ainda, numa fase posterior, que os alunos visados para ter acesso a este domínio sejam, também eles, introduzidos, tarefa que, uma vez mais, poderá ser realizada com o apoio de um ou mais operadores de domínio nomeados pelo responsável.

Na inserção de questões intervém o Modelo de Edição, em que, através de uma interface gráfica apelativa, o responsável ou os operadores poderão criar as questões com toda a informação necessária para preencher o corpo das mesmas. Dentro destes campos da questão, como já foi referido aquando o estudo da estrutura da questão, alguns possuem um carácter obrigatório, enquanto outros são opcionais. Estas questões deverão ser o mais completas possível, na mesma medida que devem ser abrangentes ao ponto de cobrir os vários subdomínios e subsubdomínios, bem como os vários níveis de dificuldade existentes. Para que estas questões fiquem disponíveis no sistema (*status* = 'V'), as mesmas devem passar por um processo de validação. Assim que inseridas as questões permanecem em estado 'E' – editada e à espera de validação – até que o seu responsável as valide. Este processo é essencial quando o responsável delega a criação de

questões para um ou mais operadores, no entanto, no caso de ser o próprio responsável pelo domínio a inserir as questões, estas obtêm imediatamente o *status* 'V' e ficam imediatamente armazenadas na base de conhecimento como disponíveis para serem apresentadas posteriormente ao aluno nos processos de avaliação que solicitar.

Chegando ao processo de avaliação propriamente dito, o mesmo é realizado segundo um conceito de *quiz*, devido ao seu carácter simples e objetivo num formato *multiple-choice/single-answer*, em que apenas uma opção de resposta poderá ser selecionada entre as várias possíveis respostas apresentadas.

Numa primeira interação entre um dado aluno e o sistema, este encontra-se sem um perfil traçado, pelo que nada poderá ser passado pelo *Profiler*. Desta forma o sistema, por intermédio do *Evaluator*, irá recorrer à configuração base do domínio (por exemplo ao capo *default_user_level*) para ponderar em que contexto enquadrará o aluno e que questão poderá lançar naquele dado momento. A partir deste momento, toda e qualquer interação homem-máquina trará informação para o *Profiler* recolher, tratar e armazenar, como forma de delinear o perfil do aluno.

Esta integração entre o *Profiler* e o *Evaluator* é o principal mecanismo do sistema Leonardo, que aglomera a capacidade de criação e reconhecimento de perfis por parte do *Profiler*, com a capacidade de decisão segundo certas regras do *Evaluator*.

Como peça fundamental desta integração encontra-se a API (Figura 14) disponibilizada pelo módulo em análise neste documento, o *Profiler*, que permite ao *Evaluator* ter acesso às informações necessárias sobre o aluno para que possa tomar as decisões de forma fundamentada.

```
get_pattern ( user = aluno, domain = None, subdomain = None, subsubdomain = None )
```

- > se não forem passados argumentos, é retornado o padrão do perfil geral do utilizador;
- > se domain = x, então é retornado o padrão do perfil do utilizador no domínio x;
- > se domain = x e subdomain = y, então é retornado o padrão do perfil do utilizador no subdomínio y, dentro do domínio x;
- > se domain = x, subdomain = y e subsubdomain = z, então é retornado o padrão do perfil do utilizador no subsubdomínio z, dentro do subdomínio y, dentro do domínio x.

```
get_decision_pattern ( user = aluno, domain = None, subdomain = None )
```

- > se não forem passados argumentos, é retornado uma lista com os domínios no qual o aluno está inserido, com os respetivos valores de performances e skill;
- > se domain = x, é retornada uma lista com os subdomínios no qual o aluno está inserido, dentro do domínio x, com os respetivos valores de performance e skill;
- > se domain = x e subdomain = y, é retornada uma lista com os subsubdomínios no qual o aluno está inserido, dentro do subdomínio y, dentro do domínio x, com os respetivos valores de performance e skill;

```
update_profile ( user = aluno, question = questão, answer = resposta,  
                time = tempo_resposta )
```

- > Efetua a atualização do perfil do aluno

Figura 14 – A API do *Profiler*.

Na Figura 14 podemos ver que existem 3 métodos passíveis de serem chamados: ***get_pattern(...)***, ***get_decision_pattern(...)*** e ***update_profile(...)***. Estes métodos, com a sua devida explicação na figura, oferecem a ponte de ligação entre o *Evaluator* e o *Profiler*, crucial para

o processo de avaliação em análise, efetuada através da chamada dos métodos mencionados e do retorno de objetos JSON, formato definido para a comunicação interna do sistema.

