

## **Utilização de ontologias para a gestão de dados educacionais: caso de estudo Learning Scorecard**

Miguel Ângelo Damião Lopes

Mestrado em Engenharia Informática

Orientador(a):

Prof. Doutora Elsa Alexandra Cabral da Rocha Cardoso, Professora Auxiliar  
ISCTE - Instituto Universitário de Lisboa

Co-Orientador(a):

Prof. Doutor José Eduardo de Mendonça Tomás Barateiro, Professor Auxiliar  
Convidado

Laboratório Nacional de Engenharia Civil e ISCTE - Instituto Universitário de Lisboa

Outubro, 2021



Departamento de Ciências e Tecnologias da Informação

## **Utilização de ontologias para a gestão de dados educacionais: caso de estudo Learning Scorecard**

Miguel Ângelo Damião Lopes

Mestrado em Engenharia Informática

Orientador(a):

Prof. Doutora Elsa Alexandra Cabral da Rocha Cardoso, Professora Auxiliar  
ISCTE - Instituto Universitário de Lisboa

Co-Orientador(a):

Prof. Doutor José Eduardo de Mendonça Tomás Barateiro, Professor Auxiliar  
Convidado

Laboratório Nacional de Engenharia Civil e ISCTE - Instituto Universitário de Lisboa

Outubro, 2021



*“Every day you make progress. Every step may be fruitful. Yet there will stretch out before you an ever-lengthening, ever-ascending, ever-improving path. You know you will never get to the end of the journey. But this, is far from discouraging, only adds to the joy and glory of the climb.”*

*- Sir Winston S. Churchill*



# Agradecimentos

O desenvolvimento desta dissertação não teria sido possível sem a intervenção e apoio incondicional de determinadas pessoas às quais eu expresso um profundo sentimento de gratidão.

Começo por agradecer à professora Dra. Elsa Cardoso por todo o apoio, confiança e dedicação depositada em mim, acompanhando-me neste percurso com uma atitude positiva e encorajadora, ao desafiar-me a encarar os obstáculos encontrados e providenciando-me com experiência e ferramentas para os ultrapassar. Agradeço também ao professor Dr. José Barateiro por todo o apoio bem como a confiança em mim depositada para realizar esta dissertação e aos meus colegas António Lorvão e Inês Palma pela ajuda prestada ao longo deste percurso.

Quero agradecer à minha namorada Margarida por todo o amor dado, por toda a sua motivação, empenho, força e pelas inúmeras horas dedicadas a apoiar-me incondicionalmente nos momentos mais difíceis e que mais exigiram de mim, momentos esses em que a sua presença foi essencial para os poder ultrapassar sempre de coração cheio, cabeça erguida e com um sorriso na cara.

Aos meus pais, Manuela e José, e à minha tia Fátima que se esforçaram para que eu chegasse onde cheguei, pondo diversas vezes as suas vontades em segundo plano para possibilitar as minhas. Quero também agradecer a todos os meus amigos, colegas e restante família que fizeram parte deste percurso.



# Resumo

O Learning Scorecard (LS) é uma plataforma de gestão de desempenho, que proporciona aos alunos e docentes do Ensino Superior um ambiente analítico para a monitorização do desempenho dos primeiros numa unidade curricular (UC), valorizando a sua experiência de aprendizagem.

O LS já passou por quatro versões anteriores, contudo, estas apresentavam código pouco modular e fracamente escalável, entre outros fatores, que impediam a adição de novas funcionalidades bem como condicionavam a utilização da plataforma num ambiente real com muitos alunos. Estas limitações motivaram esta dissertação para não só refazer integralmente toda a plataforma, aproveitando conceitos desenvolvidos nas versões anteriores, como também implementar novas funcionalidades ao criar e integrar uma ontologia que serve de base para adicionar ainda mais conhecimento à plataforma.

Inicialmente, desenvolvemos as funcionalidades base do LS num novo ambiente tecnológico, seguidas pela criação da ontologia que, para além do referido anteriormente, é utilizada para mapear os conteúdos lecionados com os componentes de avaliação bem como a avaliação da dificuldade sentida pelos alunos nesses mesmos conteúdos.

A validação foi realizada com um questionário a 75 alunos inscritos no Ensino Superior que incidiu sobre os hábitos académicos dos alunos e sobre a validação das novas funcionalidades, relativamente à sua utilidade e intenção dos alunos em utilizar frequentemente as novas funcionalidades. 93% dos alunos inquiridos concordam que as novas funcionalidades promovem a sua experiência de aprendizagem. Relativamente à intenção de utilização da plataforma, 70% dos alunos referiram que utilizariam frequentemente a plataforma para estudar (49% frequentemente e 21% muito frequentemente).

**Palavras-chave:** *Learning Scorecard*, ontologia, Ensino Superior, experiência de aprendizagem



# Abstract

Learning Scorecard (LS) is a performance management platform that provides higher education students and teachers with an analytical environment for monitoring their performance in a unit course (UC), enhancing their learning experience.

LS has already gone through four previous versions, however, these had less modular and poorly scalable code, among other factors, which prevented the addition of new features and conditioned the use of the platform in a real environment with many students. These limitations motivated this dissertation to not only fully remake the entire platform, taking advantage of concepts developed in previous versions, but also to implement new features by creating and integrating an ontology as a basis for adding even more knowledge to the platform.

Initially, we developed the LS basic functionalities in a new technological environment, followed by the creation of the ontology which, in addition to the above, is used to map the contents taught with the assessment components as well as the assessment of the difficulty felt by students in those same contents.

The validation was performed with a quiz to 75 students enrolled in Higher Education that focused on the students' academic habits and on the validation of the new functionalities, regarding their usefulness and the students' intention to frequently use the new functionalities. 93% of the surveyed students agree that these functionalities promote their learning experience. Regarding the intention to use the platform, 70% of the students stated that they would frequently use the platform to study (49% frequently and 21% very frequently).

**Keywords:** Learning Scorecard, ontology, Higher Education, learning experience



# Índice

<i>Agradecimentos</i> .....	<i>i</i>
<i>Resumo</i> .....	<i>iii</i>
<i>Abstract</i> .....	<i>v</i>
<i>Índice</i> .....	<i>vii</i>
<i>Figuras</i> .....	<i>ix</i>
<i>Tabelas</i> .....	<i>xi</i>
<i>Acrónimos</i> .....	<i>xiii</i>
<b>Capítulo 1 Introdução</b> .....	<b>1</b>
<b>1.1 Motivação e enquadramento</b> .....	<b>2</b>
<b>1.2 Objetivos</b> .....	<b>3</b>
<b>1.3 Questões de Investigação</b> .....	<b>5</b>
<b>1.4 Metodologia de Investigação</b> .....	<b>5</b>
<b>1.5 Estrutura do documento</b> .....	<b>9</b>
<b>Capítulo 2 Revisão da Literatura</b> .....	<b>11</b>
<b>2.1 Semantic Web em Sistemas Educacionais</b> .....	<b>11</b>
2.1.1 E-learning .....	11
2.1.2 Sistemas Educacionais Baseados na Web Semântica .....	12
<b>2.2 Ontologias</b> .....	<b>16</b>
2.2.1 Tipos de ontologias.....	17
2.2.2 Linguagens de desenvolvimento de ontologias.....	19
2.2.3 Metodologias para desenvolver ontologias.....	22
2.2.4 Ontologias no domínio da educação.....	24
<b>Capítulo 3 Learning Scorecard 2.0</b> .....	<b>27</b>
<b>3.1 Ambiente tecnológico</b> .....	<b>28</b>
<b>3.2 Ontologia</b> .....	<b>30</b>
3.2.1 Metodologia .....	31
3.2.2 Componentes da ontologia.....	32
3.2.3 Funcionalidade e objetivos.....	34
3.2.4 Integração da ontologia com o LS.....	35
<b>3.3 Vistas da plataforma</b> .....	<b>35</b>
<b>3.4 Decisões de desenvolvimento LS 2.0</b> .....	<b>36</b>
3.4.1 Unidade Curricular .....	37
3.4.2 Quests.....	39
3.4.3 Guilds.....	42
3.4.4 Syllabus Contents .....	45
3.4.5 Diferenciação entre vistas de aluno e docente.....	50
3.4.6 Calendar .....	51

3.4.7	Leaderboards .....	51
3.4.8	Grades .....	54
3.4.9	Evaluation .....	54
3.4.10	Journal.....	55
3.4.11	Vista de Administração.....	56
<b>Capítulo 4 Validação .....</b>		<b>59</b>
4.1	Caracterização dos respondentes .....	60
4.2	Hábitos acadêmicos .....	61
4.3	Experiência de aprendizagem.....	63
4.4	Avaliação da dificuldade de aprendizagem .....	67
<b>Capítulo 5 Conclusão .....</b>		<b>75</b>
5.1	Questões de Investigação.....	76
5.2	Limitações .....	77
5.3	Trabalho Futuro .....	78
<b>Referências Bibliográficas .....</b>		<b>79</b>

# Figuras

Figura 1: <i>Roadmap</i> de implementação das versões da plataforma LS (Cardoso, 2018).....	4
Figura 2: Metodologia DSR (Peppers et al., 2007) .....	6
Figura 3: Instanciação da metodologia DSR à dissertação .....	8
Figura 4: Ciclo de desenvolvimentos do LS 2.0 .....	9
Figura 5: Fases e Procedimentos da Metodologia criada por Bravo et al., (2019).....	23
Figura 6: Ambiente tecnológico do LS 2.0 .....	28
Figura 7: Estrutura da ontologia desenvolvida no âmbito desta dissertação .....	32
Figura 8: Visão geral da ontologia na ferramenta Protégé.....	33
Figura 9: Exemplo de uma instância da ontologia do LS 2.0.....	34
Figura 10: Exemplo de uma <i>query</i> SPARQL no <i>GraphDB</i> .....	35
Figura 11: Visão de gestão da UC .....	37
Figura 12: Interface de gestão da UC expandida para a gestão de alunos.....	38
Figura 13: <i>Tab Ranks</i> do LS 2.0.....	39
Figura 14: Interface de criação de uma <i>quest</i> .....	40
Figura 15: <i>Tab XP Values</i> do LS 2.0.....	41
Figura 16: <i>Tab Grade Values</i> do LS 2.0 .....	42
Figura 17: Visão geral das <i>guilds</i> da UC .....	42
Figura 18: Perfil da <i>guild</i> (vista de docente).....	43
Figura 19: Interface de submissão de feedback .....	44
Figura 20: Interface de criação individual de <i>guild</i> .....	44
Figura 21: SCs de uma unidade curricular.....	45
Figura 22: Matriz de mapeamento de SCs por <i>quest</i> .....	46
Figura 23: Interface de submissão do mapeamento de SCs com <i>quests</i> .....	47
Figura 24: Visualização do mapeamento entre SCs e uma determinada <i>quest</i> .....	47
Figura 25: Interface de avaliação da dificuldade sentida pelo aluno.....	48
Figura 26: Interface de visualização da dificuldade sentida pelos alunos nos SCs de uma UC .....	49
Figura 27: Comparação da barra de navegação das vistas de aluno e docente entre a versão 4 e o LS 2.0 .....	50
Figura 28: Em cima, o componente <i>Calendar</i> e, em baixo, o componente <i>Timeline</i> do LS 2.0.....	52
Figura 29: <i>Leaderboards</i> no LS v4 (em cima) e no LS 2.0 (em baixo) .....	53
Figura 30: Interface de submissão de presenças (em cima) e de distribuição de notas .....	54
Figura 31: Interface <i>Evaluation</i> do LS 2.0 .....	55
Figura 32: Interface <i>Journal</i> do LS 2.0 .....	55
Figura 33: Interface de gestão de UCs na vista de administração .....	56
Figura 34: Interface de gestão de docentes na vista de administração .....	57
Figura 35: Interface de gestão de alunos na vista de administração.....	57
Figura 36: Distribuição das idades dos alunos questionados .....	60
Figura 37: Frequência de utilização das plataformas de <i>E-learning</i> .....	62
Figura 38: Distribuição dos acessos a materiais das unidades curriculares.....	62
Figura 39: Distribuição do acompanhamento da matéria quando não se comparece a uma aula .....	63
Figura 40: Distribuição dos alunos sobre como sabem quais os conteúdos para um teste .....	64
Figura 41: Utilidade do LS 2.0 para o estudo dos alunos .....	64
Figura 42: Opinião dos alunos relativamente à sua segurança no estudo.....	65
Figura 43: Opinião dos alunos relativamente à experiência de aprendizagem numa UC.....	66
Figura 44: Distribuição da importância do LS 2.0 para concretizar os quatro objetivos mencionados ..	67

Figura 45: Utilidade da visualização média de aprendizagem dos alunos numa UC .....	69
Figura 46: Utilidade da visualização das dificuldades do aluno numa UC .....	69
Figura 47: Opinião dos alunos relativa à interface de visualização de dificuldade .....	70
Figura 48: Utilidade da visualização da dificuldade média de todos os alunos .....	71
Figura 49: Frequência da classificação de dificuldade das <i>quests</i> .....	72
Figura 50: Número aceitável de conteúdos para avaliar por cada <i>quest</i> .....	72
Figura 51: Frequência de utilização da ferramenta LS 2.0 .....	73

# Tabelas

Tabela 1: Stack tecnológica utilizada na dissertação .....	30
Tabela 2: Síntese das funcionalidades do LS 2.0.....	36
Tabela 3: Distribuição das respostas por ciclo e área de estudo .....	61



# Acrónimos

LS	Learning Scorecard
UC	Unidade Curricular
DSR	Design Science Research
SEBW	Sistemas Educacionais Baseados na Web
SWBES	Semantic Web Based Educational Systems
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
DL	Description Logic
VM	Virtual Machine
SQL	Structured Query Language
SPARQL	SPARQL Protocol and RDF Query Language
NPM	Node Package Manager
SC	Syllabus Content
LO	Learning Outcome
URI	Uniform Resource Identifier
JSON	JavaScript Object Notation
XP	Experience Point
JVM	Java Virtual Machine



# Capítulo 1 Introdução

Nos dias de hoje, com a diversidade de informação gerida pelas instituições de Ensino Superior e dos sistemas de informação heterogêneos que podem coexistir dentro de uma mesma instituição, torna-se necessário o uso de tecnologias de representação do conhecimento para recuperar, estruturar ou compartilhar informação (Tapia-Leon et al., 2018).

O *Learning Scorecard* (LS) é uma plataforma de gestão de desempenho académico que aplica o princípio de gestão de qualidade “*if you can’t measure it, you can’t manage and improve it*” (Cardoso et al., 2016; Cardoso, Costa, et al., 2017; Cardoso, Santos, et al., 2017) . O seu objetivo é proporcionar aos alunos do Ensino Superior um ambiente analítico que possibilite o acompanhamento do seu desempenho numa unidade curricular (UC) contribuindo para a valorização da sua experiência de aprendizagem. O LS não se foca só nos alunos, mas também nos docentes ao fornecer uma vista com várias informações relativas ao desempenho e prestação dos alunos (Cardoso, Costa, et al., 2017). Esta ferramenta recebeu o prémio EUNIS Doerup E-learning Award 2017<sup>1</sup>. O LS foi usado no Iscte – Instituto Universitário de Lisboa (Iscte), durante os anos letivos de 2016-17, 2017-18, e 2018-19, em UCs de Business Intelligence.

O LS já foi objeto de estudo em várias dissertações (Costa, 2017; Pedroso, 2018; Rações, 2018), tendo sido desenvolvidas quatro versões da plataforma. Contudo, as versões anteriores apresentam um código pouco modular, o que dificulta a manutenção da plataforma e a adição de novos componentes ao LS sem entrar em conflito com os já existentes. Outro fator limitador é a fraca escalabilidade do código, visto que o ambiente tecnológico da última versão antes desta dissertação não é o mais adequado à implementação eficiente da plataforma num contexto real com muitos alunos (mais de cem). Nomeadamente, as vistas do *frontend* são enviadas juntamente com os dados pelo *backend* (*server-side rendering*), o que implica uma maior carga do lado do servidor. Adicionalmente, a maior parte dos parâmetros de configuração dos elementos de avaliação estão programados diretamente no código o que impede o docente de personalizar uma unidade curricular de acordo com as suas preferências ou com o desempenho dos alunos. Estas limitações motivaram esta dissertação a não só refazer integralmente a plataforma *Learning Scorecard* como também a integrar uma ontologia no LS, de modo a criar uma base para adicionar ainda mais conhecimento ao LS bem como novas funcionalidades que tirem proveito da mesma.

---

<sup>1</sup> <https://www.eunis.org/awards/dorup-award/>

## 1.1 Motivação e enquadramento

O propósito e motivação desta dissertação assenta na necessidade da melhoria contínua do processo de acompanhamento da aprendizagem dos alunos. O principal objetivo consiste em combinar o uso de ontologias com o potencial que o *Learning Scorecard* oferece em relação ao acompanhamento educacional, criando novas formas de estruturação e visualização dos elementos de uma unidade curricular. Por exemplo, a utilização de uma ontologia permite mapear e visualizar as dependências entre os conceitos lecionados e as *quests* ou missões no LS, mas também permite inferir novo conhecimento. No âmbito desta dissertação os conceitos lecionados numa UC ou conteúdos programáticos de uma unidade curricular são denominados SC (a partir da nomenclatura inglesa *syllabus contents*).

O LS é uma plataforma que está dividida em duas vistas, a do aluno e a do docente. A vista do aluno engloba ferramentas analíticas para que o próprio possa ter noção do seu desempenho e ganhar motivação através da utilização de mecânicas de gamificação (Pedroso, 2018; Pedroso et al., 2019). A vista do docente permite uma visualização agregada sobre o desempenho académico dos alunos, organizados por *alliances* ou cursos e por *guilds* (grupos de trabalho). O objetivo do LS não é permitir ao docente a monitorização individual dos alunos, mas dar acesso a informação agregada sobre a atual experiência de aprendizagem dos alunos na UC, permitindo que o docente possa agir em tempo útil, i.e., durante a execução da UC.