Continuando no processo de avaliação, após a primeira interação do aluno com o sistema, o mesmo já apresentará um perfil na base de conhecimento, desta forma, para a próxima iteração, diga-se, lançamento da próxima questão, o *Evaluator* poderá pedir, através da API disponibilizada, o perfil do aluno em questão e receber os dados do mesmo para, assim, decidir a questão a ser lançada com base num conjunto de regras específico.

Assim que o aluno responde a uma questão, a informação é passada para o *Profiler* através do método ***update_profile(...)***, sendo que este tratará dos dados recebidos e da informação gerada e armazenará os mesmos de acordo com todos os aspetos e critérios que foram estudados e abordados ao longo desta dissertação, mantendo o perfil do aluno sempre o mais atualizado e fiel possível ao comportamento e, principalmente, conhecimento do mesmo.

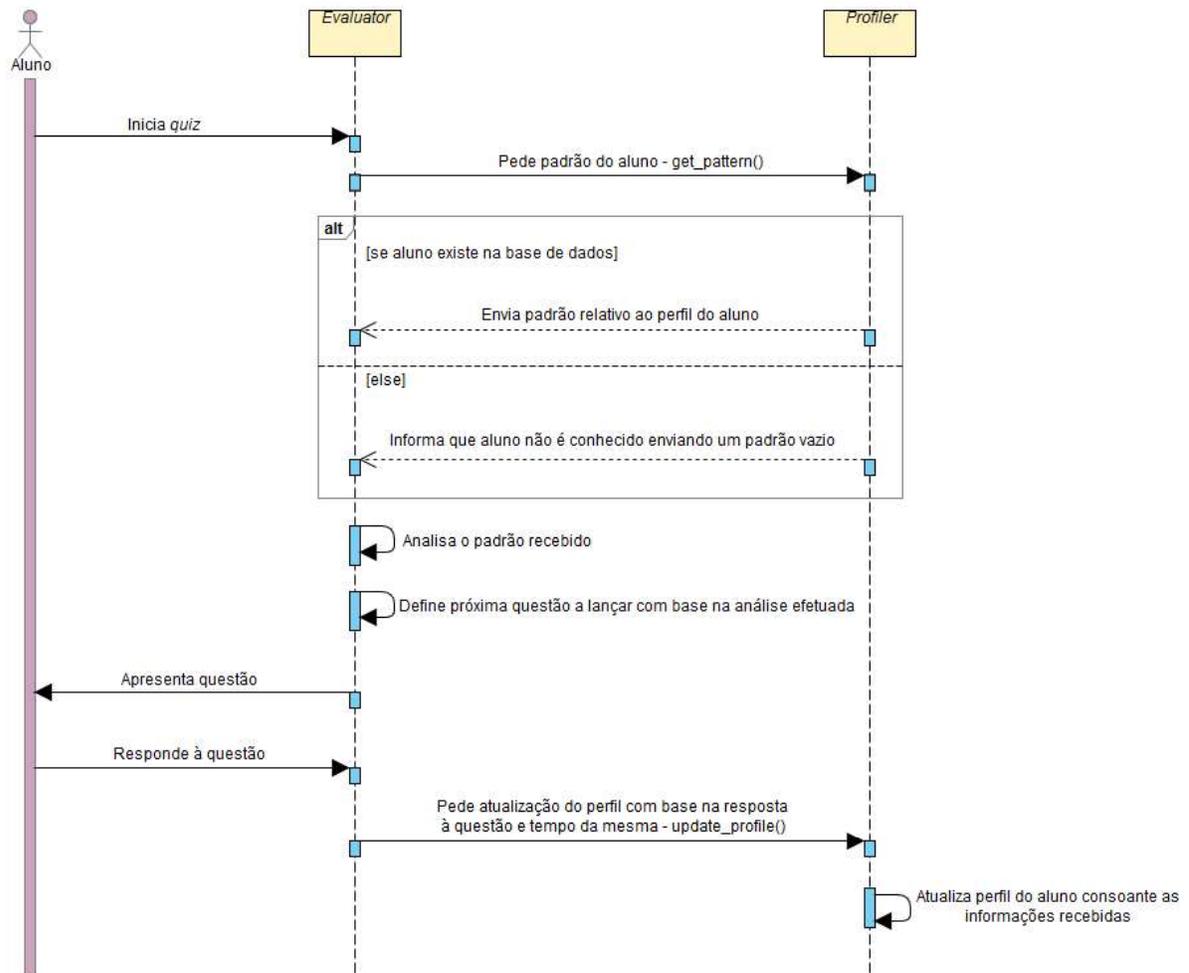


Figura 15 - Diagrama de Sequência relativo ao processo de avaliação.

Resta-nos agora, durante o resto do capítulo, visualizar a interface gráfica que possibilita todo este fluxo, bem como a interação com o utilizador, seja este o *expert* que pretende alimentar a base de conhecimento, ou o *aluno*, que pretende seguir com o seu processo apoiado de aprendizagem.

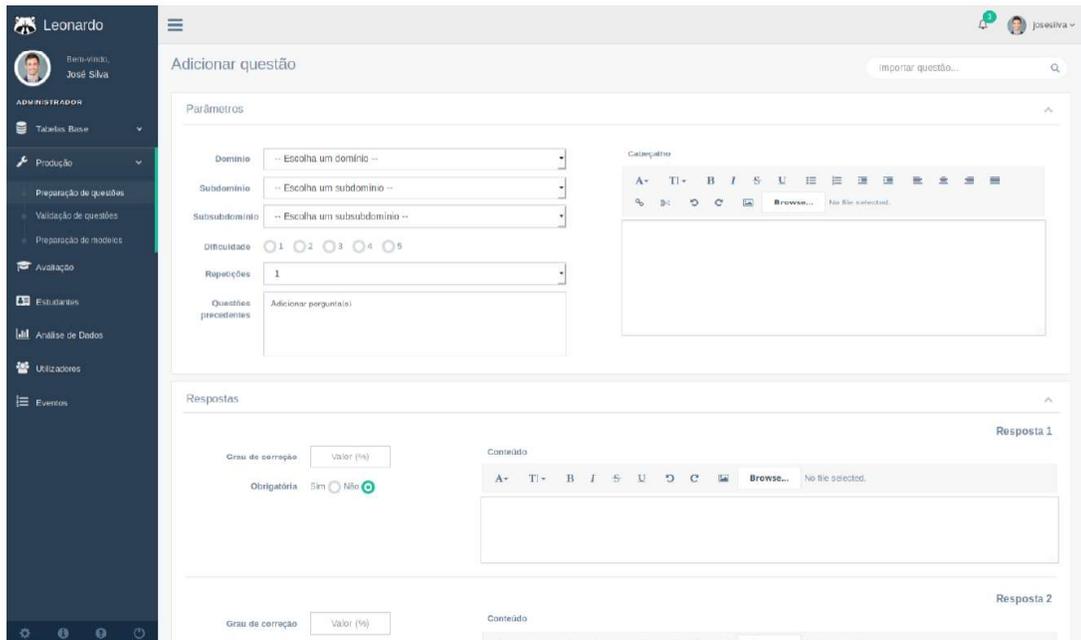


Figura 16 - Interface do modelo de edição de questões de *quiz*.

Como referido, é através da interface apresentada na Figura 16 que os *experts* podem criar questões para alimentar a base de conhecimento do sistema, passo essencial para o bom funcionamento do mesmo. Salientar que este *screenshot* foi tirado na plataforma de gestão do Leonardo, onde apenas administradores, responsáveis e operadores têm acesso. Será nesta plataforma que, mais tarde, se poderá visualizar a *dashboard* com os dados relativos ao perfil de um aluno. As questões, após editadas, são armazenadas na base de dados do sistema em formato JSON, tal como se pode ver na Figura 17.

```
{
  "id": "PTEINSBD0150",
  "language": "pt",
  "domain": {
    "study_cycle": "Ensino Superior",
    "scholarity": "Engenharia Informática",
    "description": "Sistemas de Bases de Dados"
  },
  "subdomain": "SQL",
  "subsubdomain": "",
  "difficulty_level": "2",
  "display_mode": "G",
  "answering_time": "60",
  "type": "definition",
  "precedence": [],
  "repetitions": "0",
  "header": "A SQL é uma linguagem utilizada em programação concebida
    especificamente para a descrição, manipulação e controlo de dados
    num sistema de bases de dados relacional.
    Esta afirmação é:",
  "body": [
    {
      "answer": "Verdadeira.",
      "correction": 1,
      "mandatory": true
    },
    {
      "answer": "Falsa.",
      "correction": 0,
      "mandatory": true
    }
  ],
  "solution": "",
  "images": "",
  "videos": "",
  "source": "Belo, O., Cadernos de Problemas - Sistemas de Bases de Dados,
    Departamento de Informática, Escola de Engenharia,
    Universidade do Minho, 2019",
  "notes": "Para utilização exclusiva da plataforma educacional Leonardo",
  "status": "E",
  "inserted_by": "Belo, O",
  "inserted_at": "2019-09-12",
  "validated_by": "",
  "validated_at": ""
}
```

Figura 17 - Exemplo de uma questão do sistema em JSON.