O entendimento e acompanhamento da aprendizagem dos alunos é um processo de melhoria contínua pois todos os anos letivos novos alunos levam a novas questões e desafios que ocorrem em cada unidade curricular. Todos os alunos são diferentes, têm curvas de aprendizagem distintas e a perceção de dificuldade de um determinado conceito varia de aluno para aluno.

A motivação para a reestruturação da ferramenta LS assenta na necessidade de garantir um ambiente parametrizável às necessidades de diferentes UCs. Por exemplo, as funcionalidades existentes, originalmente pensadas para a estrutura das UCs de *Business Intelligence*, com grupos de trabalhos, e tutorias de acompanhamento ao trabalho, devem ser repensadas para mapear diferentes estratégias de avaliação de conhecimentos numa UC.

As ontologias desempenham um papel importante na gestão e representação do conhecimento e podem ser utilizadas para um melhor entendimento do processo de aprendizagem dos alunos. Uma ontologia é uma hierarquia de conceitos com um conjunto de propriedades e relações que representam um domínio (Stancin et al., 2020). De uma forma geral, as ontologias facilitam o acesso ao conhecimento permitindo a troca adequada de informações entre pessoas e entre sistemas heterogêneos (Valaski et al., 2012). As ontologias têm sido usadas em várias áreas da informática como processamento de linguagem natural, gestão do conhecimento, *e-commerce*, integração inteligente da informação, *Web semântica*, *Semantic Web Systems*, entre outras (Valaski et al., 2012).

No contexto educacional, as ontologias têm uma variedade de benefícios e aplicações na educação continuada e superior, incluindo a partilha de informação entre os sistemas educacionais, fornecendo estruturas para a reutilização de objetos de aprendizagem permitindo o suporte inteligente e personalizado ao aluno (Wilson, 2004).

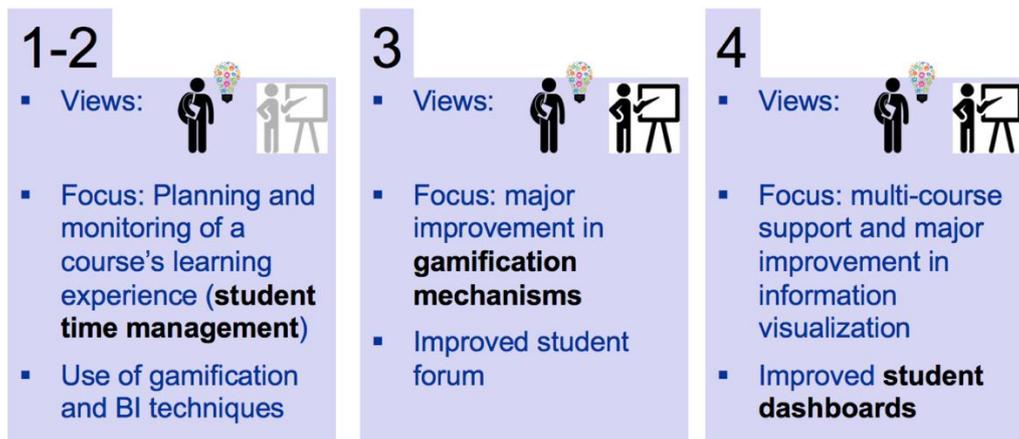
No Ensino Superior o acompanhamento da aprendizagem dos alunos não é tão presente nem frequente como nos escalões de ensino inferiores. Isto é devido à recolha de informação, pois ocorre frequentemente mais para o fim do semestre dificultando a capacidade de agir no momento oportuno (Rações, 2018).

A utilização de uma ontologia no *Learning Scorecard* para o mapeamento de conceitos (e.g., através de um *mindmap*) permite identificar padrões de precedência e/ou de dificuldades em certas matérias presentes em cada turma podendo o docente adaptar o seu método de ensino de modo a ajudar os alunos a atingirem o sucesso numa determinada unidade curricular.

## **1.2 Objetivos**

Esta dissertação tem como objetivo desenvolver uma nova versão do *Learning Scorecard* que integre o uso de uma ontologia na plataforma de forma a melhorar o processo de aprendizagem dos alunos através de um melhor entendimento entre os conteúdos lecionados e as atividades de uma unidade curricular bem como auxiliar os docentes na monitorização do desempenho dos alunos e das suas dificuldades. Tem também como objetivo melhorar e reformular o *Learning Scorecard* nas vistas de aluno e de docente através da implementação de novas ferramentas e funcionalidades que permitam uma maior parametrização do LS.

Concretamente, os objetivos deste trabalho incluem o desenvolvimento de várias funcionalidades, que são disponibilizadas em três vistas: aluno, docente e administração, bem como a criação de uma ontologia que suporte estas mesmas funcionalidades. A lista de funcionalidades a implementar na nova versão do LS (LS 2.0) combina um conjunto de algumas existentes nas versões anteriores, adaptadas e melhoradas, com novas, sobretudo ligadas à utilização da ontologia<sup>2</sup>.



**Figura 1: Roadmap de implementação das versões da plataforma LS (Cardoso, 2018)**

Como podemos observar na Figura 1, o LS já passou por vários desenvolvimentos, em que foram implementadas várias funcionalidades em cada iteração, tendo resultado em quatro versões da plataforma. As duas primeiras versões focaram-se no planeamento e monitorização da experiência de aprendizagem numa unidade curricular através da utilização de técnicas de gamificação e *Business Intelligence*. A terceira versão realizou uma otimização nos mecanismos de gamificação presentes no LS aliada à melhoria do fórum adjacente ao LS. Na quarta versão, o foco foi em permitir que o utilizador possa estar inscrito em mais do que uma UC em simultâneo, alternando entre elas como também melhorar as visualizações disponibilizadas aos utilizadores.

Nas versões anteriores, em que o código foi desenvolvido por alunos diferentes e sempre como uma prova de conceito, o código apresentava discrepâncias juntamente com uma fraca modularidade dos componentes devido a utilizar tecnologias pouco escaláveis e responsivas.

<sup>2</sup> O conjunto de funcionalidades implementadas pode ser consultado na Tabela 2, no capítulo 3.

Neste trabalho, o código da plataforma LS 2.0 foi integralmente refeito com o objetivo de oferecer escalabilidade, modularidade, coesão de código e de fácil integração na *cloud*. Foram utilizadas frameworks como *Express*<sup>3</sup> (*Node.js*) e *React*<sup>4</sup> para o *backend* e *frontend* respetivamente, já para armazenar os dados provenientes do LS foi decidido utilizar o *mongoDB*<sup>5</sup>, uma base de dados não relacional e para armazenar a ontologia decidiu-se utilizar uma instância de *GraphDB*<sup>6</sup>, uma base de dados semântica (em grafo).

A decisão de refazer integralmente o LS deve-se ao facto de o desenvolvimento iterativo das quatro versões anteriores ter sido realizado por diferentes alunos, sempre com um objetivo de prova de conceito. Já o LS 2.0 continua a ser um protótipo, mas mais “evoluído” relativamente à maturidade do código e à possibilidade de poder ser utilizado por docentes de diferentes universidades ao contrário das antigas versões que estavam desenhadas essencialmente para unidades curriculares do Iscte. Esta nova versão necessita, contudo, de resolver questões como infraestrutura de suporte, testes de *performance* e segurança antes de ser considerada um produto.

### 1.3 Questões de Investigação

Esta dissertação pretende responder às seguintes questões de investigação:

- Será possível desenvolver uma ontologia para mapear e gerir os conteúdos programáticos com as atividades de uma unidade curricular?
- Poderá a ontologia ser integrada de forma útil no *Learning Scorecard*, sendo usada pelos alunos e docentes para a monitorização da aprendizagem numa UC?

### 1.4 Metodologia de Investigação

A metodologia escolhida para esta dissertação é o DSR (*Design Science Research*) cujo objetivo foca-se em criar e validar uma proposta de solução para o problema apresentado através do desenvolvimento e posterior validação de artefactos. Artefactos podem ser entendidos como construções (vocabulário e símbolos), modelos (abstrações e representações), métodos (algoritmos e práticas) e instanciações (sistemas implementados e protótipos) (Hevner et al., 2004).

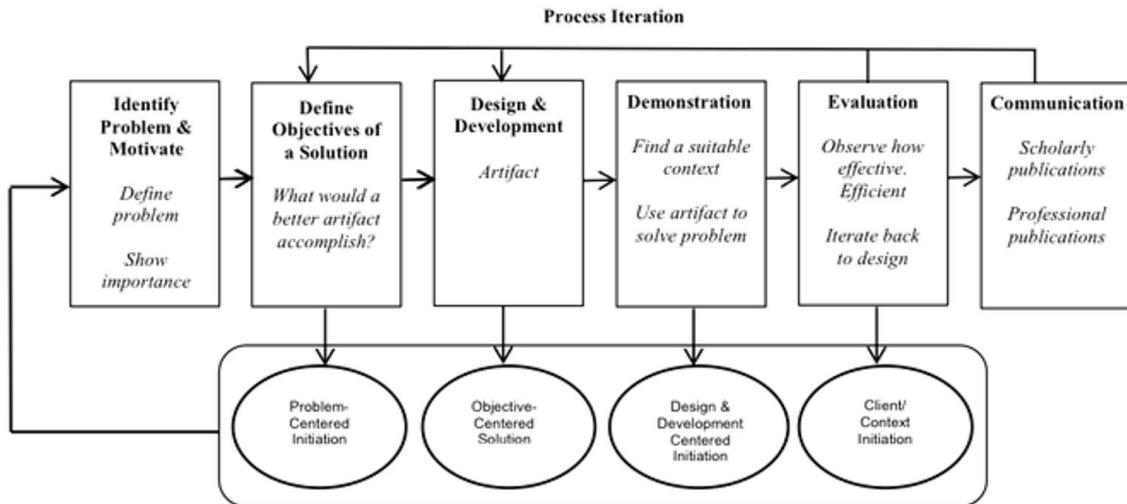
---

<sup>3</sup> <https://expressjs.com/>

<sup>4</sup> <https://reactjs.org/>

<sup>5</sup> <https://www.mongodb.com/>

<sup>6</sup> <https://graphdb.ontotext.com/>



**Figura 2: Metodologia DSR (Peffer et al., 2007)**

Podemos implementar esta metodologia seguindo um conjunto de passos (Peffer et al., 2007):

1. Identificação do problema e motivação – consiste na definição do problema que levou à iniciação do estudo e a sua respetiva motivação.
2. Definição dos objetivos para a solução – após a identificação do problema passamos a definir os objetivos da solução a desenvolver.
3. Desenho e Desenvolvimento – direcionamos o foco para desenvolvimento de artefactos definindo quais as suas funcionalidades e arquitetura e posteriormente a sua criação.
4. Demonstração – este passo tem como objetivo demonstrar a eficácia do artefacto criado para resolver o problema identificado através de experiências, simulações, estudos de caso ou outra atividade que seja considerada apropriada.
5. Avaliação – após a demonstração torna-se necessária a observação e medição da capacidade de o artefacto suportar uma solução para o problema identificado.
6. Comunicação – comunicar o problema, a sua importância, o artefacto desenvolvido, a utilidade do mesmo como também o rigor do seu design e a sua eficácia.

As etapas suprarreferidas são realizadas numa ordem sequencial, porém, o ponto de partida pode ocorrer em momentos diferentes, sendo quatro os pontos de partida possíveis (Peffer et al., 2007):

- Centrado no problema – a investigação é iniciada devido à identificação de um problema.
- Centrado em objetivos – a investigação advém de um conjunto de objetivos previamente definidos.
- Centrado no desenho e desenvolvimento – a investigação foca-se num artefacto já existente, todavia não foi formalmente considerado como solução no domínio do problema apresentado.
- Iniciação por cliente/contexto – a investigação inicia-se baseada na observação de uma solução já existente e aplicada.

O ponto de partida desta dissertação é centrado no objetivo pois a iniciação foi com base no objetivo de melhorar o *Learning Scorecard* onde serão desenvolvidos dois artefactos: a ontologia e o protótipo do LS. Nesta dissertação o DSR foi instanciado da seguinte maneira:

**Identificação do problema e motivação** – O problema identificado consiste na falta de compreensão, por parte dos docentes, da dificuldade sentida pelos alunos em relação aos conteúdos lecionados numa unidade curricular.

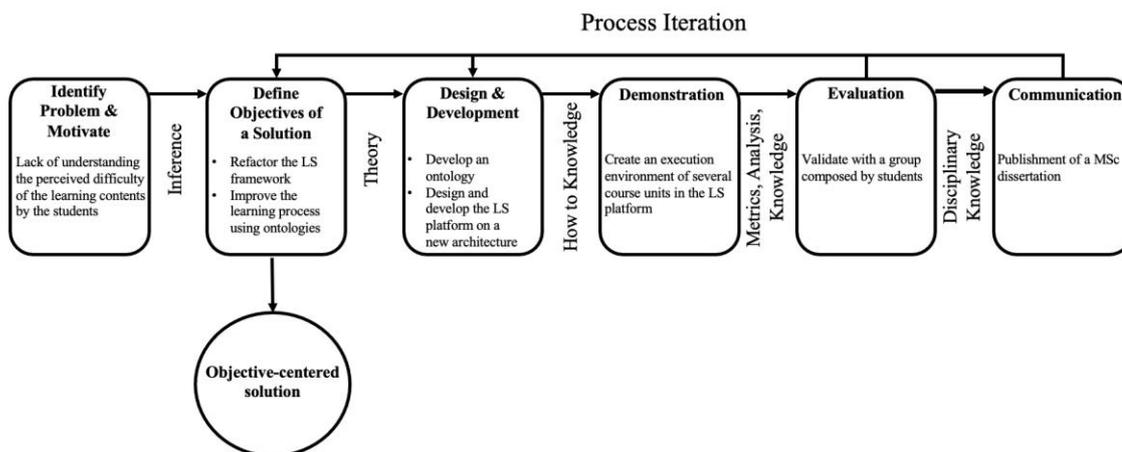
**Definição dos objetivos para a solução** – Os objetivos para esta dissertação são: melhorar a plataforma LS através da reestruturação da mesma para um ambiente robusto e escalável bem como melhorar o processo de aprendizagem através da utilização da ontologia na plataforma.

**Desenho e Desenvolvimento** – Passa pelo desenvolvimento e integração da ontologia, a definição do ambiente tecnológico a ser utilizado e o desenvolvimento da plataforma LS englobando a maioria das funcionalidades já existentes como também as novas funcionalidades relacionadas com o uso da ontologia.

**Demonstração** – A demonstração é feita através da criação de um protótipo da nova plataforma LS 2.0 com um ambiente de execução para várias unidades curriculares.

**Avaliação** – A avaliação é feita com um conjunto de alunos através de um questionário.

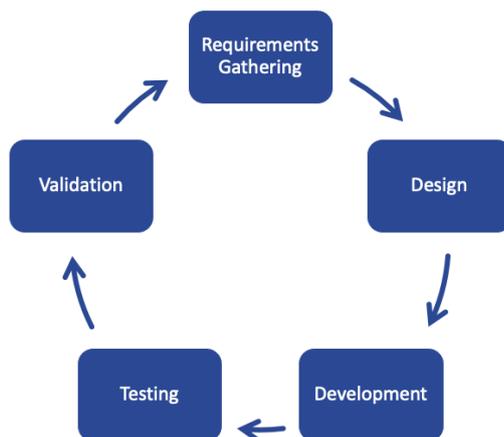
**Comunicação** – A publicação desta dissertação será a forma de comunicação da metodologia.



**Figura 3: Instanciação da metodologia DSR à dissertação**

Na fase de Desenho e Desenvolvimento está também a utilização de uma metodologia de desenvolvimento de software ágil, composta por ciclos de desenvolvimento quinzenais em que cada ciclo é constituído por cinco passos:

1. **Levantamento de requisitos** – no início de cada ciclo são definidos requisitos relativos ao desenvolvimento das funcionalidades a serem criadas no mesmo.
2. **Desenho** – após o levantamento dos requisitos segue-se a fase de desenho, em que se formulam as funcionalidades a serem desenvolvidas no *backend* como também o aspeto das mesmas no *frontend*.
3. **Desenvolvimento** – Temos depois o desenvolvimento em que se põe em prática o desenvolvimento das funcionalidades. O desenvolvimento é dividido em três fases: a construção dos *endpoints* no *backend*; a criação das funções no *frontend* para consumir os serviços do *backend*; e a criação da interface gráfica no *frontend*.
4. **Testes** – nesta fase são realizados os testes, em que numa primeira etapa são realizados a cada funcionalidade desenvolvida e noutra os mesmos são realizados num aspeto total, onde são testadas todas as funcionalidades e onde se verifica a existência de possíveis erros derivados da adição das mesmas ao código já existente.
5. **Validação** – na fase anterior as funcionalidades são testadas pelo desenvolvedor, que por sua vez tem uma perspetiva mais interna sobre as funcionalidades a serem testadas, por isso, é submetida no ambiente de execução uma “proposta” das funcionalidades desenvolvidas para que as outras partes envolvidas no projeto possam testar e confirmar que as funcionalidades desenvolvidas correspondem com o pretendido. Após a concretização bem-sucedida desta etapa inicia-se um novo ciclo, voltando ao levantamento de requisitos para outro ciclo de desenvolvimento.



**Figura 4: Ciclo de desenvolvimentos do LS 2.0**

## 1.5 Estrutura do documento

Este documento é constituído pelos seguintes capítulos. No Capítulo 2 é apresentada a Revisão de Literatura ou Estado da Arte, onde se introduz toda a pesquisa realizada sobre a utilização da *Web* semântica em sistemas educacionais, abordando tópicos como a sua definição, as plataformas de *E-learning* e sistemas educacionais baseados na *Web* semântica, e também sobre ontologias, mais concretamente a sua definição, os tipos de ontologias, linguagens utilizadas para o desenvolvimento das mesmas e ainda a sua aplicação no domínio da educação.

O Capítulo 3 aborda todo o processo de desenvolvimento não só da plataforma LS 2.0 como também da ontologia que vai suportar o LS. Para além da definição do ambiente tecnológico e do desenvolvimento da ontologia do LS 2.0 (incluindo a descrição da metodologia seguida, a sua constituição, funcionalidade, objetivos e integração com a plataforma), este capítulo apresenta as principais decisões de desenvolvimento tomadas, relativas tanto às novas funcionalidades como às antigas que foram integralmente refeitas.

O Capítulo 4 apresenta os resultados obtidos a partir do questionário desenvolvido para validar as novas funcionalidades do LS 2.0. As respostas foram analisadas em quatro secções: caracterização dos respondentes, hábitos académicos, experiência de aprendizagem e avaliação da dificuldade de aprendizagem.

Finalmente, o Capítulo 5 engloba as conclusões retiradas do desenvolvimento desta dissertação como também as respostas às perguntas de investigação, as limitações e trabalho futuro.



# Capítulo 2 Revisão da Literatura

## 2.1 Semantic Web em Sistemas Educacionais

Desde os anos 90, o ensino baseado na Web tornou-se um ramo essencial da tecnologia educacional em que o avanço da *World Wide Web* e da Internet contribuíram para o desenvolvimento de ferramentas de aprendizagem online (Bogoslov, 2018).

No contexto dos sistemas educativos, a tendência atual é de incorporar gradualmente tecnologias semânticas da *Web* que proporcionem um ambiente de aprendizagem personalizado, adaptável e inteligente (Bogoslov, 2018). Deste modo, surge a *Web Semântica* que é uma área em expansão e que recorre ao uso de ontologias para as representações dos recursos da *Web* (Bittencourt et al., 2008; Gómez-Pérez et al., 2010).

De seguida, torna-se pertinente abordar o *E-learning* que constitui uma plataforma bastante conhecida de aprendizagem online baseada na *Web*. Posteriormente, abordámos também os Sistemas Educacionais Baseados na *Web* (SEBW) semânticos uma vez que, após a implementação de ontologias, o LS torna-se de certa forma um SEBW semântico.

### 2.1.1 *E-learning*

Nos dias de hoje as plataformas de *E-learning* representam um grande envolvimento nas vidas dos alunos e docentes (Simões et al., 2013). O conceito de *E-learning* baseia-se em ensinar de uma forma moderna e mais atraente tendo um papel importante para fortalecer o conhecimento e a avaliação realizado de uma forma adaptável às necessidades do docente e com especial atenção às dos alunos (Croitoru & Dinu, 2016). Para os autores Simões et al., (2013) o *E-learning* refere-se ao uso de tecnologias de informação e comunicação para transformar e suportar o processo de aprendizagem causando um grande impacto no desenvolvimento sócio cultural e económico a nível global. Em suma, podemos considerar *E-learning* como um termo genérico que engloba os canais e sistemas de aprendizagem tais como o envio remoto de multimédia, emails, videoconferência, aulas interativas online. O *E-learning* atende a todos os tipos de ambientes educacionais como também às necessidades de aprendizagem individuais e afeta as instituições do Ensino Superior e os seus docentes e alunos como também impacta métodos de ensino e a oferta de aprendizagem. O seu propósito é conectar alunos, docentes e serviços de suporte profissionais dos quais resulta uma plataforma de *E-learning* (Zhang & Bao, 2010).

A qualidade do *E-learning* pode ser melhorada pelos seguintes aspetos (Ghirardini, 2011):

- **Conteúdo focado no aluno** – os currículos de *E-learning* devem de ser relevantes para as necessidades dos alunos, funções e responsabilidades na vida profissional.
- **Granularidade** – o conteúdo deve de ser segmentado para facilitar a assimilação de novo conhecimento e permitir uma programação flexível do tempo de ensino.
- **Conteúdo envolvente** – métodos e técnicas para serem usadas de forma criativa e motivar a experiência de ensino.
- **Interatividade** – interações frequentes entre alunos são necessárias para manter a atenção e promover a aprendizagem.
- **Personalização** – os cursos devem de ser personalizados para refletir os interesses e necessidades dos alunos e os docentes devem de ser capazes de seguir o progresso e desempenho de cada aluno.

### ***2.1.2 Sistemas Educacionais Baseados na Web Semântica***

A *Web Semântica* constitui uma ferramenta de desenvolvimento de ontologias, mais especificamente na criação de instâncias de conceitos e instâncias de relações associadas ao conteúdo dos recursos da *Web* (Gómez-Pérez et al., 2010). Desta forma as ontologias são uma solução para as representações heterogêneas de recursos da *Web* e a sua interoperabilidade.

A *Web Semântica* estende a *Web Clássica* através de uma estrutura semântica de recursos *Web* para dar suporte a agentes humanos e artificiais para entender o conteúdo, como resultado a *Web Semântica* oferece um ambiente onde os agentes de *software* podem navegar por documentos *Web* e executar tarefas sofisticadas. A *Web Semântica* permite várias melhorias para os Sistemas Educacionais Baseados na *Web*, contribuindo para a melhoria da qualidade do ensino pois pode fornecer material de ensino personalizado, recolher dados relacionados com a interação entre o aluno e o ambiente *Web*, descobrir serviços de acordo com as necessidades dos alunos e fazer recomendações (Bittencourt et al., 2008).

A *Web Semântica* é um espaço compreendido e navegável por agentes tanto humanos como de *software*, que adiciona significado estruturado e organização aos dados de navegação da *Web* atual com base em ontologias formalizadas e vocabulários controlados com ligações semânticas entre eles. Da perspetiva de *E-learning* a *Web Semântica* auxilia os alunos a localizar, aceder, interrogar, processar e avaliar recursos de aprendizagem ao longo de uma rede heterogênea distribuída como também auxilia os docentes a criar, localizar, utilizar, reutilizar, partilhar e trocar objetos de aprendizagem (dados e componentes) (Dicheva, 2008).

A visão da *Web Semântica* na educação baseia-se em três recursos fundamentais (Whitelock & Anderson, 2004):

- Capacidade de armazenamento e recuperação eficazes da informação;
- Capacidade dos agentes autónomos de melhorar a aprendizagem e o poder de recuperação e processamento de informação;
- Capacidade da Internet de suportar, estender e expandir as capacidades de comunicação dos seres humanos em vários formatos ultrapassando os limites do tempo e espaço.

A interoperabilidade semântica é definida como um estudo que liga as diferenças entre os sistemas de informação em dois níveis: de acesso e de significado (Aroyo et al., 2006). No nível de acesso o sistema e as fronteiras organizacionais devem de ser cruzados através da criação de interfaces normalizadas que compartilham serviços internos do sistema de uma forma pouco acoplada. No nível de significado os acordos sobre dados transportados devem de ser feitos para permitir a sua correta interpretação.

A utilização desta ferramenta no campo da educação é designada por SWBES (*Semantic Web Based Educational Systems*) e tem sido alvo de exploração ao longo da literatura com o objetivo de atender às necessidades da comunidade educacional (Dicheva, 2008).

Os SWBES fornecem suporte a vários utilizadores de forma eficaz para os orientar e ajudar a alcançar os seus objetivos. Representam assim a nova geração de SEBW que visa fornecer várias melhorias na qualidade dos serviços através da utilização da *Web Semântica* pois os recursos desta podem ser usados em vários sistemas educacionais (Bittencourt et al., 2008).

Num SEBW Semântico existem várias atividades educacionais envolvidas como ensinar, aprender, cooperar, colaborar, autoria entre outras que são distribuídas de acordo com os papéis de cada interveniente (Bittencourt et al., 2008):

- **Docente** – têm como dever monitorizar as interações dos alunos (resolução de problemas, avaliação) como também configurar as suas estratégias de aprendizagem.
- **Aluno** – o seu principal interesse é interagir com o sistema com base em conteúdo adaptável e personalizado para melhorar o seu conhecimento e atingir os seus objetivos de ensino.
- **Autor** – responsáveis por estruturar o conteúdo didático.
- **Grupo** – várias aplicações têm em conta o processo de aprendizagem através de grupos que se focam na aprendizagem colaborativa, na interação de um aluno com outros para

atingir objetivos pessoais como também a partilha de funções cognitivas, metacognitivas, motivacionais e emocionais entre alunos.

- **Desenvolvedor** – responsáveis por desenvolver e adicionar funcionalidades ao SEBW Semântico.
- **Auditorias** – responsáveis por recolher dados sobre a efetividade do sistema, ferramentas específicas e sobre outros participantes. O agente responsável pela avaliação de um componente pode ser artificial (agentes de controlo e suporte) como também pode ser humano (docentes, pais dos alunos).
- **Administrador** – responsáveis por gerir o sistema, cobrem todo o ciclo de vida da gestão do sistema desde o registo dos participantes até ao término.

Num SEBW Semântico os recursos educacionais representam objetos de aprendizagem (*Learning Objects*) relativos a um determinado sistema, como por exemplo, problemas, contraexemplos, unidades de atividade entre outros (Bittencourt et al., 2008).

O ambiente de interface num SEBW Semântico é a interface de comunicação entre um participante com um papel específico e o sistema. O contexto e o tipo de recursos educacionais são aspetos importantes que precisam de ser considerados quando se seleciona a tecnologia a ser utilizada no ambiente sendo que outro ponto importante a ter em conta é que o ambiente tem que suportar diversos papéis como também fornecer interfaces especializadas das ferramentas incluídas (Bittencourt et al., 2008).

### **2.1.2.1 Componentes de um SEBW Semântico**

Os SEBW Semânticos são constituídos pelos seguintes componentes: (1) ontologias; (2) ferramentas; (3) serviços *Web* semânticos; e (4) agentes inteligentes.

Os tipos de ontologias que podem ser desenvolvidas e utilizadas num SEBW Semântico são (Bittencourt et al., 2008):

- **Educacionais** – responsáveis por especificar e validar conhecimento relativo a uma característica específica do sistema como ontologias de domínio, de alunos, pedagógicas, de tarefas entre outras.
- **Interação** – contêm o conhecimento sobre os componentes do sistema como ontologias para especificar agentes, serviços *Web* semânticos e ferramentas.
- **Contexto** – responsáveis por descrever o sistema por inteiro para garantir a comunicação e a publicação do sistema na *Web*, assim, os ambientes educacionais podem requisitar e interagir através de agentes.

As ferramentas que podem ser desenvolvidas e utilizadas num SEBW Semântico são (Bittencourt et al., 2008):

- **Educacionais** – os sistemas fornecem ferramentas para comunicação síncrona e assíncrona para ensinar e guiar os alunos em sessões de aprendizagem e em tempo real, a integração destas ferramentas com tecnologias de web semântica permite novas possibilidades ao sistema através da abordagem da *Web* semântica. Várias ferramentas podem ser fornecidas de acordo com o papel do utilizador.
- **Suporte à Web Semântica** – referem-se a recursos semiautomáticos nos quais os utilizadores fornecem suporte a agentes inteligentes, por exemplo, ferramentas para mapeamento de ontologias, anotação de ontologias, *Web* semântica social, visualização de ontologias, engenharia de ontologias entre outros.
- **Administrativas** – responsáveis por configurar o ambiente em relação a papéis, utilizadores, autenticidade entre outros.

Os serviços de *Web* semântica fornecem aos utilizadores, agentes e ferramentas um acesso orientado a serviços às características de acordo com o interesse de cada um deles. Os tipos de serviços de *Web* semântica que devem de estar disponíveis são (Bittencourt et al., 2008):

- **Educacionais** – oferecem características chave para assegurar o objetivo de um sistema educacional, facilitam algumas atividades educacionais como integração de conteúdo, descoberta de recursos educacionais, avaliação e comparação entre aplicações educacionais, personalização de conteúdo e interfaces educacionais, colaboração, recomendação, resolução de problemas, diagnóstico entre outros.
- **Suporte** – referem-se a várias atividades que não têm em conta propósito educacional, estão presentes em cada sistema educacional como conversas, procura entre outros e por isso os serviços de *Web* semântica de suporte tornam-se fulcrais para estes sistemas.

Os agentes inteligentes são responsáveis por garantir a automatização das atividades, a evolução dinâmica dos requisitos entre outros e podem ser categorizados em três tipos (Bittencourt et al., 2008):

- **Agentes de Tutoria** – garantem que a interação entre as características dos sistemas e dos utilizadores de acordo com os seus papéis como a descoberta de novas informações sobre os utilizadores, conteúdo interoperacional entre aplicações e personalização do conteúdo para diferentes utilizadores.

- **Agentes de Suporte** – suportam os outros agentes para garantir a realização das respectivas atividades como suportar outros agentes a fazer *mining* da *Web*.
- **Agentes de Controle** – são importantes para as atividades administrativas como verificar o desempenho dos agentes e serviços da web semântica, fornecer acesso a agentes no sistema e identificar agentes suspeitos.

Com todas as características apresentadas podemos confirmar que o processo de ensino e de um SEBW Semântico é baseado na consulta semântica e na navegação através dos materiais de ensino, da entrega ativa dos mesmos, do desenvolvimento de ontologias e da anotação semântica (Stojanovic et al., 2001; Zem-Lopes & Isotani, 2018).

## 2.2 Ontologias

A ciência da computação tem sido uma área de aplicação considerável das ontologias, no entanto, o conceito de ontologias surgiu antes da ciência da computação (Valaski et al., 2012). O termo ontologia surgiu no ramo da filosofia como a ciência da existência ou o estudo do ser a qual estudava os tipos e estruturas de objetos, propriedades, eventos, processos e relações em todas as áreas da realidade (Smith, 2002).

A ontologia corresponde a uma descrição formal, i.e., com sintaxe e semântica bem definidas de conceitos de domínio, das suas relações mútuas e das suas limitações (Smith, 2002). Estas estruturas formais permitem a compreensão de um domínio específico através de uma hierarquia de conceitos com um conjunto de propriedades e relações que representam um domínio, permitindo categorizar o conhecimento (Al-yahya et al., 2015; Stancin et al., 2020; Valaski et al., 2012).

De acordo com Valaski et al., (2012) podemos distinguir ontologias mais simples as quais definem uma estrutura de conhecimento enquanto que, as mais complexas adicionam o poder de inferência permitindo que a ontologia através de expressões lógicas implique novos conceitos baseados em conceitos já existentes.

Atualmente, não existe uma definição consensual sendo que, este termo foi aprimorado e adaptado de acordo com a área de aplicação (Valaski et al., 2012). Com o crescente número de aplicações de ontologias em ciências da computação surge a necessidade de uma definição clara e formal (Cimiano, 2006). No âmbito da ciência da computação, podemos considerar as ontologias como um *status* de recursos que representam o modelo conceptual subjacente a um determinado domínio, descrevendo-o de forma declarativa e separando-o claramente dos aspetos procedimentais (Cimiano, 2006).

Valaski et al., (2012) refere que as ontologias têm sido usadas em várias áreas da informática nomeadamente, processamento de linguagem natural, gestão do conhecimento, e-commerce, integração inteligente da informação, *Web* semântica, oferecendo diversas vantagens:

- Conceptualização compartilhada numa lógica formal;
- Facilita a comunicação entre pessoas e agentes computacionais;
- Promove a interoperabilidade entre os sistemas organizacionais;
- Utilização por agentes computacionais para atuar substituindo seres humanos em processos ou distribuindo tarefas.

Apesar das vantagens acima referidas, as ontologias têm sido pouco utilizadas. Algumas razões para isso podem ser: tempo, custo e os recursos utilizados para o seu desenvolvimento (Valaski et al., 2012).

### ***2.2.1 Tipos de ontologias***

Dicheva et al., (2005) identificam dois tipos de ontologias: de domínio e de estrutura. As ontologias de domínio representam os conceitos básicos do domínio em consideração, juntamente com as suas inter-relações e propriedades básicas, permitindo estabelecer uma compreensão compartilhada de um domínio específico. As ontologias de estrutura definem a estrutura lógica dos conteúdos. Estas ontologias são geralmente subjetivas e dependentes dos objetivos da aplicação da ontologia, normalmente representando relacionamentos hierárquicos e de navegação.

De acordo com Falquet et al., (2011) as ontologias podem ser classificadas de acordo com a expressividade e formalidade das linguagens utilizadas e no âmbito da ontologia, ou na granularidade do domínio. As ontologias baseadas na expressividade e formalidade da linguagem incluem a relação entre vários componentes tais como, conceitos, instâncias e propriedades sendo que, este autor inclui as ontologias de informação e as ontologias formais. As ontologias formais requerem uma semântica clara para a linguagem utilizada para definir o conceito, motivações claras para as distinções adotadas entre conceitos, bem como regras estritas sobre como definir conceitos e relações. As ontologias de informação são formadas por diagramas e *sketches* com o objetivo de clarificar e organizar ideias para o desenvolvimento de um projeto sendo, normalmente, descritas através de uma linguagem visual.

Guarino (1997) identifica quatro tipos de ontologias de acordo com o nível de dependência: as de alto nível, as de domínio, as de tarefa e as aplicacionais.

As **ontologias de alto nível** descrevem conceitos bastante gerais como espaço, tempo, objeto, etc., que são independentes de um problema ou domínio em particular (Guarino, 1997). Segundo Grimm (2010) estas ontologias tentam descrever conceitos muito abstratos e gerais que podem ser partilhados em muitos domínios e aplicações, elas costumam ser muito bem pensadas e axiomatizadas pois devido à sua generalidade são reutilizadas como base para a construção de ontologias mais específicas.

As **ontologias de domínio** referem-se ao vocabulário relacionado com um domínio genérico (Guarino, 1997). De acordo com Gómez-Pérez et al., (2010) estas ontologias disponibilizam vocabulários sobre conceitos de um domínio e as suas relações, sobre as atividades que ocorrem no domínio, e sobre as teorias e princípios elementares que regem o mesmo, estas são geralmente especializações de conceitos e relações já definidos nas ontologias de alto nível.

As **ontologias de tarefa** referem-se ao vocabulário relacionado com uma tarefa ou atividade genérica (Guarino, 1997). Estas fornecem um vocabulário sistemático dos termos utilizados para resolver problemas associados a tarefas que podem ou não pertencer ao mesmo domínio (Gómez-Pérez et al., 2010).