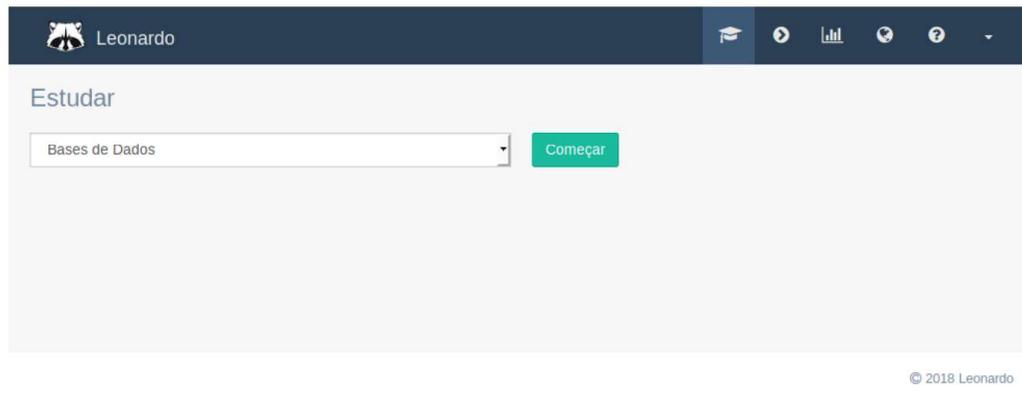


Figura 18 - Escolha do domínio de estudo.

De seguida, numa interface própria para o aluno, Figura 18, este começa por escolher o domínio de conhecimento sobre o qual pretende dar continuidade ao seu processo de aprendizagem. Seguindo-se então a apresentação de questões às quais o aluno deve responder (Figura 19).

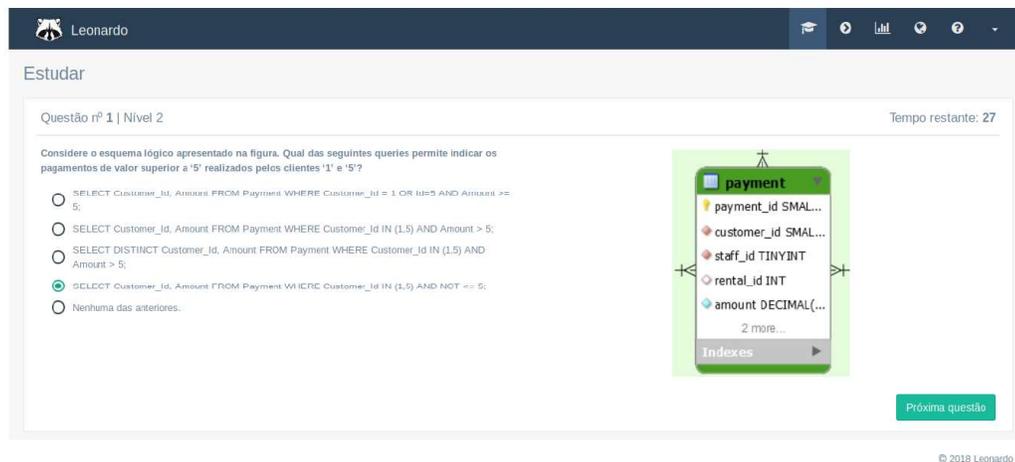


Figura 19 - Exemplo de uma possível questão.

É possível verificar que a questão apresenta várias características, desde logo o cabeçalho e opções de resposta, mas também um apoio multimédia (neste caso particular, uma imagem de

acompanhamento) e ainda um temporizador, que representa o tempo que o aluno possui para responder à questão, aspecto importante para a definição do seu perfil, como já foi visto anteriormente.

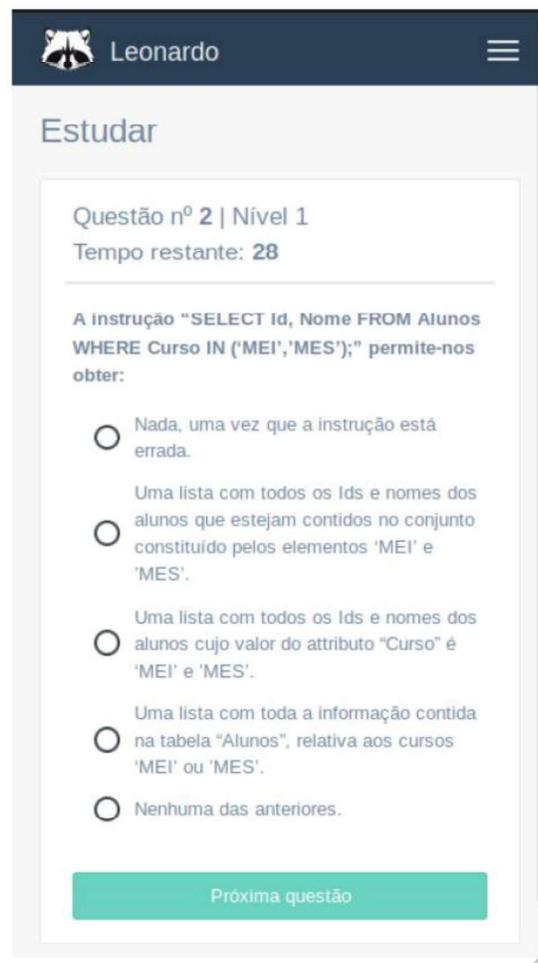


Figura 20 - Exemplo de uma possível questão com a vista de um *smartphone*.

É, também, possível ver na Figura 20 um exemplo de uma possível questão com a vista de um *smartphone*, revelando a característica *responsive* da plataforma do aluno, que irá permitir alcançar o objetivo de oferecer um sistema acessível a partir de qualquer dispositivo com acesso à

Internet, a qualquer altura, para uma aprendizagem a um ritmo apropriado e pessoal de cada indivíduo.

Ao responder a uma questão o aluno despoja um conjunto de operações no sistema, cujos principais intervenientes são o *Profiler* e o *Evaluator*. De encontro ao apresentado no diagrama de sequência na Figura 15, desde logo o *Evaluator* faz um pedido ao *Profiler*, através de uma chamada à API, para que este atualize o perfil do aluno com as novas informações recolhidas.

Admitindo que a questão apresentada ao aluno foi a exposta na Figura 17, e também que o sistema não conhecia previamente o aluno, ou seja, o seu perfil se encontrava vazio à entrada para este exercício, o novo perfil após o aluno responder, digamos que acertadamente e gastando apenas metade do tempo que dispunha, e dar-se o pedido de atualização por parte do *Evaluator*, apresentar-se-ia como consta da Figura 22. É possível verificar os perfis no sistema através de uma simples *query* MongoDB.

```
db.profiles.find().pretty()
```

Figura 21 - *Query* MongoDB para obter perfis no sistema.

```
{
  "username" : "luisfernandes",
  "user_level" : 2,
  "total_questions" : 1,
  "questions_right" : 1,
  "questions_wrong" : 0,
  "performance" : 1,
  "skill" : 0.500,
  "hypermedia_images_total": 0,
  "hypermedia_images_hitted": 0,
  "hypermedia_videos_total": 0,
  "hypermedia_videos_hitted": 0,
  "definition_total": 1,
  "definition_hitted": 1,
  "exercise_total": 0,
  "exercise_hitted": 0,
  "profile" : [
    {
      "domain" : {
        "study_cycle" : "Ensino Superior",
        "scholarity" : "Engenharia Informática",
        "description" : "Sistemas de Bases de Dados"
      },
      "user_level" : 2,
      "q_in_current_level" : 1,
      "answers_time" : 60,
      "rbacklog" : 1,
      "wbacklog" : 0,
      "hitted" : 1,
      "total" : 1,
      "skill" : 0.500,
      "hypermedia_images_total": 0,
      "hypermedia_images_hitted": 0,
      "hypermedia_videos_total": 0,
      "hypermedia_videos_hitted": 0,
      "definition_total": 1,
      "definition_hitted": 1,
    }
  ]
}
```

```
    "exercise_total": 0,
    "exercise_hitted": 0,
    "subdomains": [
      {
        "subdomain" : "SQL",
        "hitted" : 1,
        "total" : 1,
        "skill" : 0.500,
      },
    ]
  },
  "session_questions_ids" : [
    "PTEINSBD0150"
  ],
  "sessions_time" : 0,
  "questions_time" : 120,
  "answers_time" : 60,
  "last_question" : ...
},
"last_answer" : "right",
"last_difficulty_level" : 2
"language" : "pt"
}
```

Figura 22 - Exemplo de um possível perfil de um aluno no sistema.

No final de um *quiz*, o aluno é capaz de saber como se saiu durante a execução do mesmo através de uma *interface* onde lhe são apresentados alguns pontos chaves do seu estudo anterior, como é apresentado na Figura 23.

Figura 23 - Página de resultados de um *quiz*.

Para finalizar, na Figura 24, encontra-se apresentado um esboço de uma possível *dashboard* com informações e estatísticas relativas a um aluno, organizadas e disponibilizadas em gráficos para uma mais fácil compreensão das características que o aluno apresenta até aquele momento específico de interação com o sistema.



Figura 24 – Esboço de uma possível *dashboard* com dados relativos ao aluno.