As **ontologias aplicacionais** descrevem conceitos em função de um domínio e tarefa particulares, que são especializações de ambas as ontologias relacionadas e que correspondem frequentemente a papéis desempenhados por entidades de domínio no desempenho de uma determinada atividade (Guarino, 1997). Segundo Grimm (2010) estas fornecem o vocabulário específico necessário para descrever uma determinada tarefa em determinado contexto aplicacional.

## 2.2.2 Linguagens de desenvolvimento de ontologias

A linguagem utilizada para desenvolver a ontologia foi a *Ontology Web Language* (OWL) a qual é baseada no RDF e no RDF Schema (RDFS), tornando útil a apresentação destas duas linguagens para um melhor entendimento da ontologia desenvolvida nesta dissertação.

### 2.2.2.1 Resource Description Framework (RDF)

Antes de abordarmos a linguagem OWL, vamos apresentar duas linguagens nas quais o OWL se baseia: o RDF e o RDF Schema (RDFS).

RDF significa *Resource Description Framework*, que reutiliza a abordagem *Web* para identificar recursos e permitir que se represente qualquer relação entre dois recursos. Gandon et al., (2011) definem os conceitos *Resource*, *Description* e *Framework* da seguinte forma:

- Os **recursos** (*Resource*) são um conceito fundamental na *Web* semântica: tudo aquilo que pode ser referido é um recurso (uma pessoa, um lugar, um objeto);
- As **descrições** dos recursos (*Description*) são fulcrais para a compreensão e raciocínio sobre os mesmos, são um conjunto de atributos, características e relações relativas aos recursos;
- A **Framework** significa que fornece modelos, linguagens e sintaxes para as descrições.

De acordo com Lassila & Swick (1999) a *framework* RDF consiste em três tipos de objetos:

- **Recursos** – todas as coisas que possam ser descritas por expressões RDF são recursos. Os recursos são sempre nomeados por URIs e IDs opcionais. Qualquer coisa pode ter um URI e a extensibilidade dos URIs permite a introdução de identificadores para qualquer entidade imaginável.
- **Propriedades** – característica, atributo, relação ou aspeto utilizada para descrever um recurso.
- **Statements** – um recurso juntamente com uma propriedade e respetivo valor da mesma para o recurso constituem um *statement* RDF. Estes três componentes são denominados de sujeito, predicado e objeto do *statement*.

Os triplos descrevem e ligam objetos através da junção de recursos, propriedades e valores de propriedade em que temos o sujeito (recurso), o predicado (propriedade) e objeto (valor) (Gandon et al., 2011). Posto isto numa frase, o RDF fornece uma estrutura de dados e modelo padrão para codificar dados e meta dados sobre qualquer coisa.

### 2.2.2.2 RDFS

Segundo Draft (2003) RDFS ou *RDF Schema* é uma extensão semântica do RDF, fornece mecanismos para descrever conjuntos de recursos e as relações entre eles como também permite-nos criar o nosso próprio vocabulário.

É uma linguagem para declarar e descrever os tipos de recursos (classes), as relações entre eles (propriedades de objeto) e entre recursos e tipos de atributos (também propriedades de tipos de dados). O RDFS permite nomear e definir vocabulários usados na classificação de gráficos RDF como: nomear as classes dos recursos, tipos de relação entre indivíduos destas classes e fornecendo as suas assinaturas, i.e., o tipo de recursos que elas associam, deste modo, o RDFS permite declarar o “esqueleto” taxonómico de uma ontologia numa linguagem universal (Gandon et al., 2011).

### 2.2.2.3 *Ontology Web Language (OWL)*

A linguagem OWL é uma linguagem ontológica padrão recomendada pelo *World Wide Web Consortium (W3C)* (Falquet et al., 2011; Heflin, 2007; Prot et al., 2011; Stancin et al., 2020). Foi concebida para ser utilizada por aplicações que necessitam de processar o conteúdo da informação em vez de apenas apresentar informação aos seres humanos, facilitando uma maior capacidade de interpretação do conteúdo da Web (Falquet et al., 2011).

O conceito da *Web Semântica* diz que a informação deve ter um significado explícito, para que as máquinas a possam processar de forma mais inteligente (Heflin, 2007). A linguagem OWL é a melhor para representar o conhecimento na *Web semântica* pois destina-se a fornecer uma linguagem que pode ser utilizada para descrever conceitos e relações entre eles que são inerentes a documentos e aplicações *Web* (Falquet et al., 2011; Stancin et al., 2020).

A *Web Semântica* não se limita a criar conceitos, também permite aos utilizadores fornecer definições formais para os termos padrão que criam, permitindo então utilizar algoritmos de inferência para raciocinar sobre os conceitos (Heflin, 2007).

O OWL possui as seguintes características (Prot et al., 2011):

- Engloba um conjunto muito rico de operadores tais como, a intersecção, união e negação.
- É baseado num modelo lógico diferente onde é possível construir conceitos complexos a partir de conceitos mais simples.

- Ajuda a manter a hierarquia correta por exemplo, no caso em que as classes podem ter mais do que um progenitor através da utilização de um argumentista (*reasoner*) que pode verificar se todas as afirmações e definições da ontologia são ou não coerentes entre si e pode também reconhecer quais os conceitos que se enquadram nas definições.

Uma ontologia OWL consiste em três componentes: classes, propriedades e indivíduos. Os **indivíduos** representam objetos num dado domínio do nosso interesse (Prot et al., 2011). O OWL é utilizado, portanto para definir estas instâncias chamadas indivíduos e afirmar propriedades sobre eles (Falquet et al., 2011).

As **propriedades** OWL ligam dois indivíduos entre si e estas podem ter inversos (e.g., o inverso de *hasOwner* é *isOwnedBy*) (Prot et al., 2011). Elas representam relações e são divididas em três tipos: propriedades de objeto, propriedades de tipo de dados e propriedades de anotação (metadados).

**Object properties.** As propriedades de objeto representam uma relação entre indivíduos e podem ser caracterizadas de sete maneiras (Prot et al., 2011):

- **Funcionais** – um indivíduo que possua uma propriedade com esta característica apenas se pode relacionar uma vez com outro indivíduo através da mesma.
- **Funcionais Inversas** – seguem o mesmo conceito que as funcionais, pois são as propriedades inversas das mesmas.
- **Transitivas** – com uma propriedade transitiva, podemos inferir relações através da mesma propriedade, i.e., se a mesma propriedade relaciona o indivíduo A com o B e o B com o C então o indivíduo A está relacionado com o indivíduo C.
- **Simétricas** – através de uma propriedade simétrica, se A estiver relacionado com B através de uma propriedade simétrica, então B está relacionado com A pela mesma propriedade.
- **Assimétricas** – se uma propriedade for assimétrica isto significa que se A estiver relacionado com B pela propriedade então B não pode estar relacionado com A pela mesma.
- **Reflexivas** – Uma propriedade é reflexiva quando ela relaciona um determinado indivíduo com ele mesmo.
- **Irreflexivas** – Uma propriedade irreflexiva permite inferir que se o indivíduo A estiver relacionado com B pela propriedade então A e B não são o mesmo indivíduo.

**Datatype properties.** Estas propriedades descrevem uma relação entre um indivíduo e valores, podemos também especificar um conjunto de valores aos quais uma propriedade deste tipo se pode relacionar.

**Annotation properties.** A linguagem OWL permite que classes, propriedades, indivíduos e a própria ontologia tenham anotações com vários elementos de metadados, i.e., comentários, data de criação, autor ou referências a recursos.

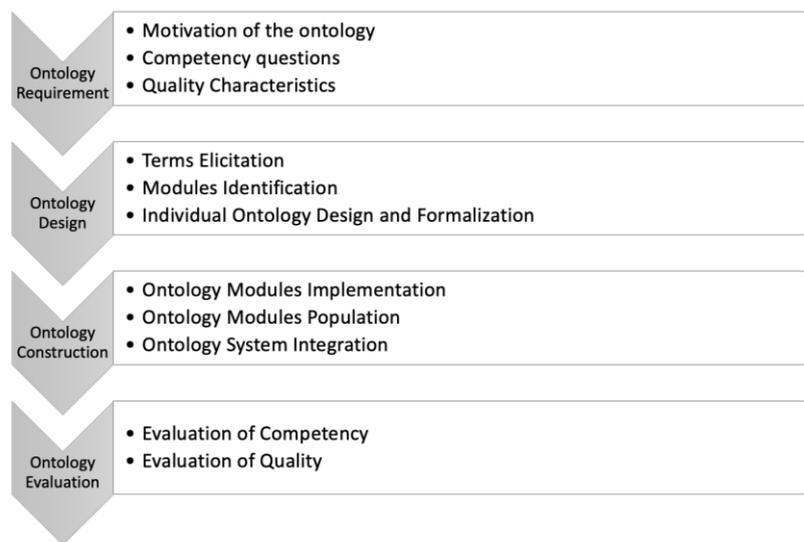
O OWL permite formalizar um domínio através da definição de conceitos chamados **classes** e das propriedades dessas classes (Falquet et al., 2011). As classes são conjuntos que contêm indivíduos e podem ser organizadas numa hierarquia superclasse e subclasse em que a primeira engloba a segunda (Prot et al., 2011). Elas são construídas através de descrições que especificam as propriedades que devem ser satisfeitas por um indivíduo para ser um membro da classe (Prot et al., 2011).

**Classes disjuntas.** Consiste em “separar” um grupo de classes, através da desunião umas das outras o que significa que, a disjunção assegura que um indivíduo que se afirmou ser um membro de uma das classes do grupo não pode ser membro de qualquer outra classe desse grupo (Prot et al., 2011).

### ***2.2.3 Metodologias para desenvolver ontologias***

A partir dos anos 90 começou a surgir o interesse pela construção de ontologias sendo que, a ausência de diretrizes comuns e estruturadas foi um entrave ao desenvolvimento, reutilização e aplicação destas (Gómez-Pérez et al., 2010). Deve-se ter em consideração que o processo de desenvolvimento de ontologias não é linear uma vez que, cada atividade pode ser repetida várias vezes (Falquet et al., 2011). Neste sentido, para a construção ontologias do zero é necessário a definição de um conjunto de metodologias gerais de modo a facilitar o processo de reengenharia de novas ontologias. Por metodologia entende-se um conjunto abrangente e integrado de técnicas ou métodos que criam uma teoria geral dos sistemas de como uma classe de trabalho deve ser realizada (IEEE, 1990).

Ao longo da literatura podemos encontrar algumas ferramentas para construir ontologias (e.g., *CommonKADS*<sup>7</sup>, *NEON*<sup>8</sup> e *Protégé*<sup>9</sup>). Dentro das *frameworks* existentes destaca-se a metodologia METHONTOLOGY, a qual engloba cinco fases para seguir durante o desenvolvimento de ontologias (Gómez-Pérez et al., 2010): Especificação (*Specification*), Conceptualização (*Conceptualization*), Formalização (*Formalization*), Implementação (*Implementation*) e Manutenção (*Maintenance*). Apesar desta metodologia ser considerada uma das mais completas para a criação de ontologias, ela gera uma sobrecarga de esforço para a documentação durante a conceção e desenvolvimento de sistemas ontológicos muito grandes (Bravo et al., 2019).



**Figura 5: Fases e Procedimentos da Metodologia criada por Bravo et al., (2019)**

Bravo et al., (2019) propuseram uma metodologia de desenho e construção ontológica tendo em consideração a criação de ontologias consistentes, modulares, coerentes, utilizáveis e reutilizáveis como podemos ver na Figura 5. Esta metodologia consiste em quatro fases e em cada uma delas é definida uma série de tarefas juntamente com métodos e técnicas para construir a ontologia, considerando as características de qualidade. As fases são as seguintes:

- **1ª Fase: Especificação de requisitos da ontologia** (*Ontology Requirements Specification*) a qual tem como objetivo a identificação do âmbito da ontologia, definição de cenários possíveis, os utilizadores, a competência da ontologia, e as características de qualidade que esta deve atender. Esta fase engloba duas tarefas, especificar a motivação da ontologia, clarificando os cenários possíveis, os utilizadores

7 <https://commonkads.org/>

8 [http://neon-toolkit.org/wiki/Main\\_Page.html](http://neon-toolkit.org/wiki/Main_Page.html)

9 <https://protege.stanford.edu/>

e as aplicações que irão beneficiar e especificar a competência da ontologia por consenso dentro de um grupo de especialistas no domínio da ontologia.

- **2ª Fase: Desenho da ontologia** (*Ontology Design*) visa a produção de um desenho formal da ontologia a partir da realização de atividades relativas a elicitacão de termos que consiste na produçã de uma lista de termos relevantes para o domínio considerado, identificacão de módulos de ontologia que consiste em decidir o conjunto de ontologias individuais que irão estar em conformidade com o sistema ontológico, conceçã e formalizacão da ontologia individual que consiste em definir os axiomas terminológicos elementares de cada ontologia individual utilizando a Lógica de Descrição (DL).
- **3ª Fase: Construçã da ontologia** (*Ontology Construction*) na qual o objetivo é codificar todos os módulos da ontologia, utilizando um editor de ontologias e uma linguagem padrão, e integrar todos os módulos num sistema ontológico. As atividades incluídas nesta fase é primeiramente a implementacão que consiste na utilizacão de um editor para implementar os módulos ontológicos, a populaçã de módulos que consiste em instanciar a ontologia com indivíduos para avaliar as definições iniciais, relações e axiomas com o objetivo de verificar que nenhuma das definições de classe e axiomas tem contradicões lógicas e, a integraçã dos módulos que consiste na importaçã do conjunto de ontologias individuais para o sistema de ontologias.
- **4ª Fase: Avaliaçã da ontologia** (*Ontology Evaluation*) a qual deve ter em consideraçã a competência da ontologia e os requisitos de qualidade de forma a avaliar se o modelo ontológico está bem construído. A competência de um modelo ontológico pode-se definir como um conjunto de perguntas a que a ontologia é capaz de responder e a sua qualidade pode ser medida como o grau de conformidade que tem em relaçã aos critérios de desenho estabelecidos (e.g., clareza, coerência, modularidade).

#### **2.2.4 Ontologias no domínio da educaçã**

A área da educaçã foi pioneira na compreensã de uma ontologia como ferramenta cognitiva. Em muitos aspetos, foi devido à ampla disseminaçã do paradigma construtivista de aprendizagem e ao amplo uso de tecnologias de conhecimento (Dicheva et al., 2005).

Dicheva et al., (2005) combinam diferentes tipos de aplicações educacionais de ontologias que se podem dividir numa perspetiva tecnológica ligada à representaçã e recuperaçã do conhecimento bem como, *Web Semântica* e numa perspetiva de aplicaçã através da utilizacão de tecnologias de aprendizagem como mapas conceptuais e *mindmaps*.

Segundo Al-yahya et al., (2015) as ontologias são utilizadas de várias maneiras em sistemas de *E-learning*:

- Na modelação e gestão curricular, os elementos do currículo são modelados de forma a facilitar o acesso e recuperação das informações do currículo permitindo, visualizar o currículo e garantir conformidade com a visão e missão da instituição. Adicionalmente, estes elementos podem fornecer uma estrutura em que as unidades curriculares podem ser ligadas a resultados e objetivos de aprendizagem específicos, e facilitar a avaliação e o alinhamento com as necessidades do mercado e requisitos de acreditação.
- Descrição de domínios de aprendizagem de diferentes perspetivas para uma descrição mais rica e uma recuperação de conteúdo de aprendizagem. Por exemplo, na ontologia de domínio de disciplina e na ontologia de tarefa de aprendizagem (lição, item de avaliação, exercícios).
- Descrição de dados do aluno para avaliação e personalização de acordo com o perfil do aluno através da monitorização do desempenho do aluno e dados históricos.
- Descrição dos serviços de *E-learning*, ao fornecer um vocabulário de intemporalidade entre os vários sistemas educacionais, o que permite também a partilha de informação entre sistemas heterogéneos de *E-learning*.

Snae & Brückner (2007) desenvolveram um sistema de *E-learning* ontologicamente orientado designado por O-DEST que fornece uma plataforma unificada para registo, avaliação, planeamento, entrega de conteúdos, gestão de registos, e elaboração de relatórios. Através do uso de ontologias apropriadas o sistema permite partilhar e integrar os conteúdos de *E-learning* disponíveis como também utilizar as ontologias para conceber o modelo de domínio dos conteúdos. Nesta abordagem as ontologias foram utilizadas especificamente em três módulos:

- **Ferramentas para os docentes** que compreendem funções para auxiliar os professores na criação de objetos de aprendizagem, a relacionar novos objetos aos já existentes, a reutilizar objetos, e na recolha de dados.
- **Ferramentas para os alunos** que permitem aos estudantes dominar o material de aprendizagem e cumprir os objetivos de aprendizagem do curso. Os alunos podem partilhar expectativas e interesses, predisposições e competências reais, sendo guiados através do material de aprendizagem e podem ativamente marcar pontos importantes, obter ajuda contextual e medir competências.

- **Ferramentas de administração** apoiam diferentes funções e tarefas de gestão do sistema como manter e atualizar os registos de alunos e professores, administrar os conhecimentos e funções do domínio para o sistema e a segurança dos dados.

## Capítulo 3 Learning Scorecard 2.0

Este capítulo está relacionado com a terceira etapa do DSR, Desenho e Desenvolvimento. É apresentada a definição do ambiente tecnológico da nova versão do LS, denominada de LS 2.0 pois toda a plataforma foi reestruturada com vista a suportar as novas funcionalidades relativas aos artefactos desta dissertação: a ontologia e o protótipo do LS. Posteriormente abordamos as alterações feitas na plataforma a nível funcional bem como a reformulação de diversos conceitos existentes na versão anterior do LS.

O *Learning Scorecard* é uma plataforma cujo intuito é oferecer apoio a docentes e alunos através de dados gerados pela mesma que auxiliam os docentes a monitorizar de forma contínua as unidades curriculares que lecionam e permite aos alunos visualizar o seu desempenho em cada cadeira em que estão inscritos.