Este esboço de uma *dashboard* pode, a qualquer momento, ser acedida por qualquer utilizador, com privilégios de acesso à plataforma de gestão e, escolhendo um dos seus domínios pelo qual está responsável (o caso do administrador de sistema é especial, uma vez que tem acesso a todos os domínios), consegue aceder à lista de alunos nesse domínio escolhido e, posteriormente, entrar na *dashboard* relativa a esse aluno, com as informações e características do mesmo, organizadas em gráficos, sobre esse domínio em concreto. Este esboço apresenta alguns possíveis gráficos passíveis de apresentação nesta *dashboard*, no qual é possível visualizar alguns dos atributos básicos do aluno no domínio apresentado, mas também algumas estatísticas relativas a comparações com outros alunos no mesmo domínio ou até um histórico de avaliações. Outros

gráficos poderiam ser apresentados, como um gráfico representativo do historial do aluno, no domínio apresentado, no que à *performance* diz respeito, ou um historial do seu *user_level* ao longo do tempo.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Análise ao Trabalho Realizado

Ao longo desta dissertação foi estudado e desenvolvido um módulo passível de integrar no ITS que serviu de caso de estudo, o Leonardo. Este ITS pretende servir como suporte à aprendizagem clássica, podendo ser utilizado dentro e fora da sala de aula e para o qual o módulo desenvolvido – o *Profiler* – é crucial.

Cada vez mais, e com clara tendência a aumentar, a informação e os dados são digitais e a facilidade e acessibilidade são palavras de ordem. Tendo isto em mente, também na educação é necessário que esses avanços surjam. É aqui que um tutor artificial inteligente entra, numa tentativa de combater a estagnação do setor. Os alunos podem ter acesso a um sistema que lhes permite aprender quando e onde quiserem, a partir de qualquer tipo de dispositivo com acesso à internet e a qualquer hora do dia. Além disto, os alunos passam a ter em sua posse um sistema que se irá adaptar conforme as suas necessidades e que irá basear as suas decisões na apresentação de conteúdos ao conhecimento que o aluno demonstrou ao longo do tempo.

Em conformidade com o referido, foi analisado o processo de avaliação, que é parte integrante e fulcral do sistema e, para tal, a integração do *Profiler* neste mesmo sistema e com os

restantes módulos em desenvolvimento tornou-se fundamental. A definição de características a reter do aluno, para que possam, posteriormente, ser aplicadas métricas de modo a construir o perfil do mesmo, em simbiose com o algoritmo desenvolvido para alteração de nível constituem, só por si, grande parte do foco desta dissertação e da orgânica do *Profiler*, mas é na comunicação e interligação com o módulo de raciocínio – o *Evaluator* – que o sistema realmente se torna capaz de satisfazer as deficiências que eram pretendidas colmatar no ensino tradicional.

Como aspetos positivos deste trabalho é possível destacar as alterações concisas, mas precisas, na estrutura da questão que viriam a possibilitar um perfil de aluno mais completo e funcional. O mecanismo de *backlog* apresentado surge também como uma mecânica importante no fluxo do processo de avaliação e representa a materialização de ideais de *gamification*, onde a recompensa é peça fulcral, em sintonia com conceitos provenientes diretamente da psicologia.

Não menos importante, e de ressaltar também em termos de aspetos positivos, a integração deste módulo com o *Evaluator*, através da API disponibilizada, e ainda da interligação com a *interface* do utilizador na plataforma de gestão do Leonardo, na construção da *dashboard* onde os dados recolhidos podem ser visualizados de forma organizada.

Do outro lado do espectro, é necessário apontar pontos menos positivos do projeto e aqui, sem dúvida, aparece a falta de testes do mesmo em contexto real, até ao momento. Estando o sistema Leonardo, como um todo, ainda em fase de desenvolvimento, o mesmo ainda não foi posto à prova por estudantes reais, num ambiente real, e como tal ainda não teve a oportunidade de ser verdadeiramente testado.

5.2 Trabalho Futuro

Em termos de possível trabalho a realizar no futuro, em primeiro plano surge a introdução de técnicas de *machine learning*, como observado em alguns dos estudos abordados no capítulo 2 deste documento. Entre outras opções possíveis, a introdução de *bayesian networks* e/ou *association rules* seria um dos primeiros passos para a deteção de padrões nos erros mais

comummente cometidos pelos alunos e para o sistema poder almejar ajudar na prevenção dos mesmos antes que aconteçam.

Por outro lado, a um nível mais focado na personalização e captação de atenção, a introdução de técnicas de *gamification* propriamente ditas, como um sistema de *badges*, por exemplo, poderia ser um passo interessante para o futuro do Leonardo, na expectativa de chamar mais alunos à adoção da plataforma como sistema de suporte ao seu processo de aprendizagem, apelando ao seu lado mais competitivo, como é usual nas técnicas de *gamification*.