A plataforma é principalmente dividida em duas vistas: a de aluno e a de docente e nesta nova versão conta com uma terceira: a de administração. Estas vistas são posteriormente aprofundadas ao longo deste capítulo pois foram alvo de alterações no âmbito desta dissertação.

A vista de aluno fornece as visualizações necessárias para que o aluno possa ter conhecimento do seu percurso numa determinada unidade curricular como também oferece ferramentas para que possa fornecer dados importantes para a monitorização contínua da unidade curricular pelo docente. A vista de docente providencia ferramentas de gestão, avaliação e visualização ao docente relativamente a cada uma das unidades curriculares que o mesmo leciona. O docente dispõe de várias visualizações para acompanhar o progresso de cada aluno em todas as fases da unidade curricular como também de ferramentas para avaliar o mesmo. A vista de administração permite a um novo tipo de utilizador, o administrador, realizar operações de gestão da plataforma de modo a providenciar o suporte necessário aos docentes durante a execução das unidades curriculares que os mesmos lecionam, nomeadamente a nível de parametrização ou personalização das funcionalidades à realidade de cada unidade curricular.

Até à presente data o LS passou por quatro versões, cada uma adicionando funcionalidades à anterior, nesta dissertação é apresentada a quinta versão, que devido à reestruturação de que a plataforma foi alvo, foi denominada de LS 2.0, pois foi refeita integralmente, tendo por base os conceitos fundamentais presentes nas versões antecedentes.

### 3.1 Ambiente tecnológico

O ambiente tecnológico do LS 2.0 apresentado na Figura 6 suporta todos os requisitos definidos nas dissertações de Costa (2017), Pedroso (2018) e Rações (2018) respondendo às necessidades que advieram das mesmas, bem como inclui os componentes necessários para implementar as novas funcionalidades apresentadas nesta dissertação. Este ambiente tecnológico foi escolhido tendo em consideração a facilidade de manutenção e extensão do LS. A versão anterior, ao não seguir um ambiente modular, era bastante limitativa para a rápida manutenção e evolução da plataforma LS.

Para hospedar tanto o *frontend* como o *backend* foi utilizada a plataforma *Heroku*<sup>10</sup>. Sendo o objetivo o desenvolvimento apenas de uma prova de conceito, o *Heroku* é um ambiente na *cloud*, gratuito, que simplificou a disponibilização (*deployment*) do LS.

O *frontend* em *React* comunica com o *backend* em *Express (NodeJS)* que por sua vez comunica com duas bases de dados a funcionarem numa máquina virtual (VM) hospedada na plataforma *Azure*<sup>11</sup>: uma instância de *mongoDB*, para armazenar e aceder a todos os dados necessários para o funcionamento da plataforma e uma instância de *GraphDB*, para armazenar e aceder a toda a informação relativa à ontologia desenvolvida nesta dissertação. É importante referir que toda a comunicação entre os componentes é feita através do *standard REST*.

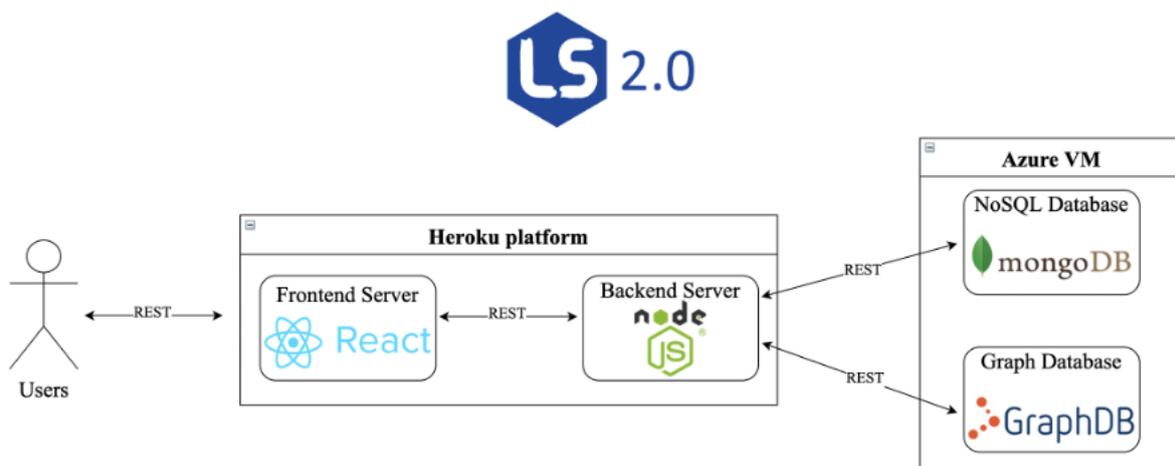


Figura 6: Ambiente tecnológico do LS 2.0

<sup>10</sup> <https://www.heroku.com/>

<sup>11</sup> <https://azure.microsoft.com/>

A semelhança entre as últimas versões do LS e o LS 2.0 é a utilização da *framework Express* para o *backend*. Relativamente ao *frontend*, que antes era gerado pelo *Express (server-side rendering)*, passou a ser disponibilizado através de uma aplicação em *React*, cuja principal vantagem é que todo o processamento das vistas passa a ser feito na máquina do utilizador (*client-side rendering*) reduzindo significativamente a carga imposta ao servidor de *backend*.

É a partir do *backend* que se encontra toda a lógica que é fornecida ao *frontend* e onde são efetuadas as chamadas para ambas as bases de dados. O *backend* é composto por um conjunto de serviços REST, constituindo assim uma API REST. Desta forma, garante-se o isolamento entre a camada de dados e a camada de visualização, que não precisa de conhecer detalhes sobre a organização e estrutura de dados, nem sobre a tecnologia de armazenamento dos mesmos. Adicionalmente, através da API REST consegue-se independência tecnológica entre as diferentes camadas. Isto é, o ambiente de visualização é completamente independente do restante ambiente tecnológico da plataforma LS. Esta contribuição é relevante, uma vez que simplifica a extensão do LS. Por exemplo, no futuro podemos explorar informação com o Power BI, bastando para isso aceder à API REST. O *frontend*, atualmente desenvolvido em *React*, poderá vir a ser desenvolvido numa outra tecnologia e a API REST continua a ser utilizável.

Na versão anterior do LS usava-se *MySQL*, uma base de dados relacional, já nesta versão é utilizada uma instância de *mongoDB*, uma base de dados não-relacional, que armazena a informação como documentos (objetos) que, auxiliada pela *framework mongoose* no *backend*, permite aceder à informação de uma forma fácil e intuitiva.

A adição mais significativa no ambiente tecnológico relativa a esta dissertação é a instância de *GraphDB*, uma base de dados em grafo que armazena a ontologia desenvolvida nesta dissertação e os respetivos triplos. Esta base de dados permite não só armazenar informação explícita (definida pela *query SPARQL*) como também informação implícita, inferida através de um conjunto de regras e de propriedades (relações entre classes). Com a adição da ontologia podemos usufruir de uma estrutura de dados que nos permita no futuro inferir um vasto conhecimento sobre vários componentes do LS e tornar a plataforma mais completa ao dotá-la da capacidade de mapear mais relações entre os elementos de avaliação podendo calcular e atribuir notas.

É importante referir que os componentes do ambiente tecnológico foram escolhidos de modo a tirar o máximo proveito de soluções *open-source*. Podemos definir *open-source* como software informático cujo código fonte está disponível sob uma licença (ou acordo tal como o domínio público) que permite aos utilizadores estudar, alterar e melhorar o software, e redistribuir de forma modificada ou não modificada (Randhawa, 2008).

As *frameworks* utilizadas para construir o *frontend* e *backend*, respetivamente *React* e *Express (NodeJS)*, tal como ambas as bases de dados como o *mongoDB* e o *GraphDB*, são *software open-source*. De notar também que todos os componentes instalados tanto no *frontend* como no *backend* através do NPM – *Node Package Manager* (ferramenta utilizada para integrar componentes feitos pela comunidade no nosso código), são também de uso gratuito à exceção do componente necessário para a criação do *mindmap*, o *goJS* desenvolvido pela Northwoods Inc., que após pedida nos foi concedida autorização para o utilizar.

Em relação à infraestrutura de suporte para o LS 2.0, todas as ferramentas necessárias à implementação da plataforma na *cloud* foram disponibilizadas de graça devido à iniciativa *Github Student Pack*, fornecida pelo *Github*. Graças a esta iniciativa tivemos a possibilidade de hospedar na *cloud* o nosso *frontend* e *backend* dentro da plataforma *Heroku* como também criar um servidor em *CentOS*, uma distribuição de *Linux*, na plataforma *Azure*, onde estão hospedadas as nossas bases de dados (*mongoDB* e *GraphDB*). A Tabela 1 sumariza o conjunto de ferramentas que foram utilizadas no âmbito desta dissertação.

**Tabela 1: Stack tecnológica utilizada na dissertação**

Ferramenta	Descrição	Propósito
<i>Visual Studio Code</i>	IDE escolhido	Desenvolver <i>frontend</i> (React) e <i>backend</i> (NodeJS)
<i>mongoDB</i>	Base de dados não relacional	Armazenar dados e informação funcional do LS
<i>GraphDB</i>	Base de dados semântica	Armazenar conhecimento do domínio do LS
<i>Postman</i>	Plataforma para testes a <i>endpoints</i>	Testar os <i>endpoints</i> do backend
<i>Protégé</i>	Editor de ontologias	Desenvolver a ontologia do LS

## 3.2 Ontologia

No âmbito desta dissertação foi proposta a criação de uma ontologia de modo a suportar a plataforma *Learning Scorecard*. Esta ontologia serve de base para um conjunto de novas funcionalidades e através de funções integradas no *GraphDB*, como por exemplo, a inferência que auxilia todo o processo de criação de novo conhecimento implícito a partir de informações fornecidas pelos utilizadores.

### 3.2.1 Metodologia

Para o desenho e desenvolvimento da ontologia foi seguida a metodologia criada por Bravo et al., (2019) utilizando a ferramenta *Protégé* desenvolvida pela universidade de Stanford.

De acordo com a metodologia, o primeiro passo consistiu em encontrar a motivação para criar a ontologia e o contacto com indivíduos especialistas no domínio do LS. A motivação deve-se a não existir uma ontologia disponível na Web a qual nos seja possível estender que abrangesse os conteúdos requisitados para esta versão do LS. No contexto de ontologias, é sempre preferível estender uma já existente de modo a aumentar o conhecimento sobre um certo domínio em vez de criar de raiz uma nova ontologia. Após o contacto com especialistas neste domínio, procedeu-se à recolha de uma lista com questões de competência, ou seja, questões que a ontologia deverá responder e que será usada para avaliar a ontologia.

O segundo passo nesta metodologia passa pela eliciação de termos, identificação de módulos, desenho da ontologia e consequente formalização. Na primeira fase deste passo são escolhidos os termos relevantes para o domínio pretendido. De seguida, passamos a decidir o conjunto de termos que vão constituir a ontologia, que para esta dissertação são: *Course Units*, *Quests*, *Syllabus Contents* e *Learning Outcomes*, de frisar que os últimos não serão abordados no âmbito desta dissertação e ficam para trabalho futuro. A fase seguinte passa por desenhar e formalizar a ontologia, ou seja, definir as relações hierárquicas, propriedades de objeto e propriedades de dados.

O terceiro passo foi realizado com a ferramenta *Protégé* e consiste na implementação, povoamento e integração da ontologia. Na implementação foram criadas todas as classes, propriedades e definidas todas as hierarquias.

Para o povoamento, foi primeiro utilizada a ferramenta *Cellfie* do *Protégé* em que importamos os axiomas via um ficheiro *Excel* para podermos criar os indivíduos e respetivas relações para testar a consistência da ontologia através do *reasoner* (um agente que valida a ontologia à procura de inconsistências). Este procedimento será depois feito via interações dentro do próprio LS para povoar a ontologia no *GraphDB*. Na integração, importamos a ontologia para a instância de *GraphDB* como também o conjunto de regras, uma verificação de consistência é feita durante a importação da ontologia e a nossa ontologia fica pronta para ser povoada através de interações dentro do LS, como referido acima.

O quarto passo é a avaliação, em que numa primeira fase revemos a lista de questões de competência desenvolvida no primeiro passo e com a ajuda dos especialistas verificamos se a ontologia responde às mesmas, de seguida verificamos a qualidade, clareza, coerência e modularidade da ontologia. A validação foi realizada através da integração da ontologia no LS de forma a obter e criar novo conhecimento sobre as unidades curriculares e os seus constituintes.

Após a realização de todos os passos que constituem a metodologia seguida, obtemos o resultado pretendido: a ontologia do LS (ver Figura 7).

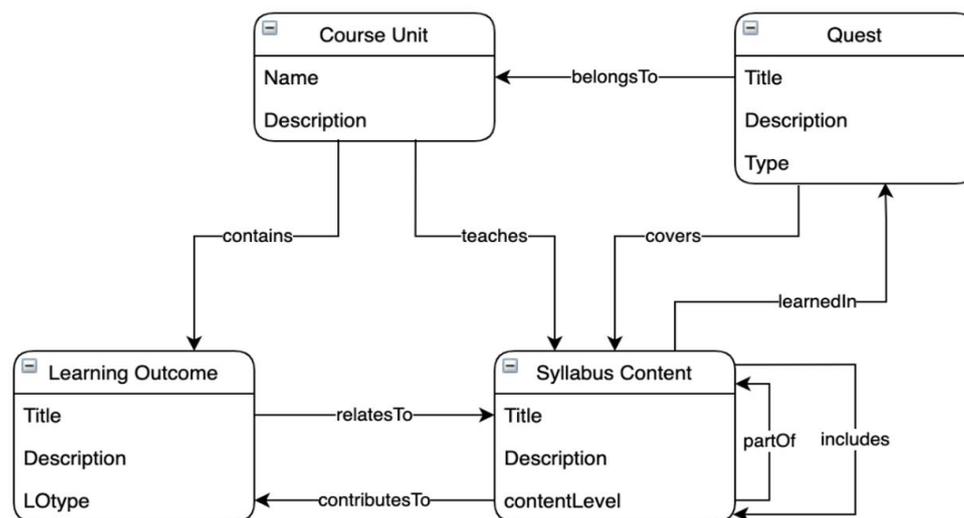


Figura 7: Estrutura da ontologia desenvolvida no âmbito desta dissertação

### 3.2.2 Componentes da ontologia

A ontologia é composta por quatro classes que representam alguns componentes do LS: *Course Unit*, *Syllabus Content* (SC), *Learning Outcome* (LO), *Quest* como podemos observar na Figura 7.

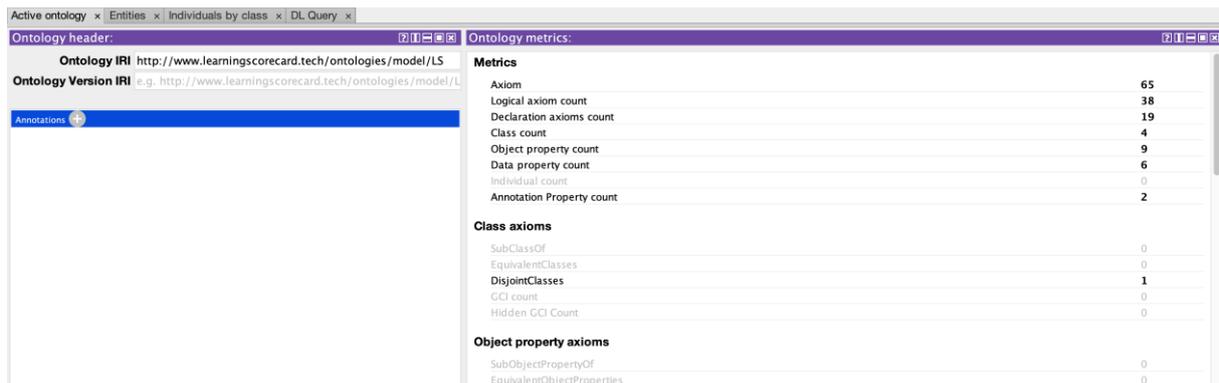
Estas classes possuem várias propriedades: as de objeto (relações) e as de tipos de dados (atributos). Relativamente às de tipos de dados podemos destacar:

- **Type** – relativa à classe *Quest*, esta propriedade define o tipo da *quest*;
- **LOtype** – relativa à classe *Learning Outcome*, refere-se ao tipo do *Learning Outcome*;
- **contentLevel** – relativa ao nível dos *Syllabus Contents*, temos os de contentLevel 1 que englobam os de contentLevel 2 que por sua vez englobam os de contentLevel 3 e por aí adiante.

Passamos assim para as propriedades de objeto das classes:

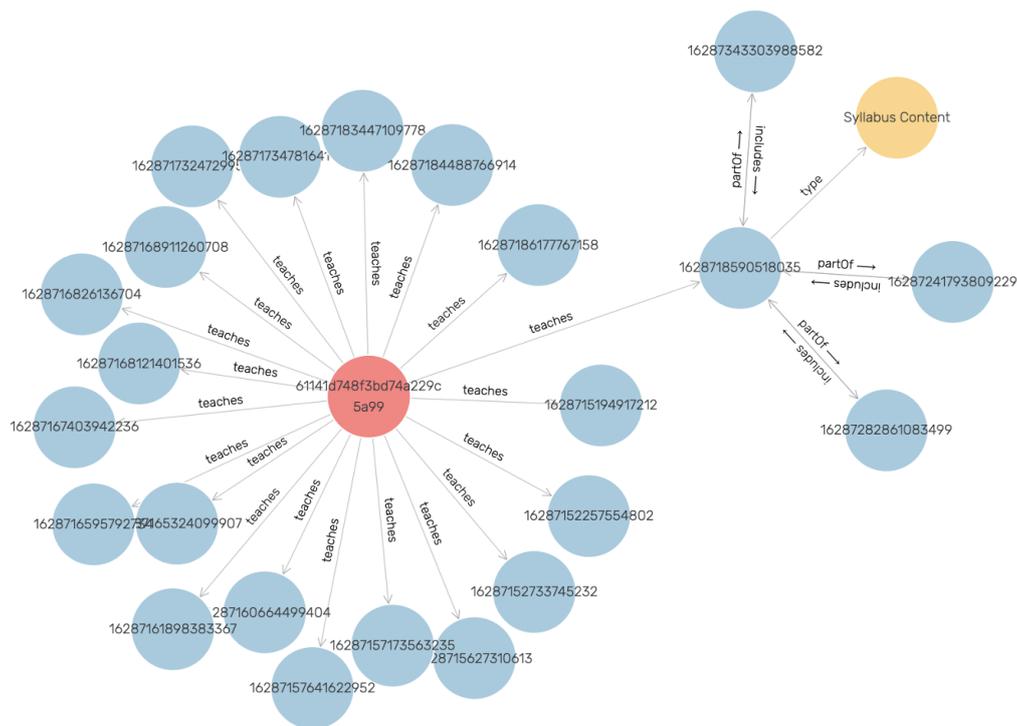
- **belongsTo** – representa a relação entre uma *quest* e uma unidade curricular, é uma propriedade **funcional** (uma *quest* pertence apenas a uma unidade curricular);
- **contains** – representa a relação entre uma unidade curricular e um *Learning Outcome* (uma UC contém *Learning Outcomes*);
- **teaches** – representa a relação entre uma unidade curricular e um *Syllabus Content* (uma UC ensina *Syllabus Contents*);
- **covers** – representa a relação entre uma *quest* e um *Syllabus Content* (uma *quest* cobre *Syllabus Contents*);
- **learnedIn** – representa a propriedade inversa da propriedade **covers** (se uma *quest* cobre um SC então o SC é aprendido naquela *quest*);
- **includes** – representa a relação entre um SC de nível superior e um de nível inferior, é uma propriedade **transitiva** (se um SC de nível 1 inclui SCs de nível 2 e os SCs de nível 2 incluem SCs de nível 3 então os SCs de nível 1 incluem os SCs de nível 3);
- **partOf** – representa a propriedade **inversa** da propriedade **includes** (se um SC A inclui o SC B então o SC B pertence ao SC A);
- **relatesTo** – representa a propriedade entre um LO e um SC (um LO está associado a um SC);
- **contributesTo** – representa a propriedade **inversa** da propriedade **relatesTo** (se um LO está associado a um SC então o SC contribui para o LO).

A Figura 8 e a Figura 9 representam, respetivamente, a vista geral da ontologia do LS no *Protégé* e um exemplo de uma instância da ontologia do LS 2.0.



Ontology metrics:	
<b>Axiom</b>	65
Logical axiom count	38
Declaration axioms count	19
Class count	4
Object property count	9
Data property count	6
Individual count	0
Annotation Property count	2
<b>Class axioms</b>	
SubClassOf	0
EquivalentClasses	0
DisjointClasses	1
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0

Figura 8: Visão geral da ontologia na ferramenta Protégé



**Figura 9: Exemplo de uma instância da ontologia do LS 2.0**

### 3.2.3 Funcionalidade e objetivos

Devido ao *GraphDB* possuir um *reasoner*, que para além de validar a consistência da ontologia infere sobre a informação existente para criar novas relações, a utilização desta ontologia permite-nos obter toda a informação sobre o domínio do LS relevante ao contexto desta dissertação. Para otimizar a capacidade de inferência do *reasoner* importamos também um conjunto de regras (*ruleset*) para que o *reasoner* possa inferir novas relações a partir das mesmas regras. Estas regras são expressas em triplos OWL, por exemplo: se a UC A ensina o SC 1 e se o SC 1 inclui o SC 2 então a UC A ensina o SC 2. Com estas regras podemos criar novas relações e dar azo a que o *reasoner* possa criar cada vez mais inferências.

Esta ontologia torna-se fulcral para desenvolver uma nova funcionalidade no LS, que necessita do mapeamento das relações entre *quests* e SCs, um *mindmap* interativo com a hierarquia dos *Syllabus Contents* pertencentes a uma unidade curricular que permita ao docente visualizar a dificuldade sentida pelos alunos em cada um dos SCs que os mesmos avaliam ao validar cada *quest*. Já os alunos obtêm a mesma visualização, mas apenas com a sua própria avaliação de dificuldade. Esta funcionalidade será abordada posteriormente.

### 3.2.4 Integração da ontologia com o LS

A integração da ontologia com o LS começa pela criação de uma ligação entre a plataforma e a instância de *GraphDB*, onde a ontologia está armazenada. A interação entre o LS e o *GraphDB* é feita via SPARQL, uma linguagem de *query* semelhante ao SQL, mas em que a sintaxe se baseia em triplos OWL. A linguagem SPARQL tem dois tipos de interação: QUERY e UPDATE. A primeira serve para realizar consultas (SELECT e ASK), já a segunda tem como função inserir, atualizar ou apagar triplos (INSERT e DELETE).

A Figura 10 apresenta um exemplo simples de como utilizamos SPARQL no LS:

```
PREFIX LS: <http://www.learningscorecard.tech/ontologies/model/LS#>
select * where {
    ?quest LS:belongsTo LS:${uc};
    LS:title ?title;
    LS:type ?type.
}
```

**Figura 10: Exemplo de uma query SPARQL no GraphDB**

Em primeiro lugar definimos os prefixos, i.e., os URIs das ontologias que vamos utilizar; neste caso, vamos utilizar apenas a ontologia do LS. Como foi referido acima, a sintaxe das *queries* é feita em triplos OWL, i.e., em <sujeito> <predicado> <objeto>. De notar que os recursos (classes, instâncias e relações) de uma ontologia seguem todos o mesmo padrão: <URI da ontologia>:<ID do recurso>.

Nesta *query* todas as palavras que são antecidas por “?” são *wildcards*, i.e., componentes do triplo que podem assumir qualquer valor. Estamos, portanto, a seleccionar todas as *quests* que pertencem (*belongsTo*) a uma UC, que é definida pelo prefixo/URI da ontologia seguido por “:” e o ID da UC. No SPARQL, todos os triplos devem de ser sucedidos por um “.” salvo se as linhas seguintes continuarem a usar o mesmo sujeito, como é o caso desta *query* em que para além da primeira linha, seleccionamos também o título e o tipo das *quests*.

Os dados são recebidos em formato JSON e processados pelo *backend* para de seguida serem enviados para o utilizador. Com isto obtemos informações tais como o mapeamento de *quests* com SCs, essencial para o desenvolvimento do componente *mindmap*.

## 3.3 Vistas da plataforma

O LS 2.0 apresenta três vistas, Aluno, Docente e a mais recente, Administração. A Tabela 2 apresenta uma visão geral sobre todos os componentes do LS 2.0 em comparação com a última versão a ser disponibilizada aos utilizadores.

**Tabela 2: Síntese das funcionalidades do LS 2.0**

VISTA	FUNCIONALIDADE	VERSÃO 4	LS 2.0
ALUNO	Visualização de XPs - <i>History</i>	✓	✓
	Visualização de <i>Dashboards</i>	✓	(Trabalho Futuro)
	Validação de <i>quests</i>	✓	✓
	Visualização de SCs e dificuldade via <i>mindmap</i>		✓
	Avaliação de dificuldade avaliada de SCs por <i>quest</i>		✓
	Visualização do mapeamento de <i>quests</i> e SCs via <i>mindmap</i>		✓
	Recuperação de <i>password</i>	✓	✓
	Alternar entre UCs	✓	✓
	Calendário	✓	✓
	<i>Timeline</i>		✓
	Visualização <i>Grades</i> (notas)	✓	✓
	Visualização <i>Guild</i>	✓	✓
	<i>Leaderboards</i>	✓	✓
DOCENTE	Consulta e distribuição de notas	✓	✓
	<i>Dashboards</i> relativos a performance dos alunos	✓	(Trabalho Futuro)
	Visualização <i>Guilds</i>	✓	✓
	Criação e gestão de alunos, docentes, <i>quests</i> e <i>guilds</i>	✓	✓
	Interface única de gestão da UC		✓
	Definição geral de limites de XP por <i>rank</i>		✓
	Definição geral de valores para as notas (A - F)		✓
	Definição geral de XPs atribuídos por <i>quest</i>		✓
	Definição geral da penalidade de <i>Last Chance</i> por <i>quest</i>		✓
	Divisão entre presenças e notas de <i>quests</i>		✓
	Criação de SCs		✓
	Visualização da perceção de dificuldade dos alunos via <i>mindmap</i>		✓
	Visualização de atividades dos alunos ( <i>Journal</i> )	✓	✓
	Importação dos alunos para a UC	✓	✓
	Mapeamento de <i>quests</i> com SCs		✓
	Visualização do mapeamento de <i>quests</i> e SCs via <i>mindmap</i>		✓
Calendário	✓	✓	
<i>Timeline</i>		✓	
ADMINISTRAÇÃO	Criação e gestão de UCs		✓
	Criação de utilizadores (docentes, opcionalmente alunos)		✓
	Gestão da plataforma		✓

Analisando a Tabela 2, é importante salientar que a funcionalidade dos *dashboards* ficou para trabalho futuro, pois vai ser alvo do trabalho específico de outra dissertação de mestrado.

### 3.4 Decisões de desenvolvimento LS 2.0

Como referido no início deste capítulo, o LS foi integralmente refeito, com certas funcionalidades das versões anteriores a serem recriadas. Ao longo desta dissertação foram tomadas várias decisões relativas à funcionalidade da plataforma bem como à lógica inerente

aos processos da mesma. Nesta seção vamos detalhar as decisões de desenho, apresentando as interfaces desenvolvidas.

### 3.4.1 Unidade Curricular

A unidade curricular no LS é a peça principal da plataforma LS, é o objeto onde as interações mais importantes ocorrem. Nesta dissertação a UC foi alvo de várias alterações de modo a dinamizar o processo de aprendizagem da mesma.

#### 3.4.1.1 Gestão

Para que o docente possa ter uma visão geral sobre os componentes que constituem uma UC foi criada uma visualização do estilo “cockpit”, representada na Figura 11. Esta interface permite adicionar/remover: alunos, docentes, *guilds* e *quests*; iniciar a UC; personalizar a UC, editando as opções em termos de: valores de XP por *rank*, valores de notas qualitativas e valores de XPs atribuídos por *quest* (caso a UC ainda não tenha sido iniciada).

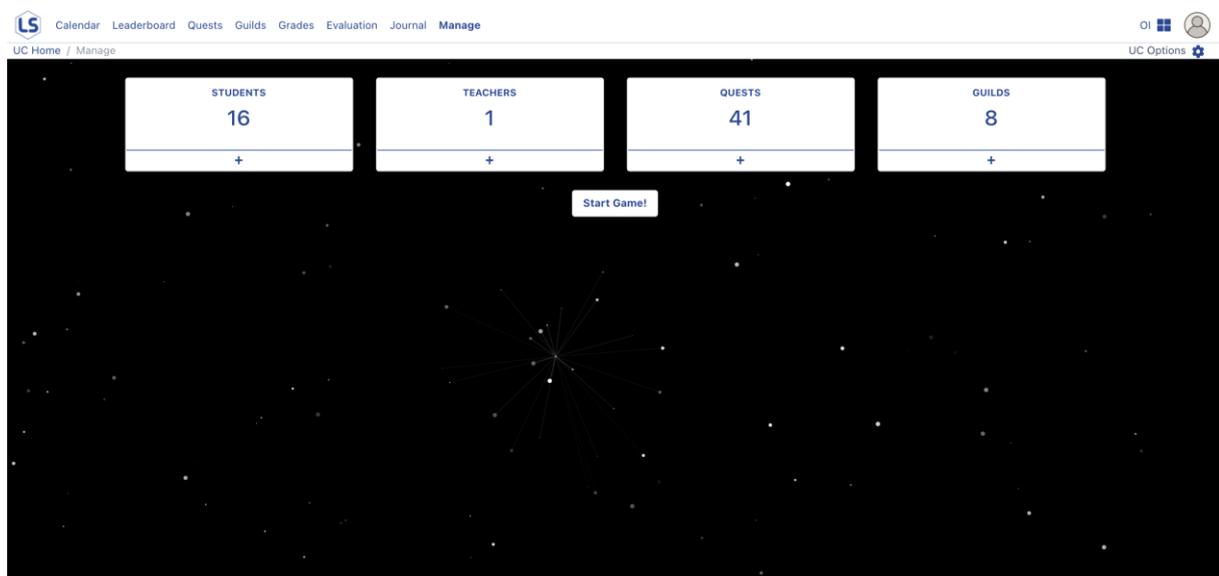


Figura 11: Visão de gestão da UC

Na Figura 11, o docente ao clicar num dos quadrados expande uma vista que contém uma tabela relativa ao quadrado que clicou; em cada tabela o docente tem a opção de remover os conteúdos presentes na mesma. A Figura 12 exemplifica a interface expandida para o quadrado de alunos. Nos quadrados existe outro botão adjacente com o símbolo “+” com o intuito de adicionar alunos, docentes, *quests* e *guilds*. O docente tem também a opção de aceder à interface para editar *ranks*, notas e valores de XPs através do botão “*UC Options*” no canto superior direito da barra de navegação. Caso a UC ainda não tenha sido iniciada o docente pode fazê-lo através do botão “*Start Game!*”.

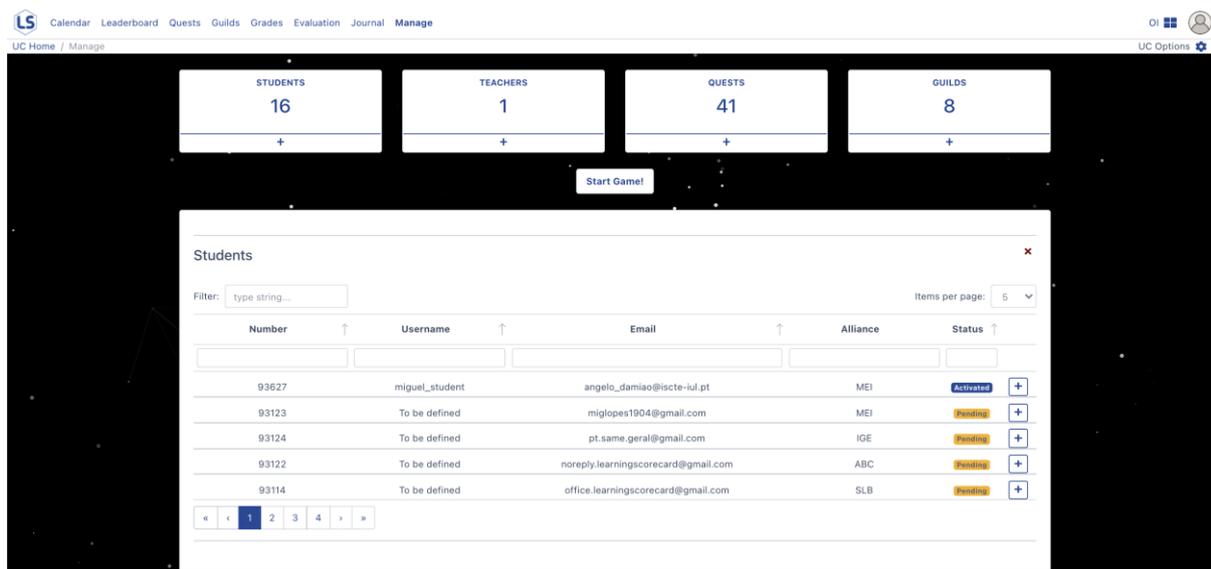


Figura 12: Interface de gestão da UC expandida para a gestão de alunos

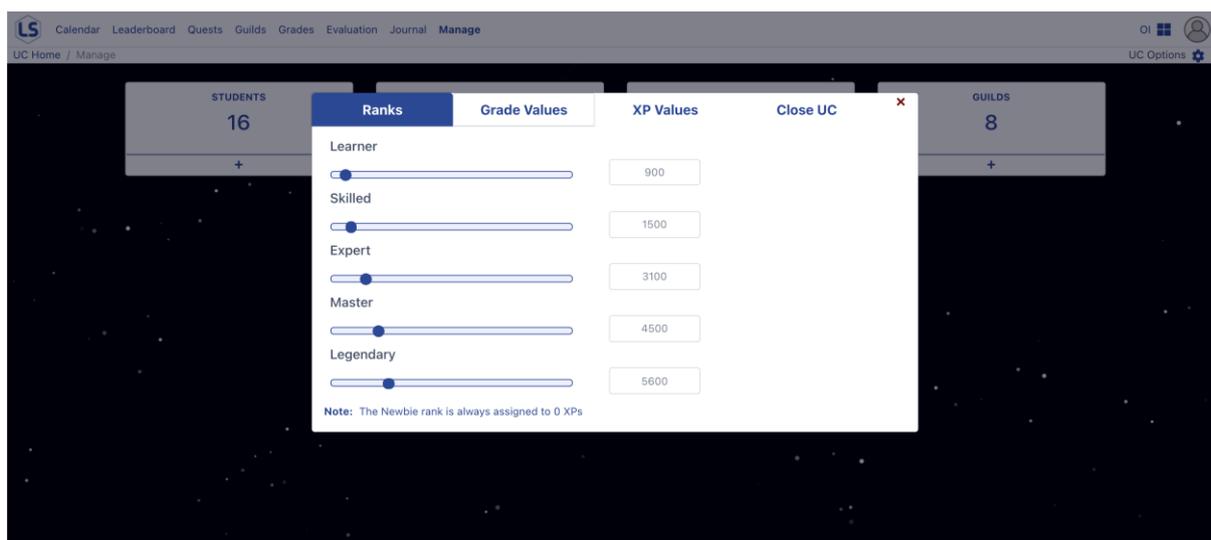
### 3.4.1.2 Start Game

O conceito de unidade curricular manteve-se relativamente igual no LS 2.0 com novas adições. Quando uma UC é criada entra agora na fase de preparação. Nesta fase o docente pode alterar os valores padrão relativos aos *ranks*, *quests* e notas. Durante esta fase não podem ser atribuídos XPs ou quaisquer outras interações de avaliação pois o “jogo ainda não começou”.