No âmbito da personalização, a implementação de um sistema de opiniões (*opinion maker*), onde os alunos pudessem dar as suas opiniões sobre o sistema e sobre a forma como o mesmo está a avaliar o seu conhecimento e colmatar as deficiências demonstradas, seria um passo necessário para o avanço do sistema num sentido mais autónomo e preciso, aliado a uma personalização de conteúdos e *feedback* ainda mais apropriado a cada individualidade.

Bibliografia

1. Alqahtani, M. (2011) 'Architecture of Intelligent Tutoring Systems'.
2. Anohina, A. (2007) 'Advances in Intelligent Tutoring Systems: Problem-solving Modes and Model of Hints', *International Journal of Computers Communications & Control*, 2(1), p. 48. doi: 10.15837/ijccc.2007.1.2336.
3. Belo, O., Coelho, J. and Fernandes, L. (2019) 'AN EVOLUTIONARY SOFTWARE TOOL FOR EVALUATING STUDENTS ON UNDERGRADUATE COURSES'.
4. Canhoto, A. I. (2007) *Profiling behaviour: The social construction of categories in the detection of financial crime*.
5. Cherry, K. (2019) *Overview of vark learning styles*. Available at: <https://www.verywellmind.com/vark-learning-styles-2795156>.
6. Chopra, A. *et al.* (2010) 'Frame based Intelligent Tutoring System with weighted attributes and adaptive hypermedia', *International Journal of Electronic Business*, 8(4/5), p. 342. doi: 10.1504/ijeb.2010.035291.
7. Chrysafiadi, K. and Virvou, M. (2013) 'Student modeling approaches: A literature review for the last decade', *Expert Systems with Applications*. Elsevier Ltd, 40(11), pp. 4715–4729. doi: 10.1016/j.eswa.2013.02.007.
8. Ciloglugil, B. (2016) 'Adaptivity based on Felder-Silverman Learning Styles Model in E-Learning Systems', (November).
9. Django Project (2019) *Django - The web framework for perfectionists with deadlines*. Available at: <https://www.djangoproject.com/>.

-
10. Epic.org (2019) *Privacy and Consumer Profiling*. Available at:
<https://epic.org/privacy/profiling/>.
 11. García, P. *et al.* (2007) 'Evaluating Bayesian networks' precision for detecting students' learning styles', *Computers and Education*. doi: 10.1016/j.compedu.2005.11.017.
 12. González Serrano, C. *et al.* (2013) 'Designing intelligent tutoring systems: A personalization strategy using case-based reasoning and multi-agent systems', *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 2(4), pp. 41–54. doi: 10.14201/ADCAIJ2013244154.
 13. ISProf (2017) *What is Profiling?* Available at:
<http://www.bioscopegroup.org/isprof2017/index.php/what-is-profiling/index.html>.
 14. Kanoje, S., Girase, S. and Mukhopadhyay, D. (2014) 'User Profiling Trends, Techniques and Applications', *International Journal of Advance Foundation and Research in Computer*, 1(11), pp. 2348–4853.
 15. Kocsis, R. N. (2006) 'What is Criminal Profiling?', in, pp. 1–11.
 16. Kolb, D. A. (1984) 'Experiential learning: Experience as the source of learning and development, David A. Kolb, Prentice-Hall International, Hemel Hempstead, Herts., 1984. No. of pages: xiii + 256', *Journal of Organizational Behavior*.
 17. Krishnapuram, R. and Keller, J. M. (1993) 'A Possibilistic Approach to Clustering', *IEEE Transactions on Fuzzy Systems*. doi: 10.1109/91.227387.
 18. Lu, S. (2019) *Visualizing Beta Distribution and Bayesian Updating*. Available at:
<https://towardsdatascience.com/visualizing-beta-distribution-7391c18031f1>.
 19. Marques, A. and Belo, O. (2010) 'Discovering student web usage profiles using Markov chains', *9th European Conference on eLearning 2010, ECEL 2010*, 9(1), pp. 335–342.
 20. Martin, B. (1999) 'Constraint-based modelling: Representing student knowledge', *New Zealand Journal of Computing*, 7(2), pp. 30–38. Available at:
http://www.cosc.canterbury.ac.nz/~bim22/paper_nzjc.pdf.
 21. McLeod, S. (2017) *Kolb's Learning Styles and Experiential Learning Cycle*. Available at:
<https://www.simplypsychology.org/learning-kolb.html>.