O docente tem então que iniciar a execução da UC através do botão “*Start Game*”. A partir desse momento, o docente não poderá alterar os valores padrão pois não se deve alterar as “regras” do jogo quando este já teve início. No entanto, é possível adicionar novos alunos, *quests*, *guilds* e SCs de modo a dinamizar o ensino na UC.

### 3.4.1.3 Ranks

Na última versão do LS os *ranks* eram divididos segundo a percentagem total de XPs (*experience points*) obtidos por *quests*. Uma das bases em que o LS 2.0 se assenta é o dinamismo. É o docente que define os valores que vão estar associados a cada *rank*, ou seja, entre que valores um *rank* se encontra. Esta alteração parte do princípio de que o número de *quests* numa UC nunca se mantém durante toda a duração da mesma pois o docente pode adicionar ou remover *quests* à medida que a execução da UC decorrer, de modo a fornecer aos alunos todas as ferramentas necessárias para terem sucesso na mesma. Posto isto, se os *ranks* tivessem por base o valor total de XPs das *quests* de uma UC, quando um docente adicionasse ou remove uma *quest* o *rank* do jogador podia também alterar-se, o que vai contra a ideia de que não devemos alterar as regras do jogo.



**Figura 13: Tab Ranks do LS 2.0**

De notar que o número de *ranks* mantém-se, bem como os nomes da versão anterior. Resta referir que o *rank Newbie* não está incluído pois é o *rank* de entrada para todos os jogadores, logo o seu valor é definido como 0 XPs. A Figura 13 apresenta a interface onde os docentes podem parametrizar o valor máximo de XP para cada *rank*.

### 3.4.2 *Quests*

As *quests* são uma peça fundamental nesta nova versão do LS. É através delas que: são atribuídos os XPs aos alunos (componente que quantifica a avaliação do aluno), se consegue o mapeamento entre as mesmas e os SCs, os SCs associados são avaliados. Na plataforma LS 2.0 as *quests* podem ser criadas de duas maneiras: em conjunto através da importação de um ficheiro *Excel* ou unicamente através da interface própria na plataforma.

#### 3.4.2.1 *O conceito de quest*

Nas versões anteriores do LS, as *quests* eram um tipo de evento. No LS 2.0, o conceito de *quest* foi mais aprofundado de modo que as funcionalidades refeitas que utilizem *quests* sejam aprimoradas, para além das novas funcionalidades relacionadas com a ontologia. Uma *quest* passa assim a ser diferenciada por cinco tipos: *Class Attendance*, *Practical Assignment*, *Quiz*, *Exercise* e *Event*. O conceito de evento deixa de existir e passa a ser um tipo de *quest*, isto é explicado pelo facto de, se uma *quest* for um tipo de evento então teria que ter um subtipo associado ao tipo de *quest*, que se apenas ao ler é confuso, a nível de código mais o seria.

De notar que as *milestones* são *quests* ou eventos obrigatórios, fundamentais para que o aluno se mantenha em jogo (com avaliação contínua no LS), com isto faz sentido englobar o conceito de evento como tipo de *quest* e definir *milestones* apenas pelo atributo *mandatory* (que se verdadeiro torna a *quest* obrigatória).

### 3.4.2.2 Criação individual de *quests*

Como referido anteriormente, existem duas formas de criar *quests*: em conjunto através da importação de um ficheiro *Excel* ou individualmente através de uma interface no LS 2.0 como apresentado na Figura 14. Apesar de existirem valores padrão para o valor máximo de XPs para cada tipo de *quest*, o docente pode definir outro valor nesta interface. Esta decisão foi tomada para permitir ao docente a possibilidade de escolher um valor específico de XP para uma *quest* em particular, mesmo quando a UC entra em fase de início de jogo.

É importante referir que esta interface é também adaptada para a edição de *quests*, pois caso o docente queira alterar algum campo ou atualizar uma *quest* tem a oportunidade de o fazer através desta interface, instanciada com os valores da *quest* pretendida.

Figura 14: Interface de criação de uma *quest*

### 3.4.2.3 Atribuição de XPs

Uma *quest* tem vários atributos associados, entre eles o valor máximo de XPs que podem ser atribuídos, o tipo de *quest*, o *Last Chance Penalty* (penalidade de percentagem de XPs atribuídos aplicada se o aluno não validar a *quest* dentro das datas definidas), a nota (em percentagem se for *Quiz*, A – F se for *Exercise* ou *Practical Assignment*) e a dificuldade sentida. Os dois últimos atributos são atualizados na entidade do aluno, pois são individuais para cada aluno.

Nas versões anteriores do LS, o valor máximo de XPs era definido por *quest*, quando se importava o ficheiro *Excel* com a informação relativa às *quests*. Já a *Last Chance Penalty* era algo que estava embutido no código (“*hardcoded*”). Nesta versão os valores dos XPs como também do *Last Chance Penalty* são definidos no início da preparação da UC, através de uma interface própria. Na Figura 15 é apresentada a interface para a definição dos valores padrão dos XPs atribuídos por tipos de *quest* juntamente com a das penalidades. É importante frisar que estes valores podem ser alterados até ao início da unidade curricular. Em relação às penalidades, a percentagem apresentada é deduzida do valor de XPs atribuído ao aluno no caso de ele validar a *quest* depois dos prazos estipulados pelo docente.

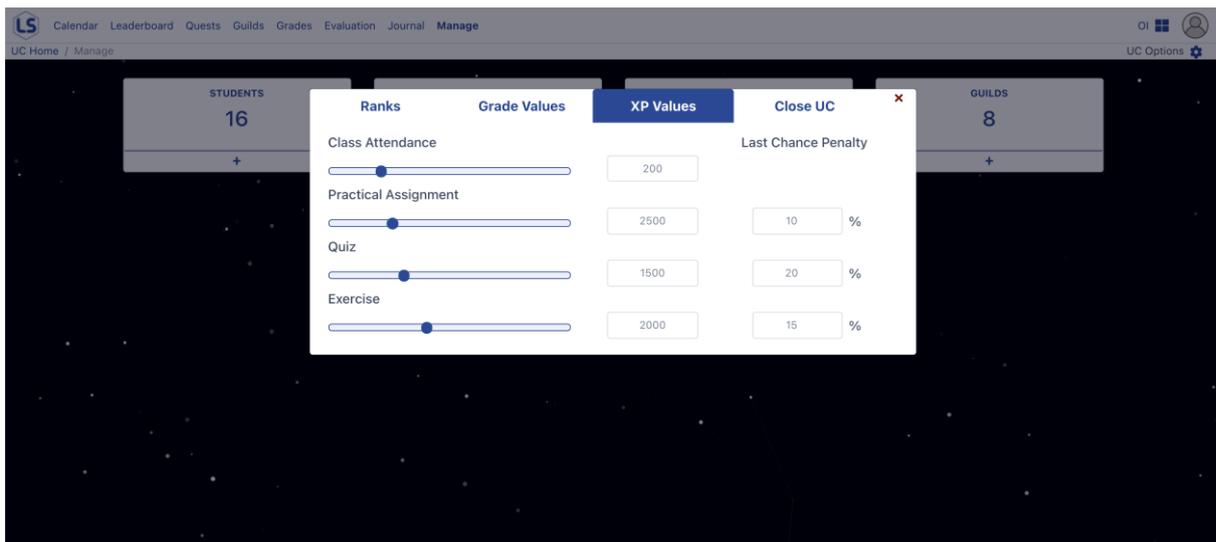


Figura 15: Tab XP Values do LS 2.0

#### 3.4.2.4 Notas

As notas foram um assunto bastante repensado nesta nova versão. Tal como os *ranks*, os valores das notas eram “*hardcoded*” com o mapeamento de A – F a ser feito com base em percentagens pré-definidas no código. No LS 2.0 o dinamismo volta a falar mais alto, tal como os *ranks*, as percentagens sobre o qual as notas são mapeadas são definidas pelo docente, através de uma interface própria, apresentada na Figura 16. É importante referir que as notas qualitativas (A – F) utilizadas para atribuição de XPs são apenas utilizadas em *quests* do tipo *Exercise* e *Practical Assignment*.

Da mesma forma que os *ranks* na categoria *Newbie*, a nota F é sempre cotada como 0% de forma que não existam discrepâncias no processo de avaliação pelo LS 2.0.

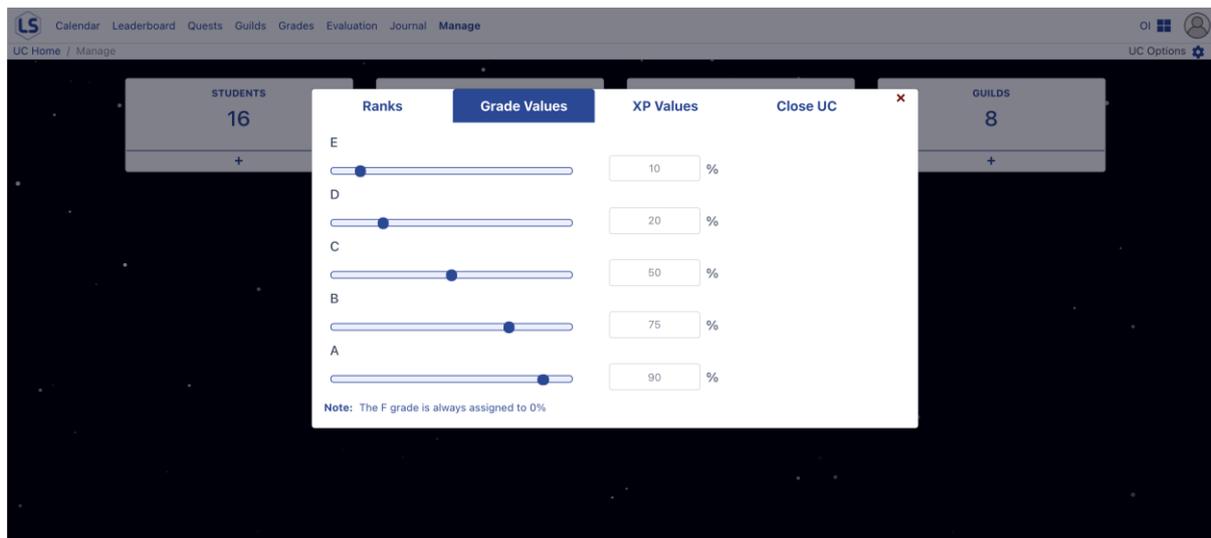


Figura 16: *Tab* Grade Values do LS 2.0

### 3.4.3 Guilds

As *guilds* representam os grupos de alunos designados para o trabalho de grupo. Tal como as *quests*, as *guilds* podem ser criadas de duas maneiras: em conjunto através da importação de um ficheiro *Excel* ou individualmente através de uma interface do LS 2.0. Nesta versão do LS, as funcionalidades relacionadas com as *guilds* foram melhoradas, especialmente o *feedback* do docente nas *quests* do tipo *Practical Assignment*, relacionadas com o trabalho de grupo.

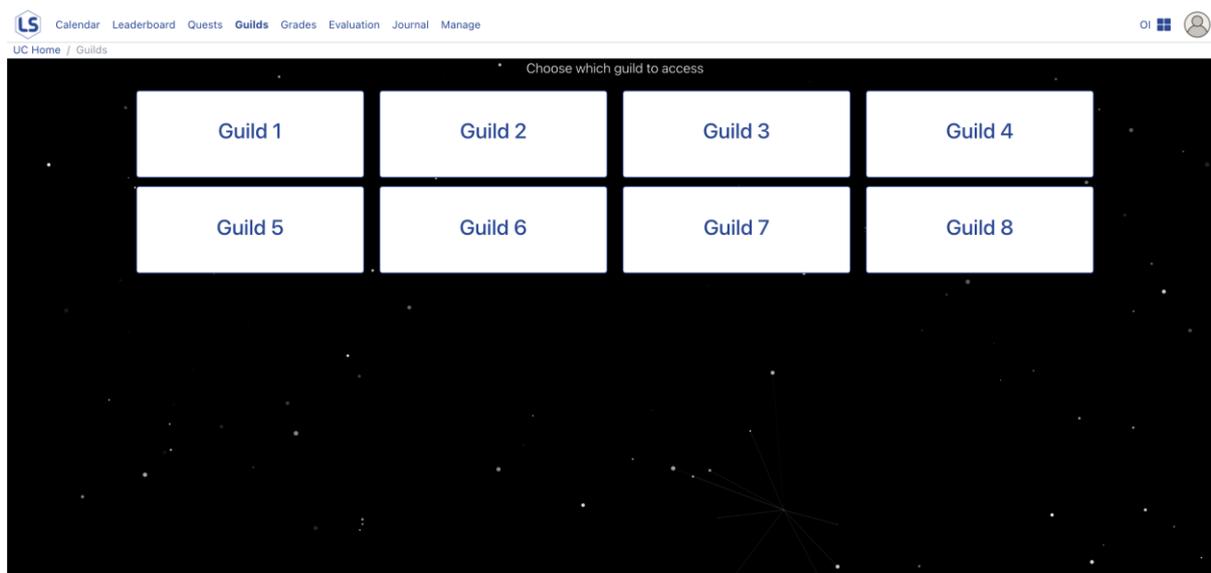


Figura 17: Visão geral das guilds da UC

#### 3.4.3.1 Visão geral do docente

De modo a gerir todas as *guilds* da UC de forma dinâmica e célere foi desenvolvida uma interface inicial para o docente quando entra na *tab* Guilds da barra de navegação.

Nesta visualização apresentada na Figura 17, cada *guild* é representada como um quadrado que, quando é clicado, leva o docente para a página da *guild*, contendo as informações sobre o trabalho de grupo e sobre os alunos.

### 3.4.3.2 Perfil da guild

O perfil da *guild* demonstrado na Figura 18 é uma visualização disponível tanto para o aluno como para o docente sendo que a única diferença é que o docente pode editar informações sobre a *guild*, tais como: o título do projeto, descrição do projeto e adição/edição de *feedback* relativos aos *practical assignments* realizados pela *guild*.

The screenshot shows the 'Guild 7' profile page. At the top, it displays 'Guild 7' and 'MEI'. Below this is a section for 'Title 7' and 'Project Description' with a text area containing a paragraph of placeholder text. Underneath is a table of members:

Username	Number	Total XPs	Continuous Assessment
To be defined	93888	0	No
miguel_student	93627	0	Yes

Below the members table is a section for 'Practical Assignments' with a dropdown menu set to 'AQ'. It contains a table of assignments:

Username	Number	Earned XPs	Grade
To be defined	93888	-	-
miguel_student	93627	1050	A

At the bottom, there is a 'Feedback' section with a text area containing the same placeholder text as the project description.

**Figura 18: Perfil da guild (vista de docente)**

Nesta página temos informação como o nome, *alliance*, título e descrição do trabalho de grupo e *practical assignments* realizados. Nos *practical assignments* temos informação como título, notas de cada membro que realizou o *practical assignment* e feedback por parte do docente sobre o mesmo. A *alliance* corresponde ao curso a que *guild* foi associada.

### 3.4.3.3 Feedback

Após as *guilds* serem criadas o docente pode aceder a cada uma delas e editar as informações relativas ao trabalho de grupo tais como o título e descrição do mesmo.

Tornou-se possível também a opção de o docente fornecer feedback relativo a cada um dos *practical assignments* existentes na unidade curricular. Quando se submetem as notas para um determinado *practical assignment*, o mesmo fica automaticamente disponível na página de cada *guild* que foi avaliada nesse *practical assignment* dando a opção ao docente de providenciar o feedback do mesmo. Tal como o docente, os alunos podem também visualizar as notas e o feedback correspondentes a cada um dos *practical assignments* associados à *guild* a que pertencem.

Através desta interface, apresentada na Figura 19, o docente pode submeter o feedback relativo aos *practical assignments* relativos à UC para uma determinada *guild* que fica depois visível para os alunos.

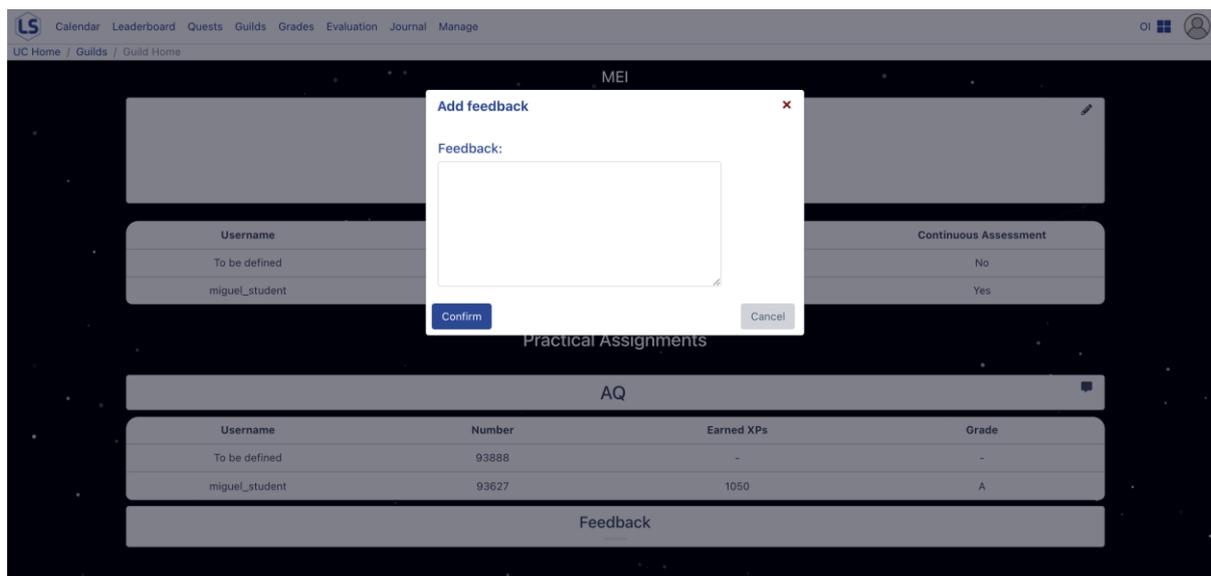


Figura 19: Interface de submissão de feedback

#### 3.4.3.4 Criação individual de guilds

Tal como as *quests*, o LS 2.0 oferece a oportunidade ao docente de criar *guilds* individualmente, em vez de submeter um ficheiro *Excel* para uma ou duas *guilds*.