-
22. MongoDB (2019) *MongoDB: The database for modern applications*. Available at: <https://www.mongodb.com/>.
23. Morales-Rodríguez, M. L. *et al.* (2011) 'Architecture for an Intelligent Tutoring System that considers learning styles', *Advances In Artificial Intelligence And Soft Computing*, 47(2012), pp. 37–47.
24. Nabeth, T. (2008) 'User profiling for attention support at school and work', in *Profiling the European Citizen: Cross-Disciplinary Perspectives*. doi: 10.1007/978-1-4020-6914-7_10.
25. Niesler, A. and Wydmuch, G. (2009) 'User Profiling in Intelligent Tutoring Systems Based on Myers-Briggs Personality Types', *Imeccs*, I, pp. 144–149. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.4060>.
26. Ohlsson, S. (1994) 'Constraint-based student modeling', in *Student modelling: the key to individualized knowledge-based instruction*, pp. 167–189.
27. Pallets Projects (2019) *Flask*. Available at: <https://palletsprojects.com/p/flask/>.
28. Python.org (2019) *Python*. Available at: <https://www.python.org/>.
29. Razmerita, L., Angehrn, A. and Maedche, A. (2003) 'Ontology-based user modeling for knowledge management systems', in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*. doi: 10.1007/3-540-44963-9_29.
30. Sani, S. M., Bichi, A. B. and Ayuba, S. (2016) 'Artificial Intelligence Approaches in Student Modeling: Half Decade Review (2010-2015)', *IJCSN International Journal of Computer Science and Network*, 5(5), pp. 2277–5420. Available at: <http://eprints.rclis.org/30166/1/Artificial-Intelligence-Approaches-in-Student-Modeling-Half-Decade-Review-2010-2015.pdf>.
31. Schiaffino, S. and Amandi, A. (2006) 'Polite personal agents', *IEEE Intelligent Systems*. doi: 10.1109/MIS.2006.15.
32. Schiaffino, S. and Amandi, A. (2009) 'Intelligent user profiling', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5640 LNAI(May), pp. 193–216. doi: 10.1007/978-3-642-03226-4_11.
33. VARK Learn Limited (2019) *Vark : a guide to learning preferences*. Available at: <http://vark-learn.com/>
-

-
34. Webb, G. I., Pazzani, M. J. and Billsus, D. (2001) 'Machine learning for user modeling', *User Modeling and User-Adapted Interaction*. doi: 10.1023/A:1011117102175.
35. Wikipedia (2018) *Profiling (information science)*. Available at:
[https://en.wikipedia.org/wiki/Profiling_\(information_science\)](https://en.wikipedia.org/wiki/Profiling_(information_science)).

Anexos

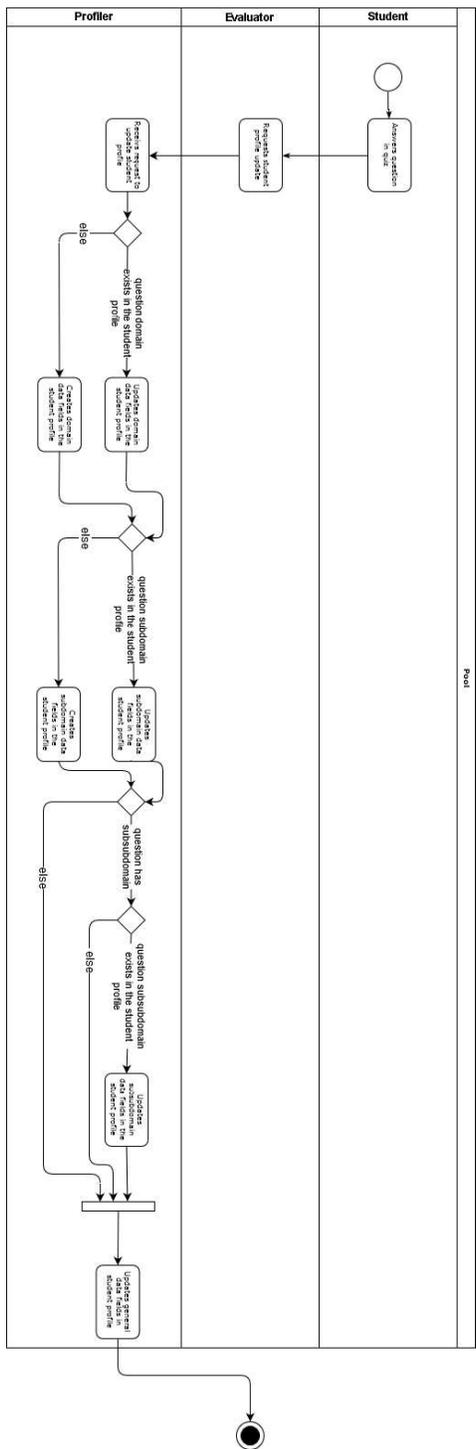


Figura 25 - (Anexos) Diagrama de atividades "Criar e atualizar perfis de utilizadores".