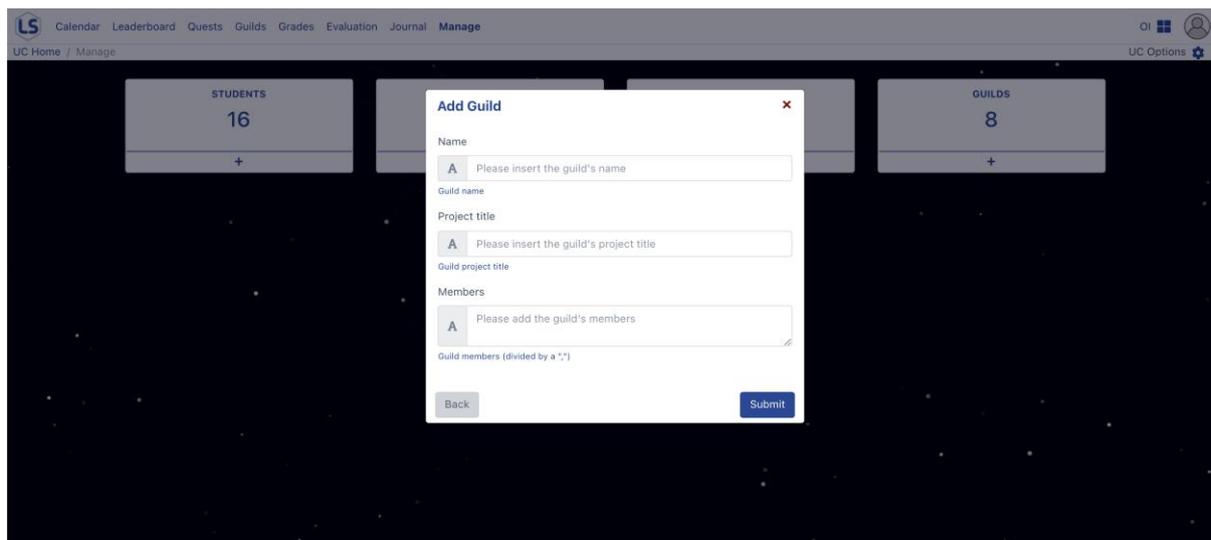


Figura 20: Interface de criação individual de *guild*

Nesta interface mostrada na Figura 20, o docente pode definir o nome da *guild* (único no âmbito de cada UC), o título do trabalho de grupo e os membros que constituem a *guild*, apenas necessitando de pôr os números dos alunos entre vírgulas.

### 3.4.4 Syllabus Contents

A nova adição ao LS 2.0 são os *Syllabus Contents* (conteúdos programáticos). Nesta plataforma os SCs representam os conteúdos lecionados numa unidade curricular e passam a ter um papel fundamental no acompanhamento da aprendizagem dos alunos por parte do docente. Não só o docente pode ter a perceção da dificuldade dos alunos nos conteúdos da unidade curricular como também o aluno passa a ter conhecimento de como se organizam os conteúdos da unidade curricular que o mesmo frequenta e um resumo da dificuldade sentida nos conteúdos lecionados na mesma.

#### 3.4.4.1 Hierarquia

Os SCs são organizados de forma hierárquica, em que a hierarquia pode ter  $N$  níveis, e cada SC tem o nível da hierarquia associado. É importante referir que como os SCs são armazenados no *GraphDB*, o que nos permite inferir sobre o conhecimento explícito que temos sobre os mesmos. Por exemplo, um SC de nível 1 (primeiro nível na hierarquia) contém dois SCs de nível 2 que por sua vez contém SCs de nível 3. Como referido acima, a propriedade *includes* dos SCs é transitiva, pelo que através de inferência temos que o SC de nível 1 inclui os SCs de nível 2 e os subsequentes de nível 3. No LS 2.0 esta hierarquia é representada através de um *mindmap*. A Figura 21 demonstra um exemplo dos conteúdos (SCs) que são ensinados numa unidade curricular.

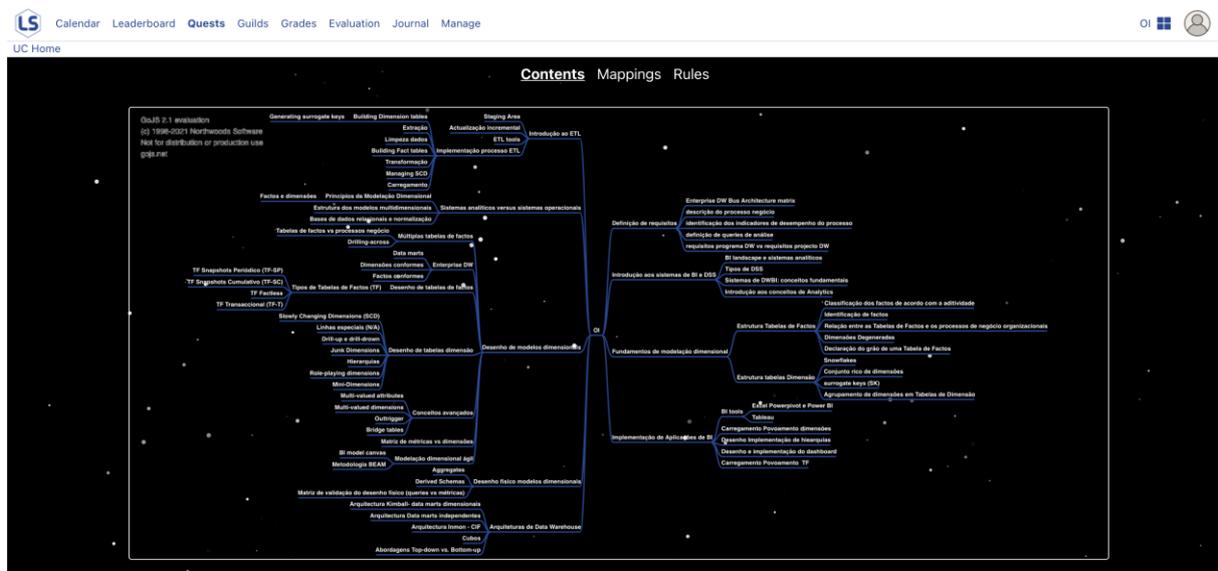


Figura 21: SCs de uma unidade curricular

Como podemos observar na Figura 21, a visualização escolhida, o *mindmap*, permite observar diretamente a organização hierárquica dos conteúdos de uma unidade curricular e entender quais as relações presentes entre os conteúdos. Mais adiante são demonstradas outras utilizações do *mindmap* por esta dissertação.

### 3.4.4.2 Mapeamento de conteúdos por quest

Um dos objetivos propostos para esta dissertação era de o docente ter a possibilidade de saber a dificuldade sentida pelos alunos ao longo do decorrer da execução de uma UC. Para concretizar este objetivo temos que primeiro mapear quais os conteúdos que são necessários para os alunos desempenharem com sucesso cada *quest*. Para criar este mapeamento, o docente tem que gerar a matriz de mapeamento, i.e., uma matriz com os conteúdos como linhas e as *quests* como colunas como podemos observar na Figura 22. Nesta matriz, apenas os SCs de nível 2 são tidos em consideração, não só pelas questões referidas na secção seguinte, mas também para evitar a complexidade e extensividade da matriz.

Contents / Quests	Quiz 1	Quiz 2
Enterprise DW Bus Architecture matrix		
requisitos programa DW vs requisitos projecto DW		
descrição do processo negócio		
identificação dos indicadores de desempenho do processo		x
definição de queries de análise		
Matriz de validação do desenho físico (queries vs métricas)		
Derived Schemas		
Aggregates		
Desenho Implementação de hierarquias		
BI tools		
Carregamento Povoamento TF	x	

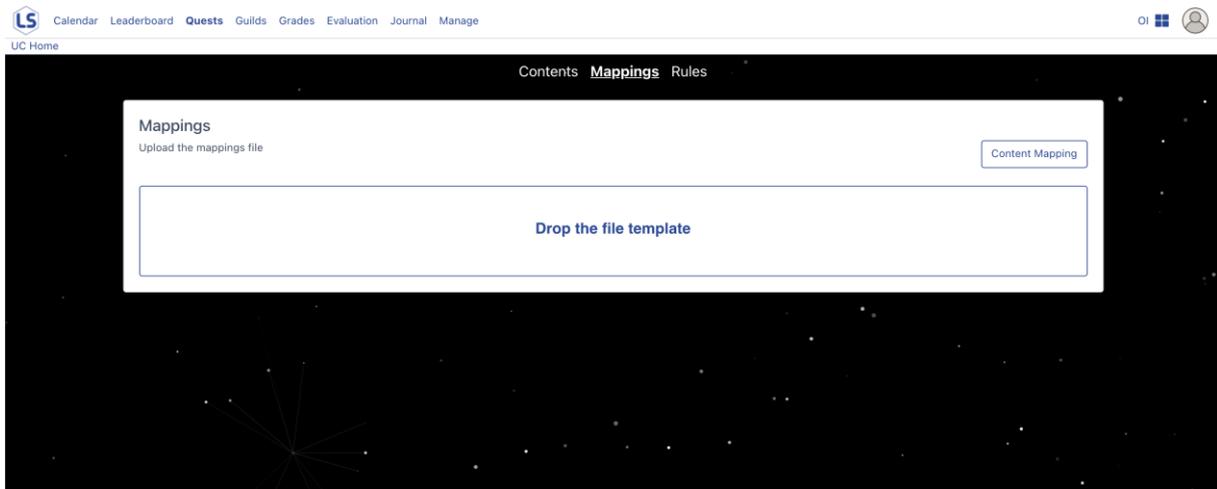
Figura 22: Matriz de mapeamento de SCs por *quest*

Através do preenchimento desta matriz o docente fornece a informação relativamente às relações entre SCs e *quests*. O mapeamento é uma funcionalidade que pode ser utilizada por todos os tipos de *quest* (quer tenham de ser validadas ou não), pois representam uma parte da metodologia de ensino da UC. Desta forma, faz sentido averiguar o impacto desta funcionalidade para o aluno e para o docente para cada tipo de *quest*.

**Class Attendance** – o aluno passa a ter acesso a informação mais detalhada do que o sumário sobre os conteúdos lecionados numa aula como também a possibilidade de o mesmo estudar autonomamente em caso de faltar a uma aula. Já o docente consegue organizar melhor a distribuição de SCs pelas aulas, de maneira a simplificar a aprendizagem dos alunos.

**Quiz, Practical Assignment e Exercise** – o aluno pode saber quais os SCs que deve estudar para realizar a *quest*, enquanto o docente passa a saber quais os conteúdos em que os alunos têm mais dificuldade através do mecanismo de validação de *quests*.

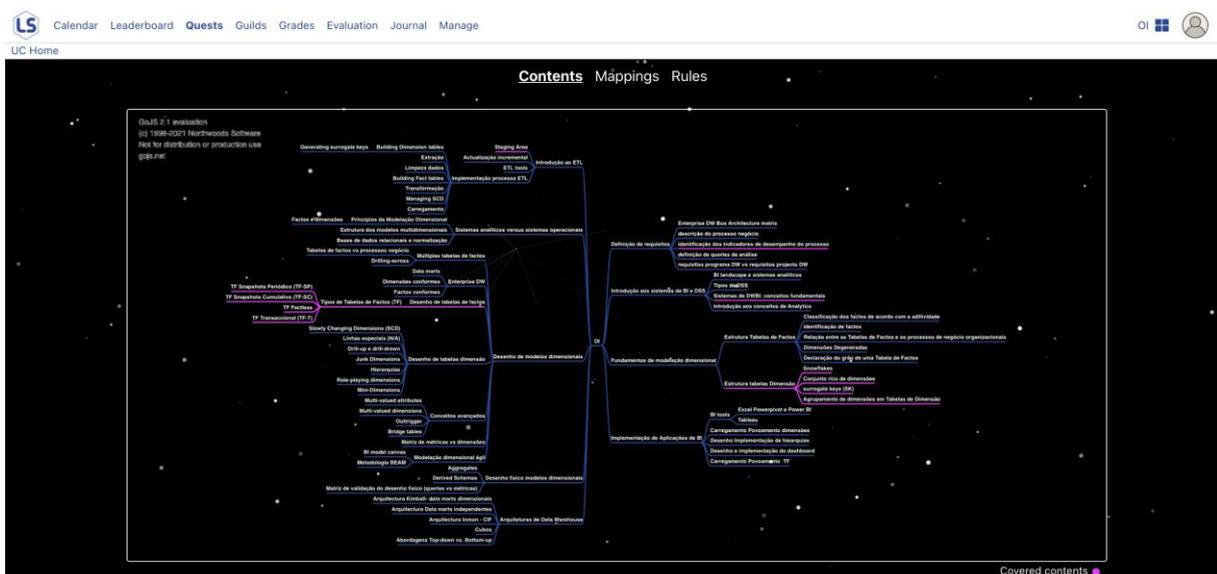
**Event** – este tipo de *quest* permite ao aluno saber a quais os SCs que serão abordados num determinado evento, de modo a estudar ou aprofundar o conhecimento sobre os mesmos.



**Figura 23: Interface de submissão do mapeamento de SCs com quests**

Para efetuar o mapeamento dos SCs com as *quests* o docente tem então que submeter a matriz de mapeamento preenchida através de uma interface específica no LS (ver Figura 23). Nesta interface, a matriz é processada e toda a informação relacionada com o mapeamento é depois submetida no *GraphDB*, de modo a criar as relações entre SCs e *quests* bem como inferir novas. De seguida, as relações de mapeamento para a UC ficam disponíveis para visualização.

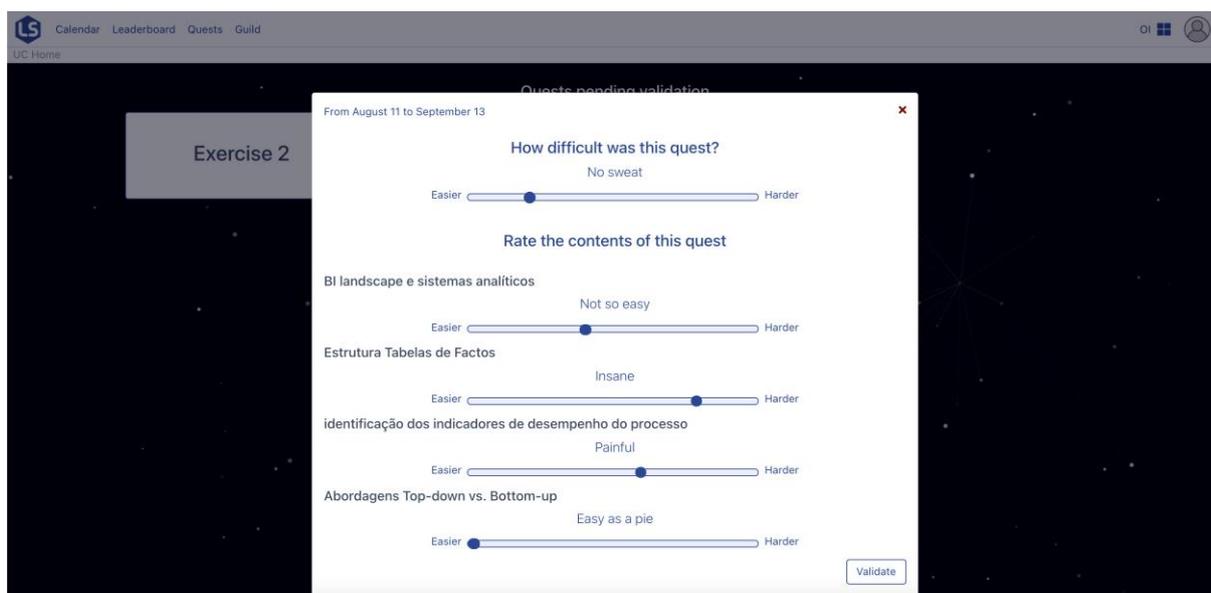
Na Figura 24 podemos observar quais os conteúdos que são abordados numa determinada *quest*, deste modo o aluno sabe quais os conteúdos que são lecionados ou que deve de estudar numa determinada *quest*.



**Figura 24: Visualização do mapeamento entre SCs e uma determinada quest**

### 3.4.4.3 Dificuldade sentida pelos alunos

Outra nova funcionalidade disponibilizada com a adição dos SCs é a visualização por parte do docente da dificuldade sentida pelos alunos relativamente aos conteúdos da UC. Devido ao número de conteúdos poder ser extenso, como também os níveis hierárquicos não terem limite, foi decidido que apenas seriam avaliados os conteúdos de nível 2. Esta decisão é um compromisso: por um lado, permite que o docente possa ter informação específica e ao mesmo tempo abrangente, e por outro, facilita a avaliação da dificuldade por parte dos alunos. Pedir aos alunos para validar todos os SCs poderia resultar numa grande quantidade de conteúdos para avaliar, isto se as UCs forem definidas com muito detalhe. De frisar que o aluno apenas pode validar *quests* do tipo *Quiz*, *Exercise* e *Practical Assignment*.



**Figura 25: Interface de avaliação da dificuldade sentida pelo aluno**

A Figura 25 apresenta a interface de avaliação de uma *quest* onde o aluno pode avaliar a *quest* em geral, bem como os conteúdos inerentes à mesma. A dificuldade é definida de 0 (mais fácil) a 5 (mais difícil) e codificada por frases. De notar que as frases associadas a cada valor, nomeadamente, *Easy as a pie* (0); *No sweat* (1); *Not so easy* (2); *Painful* (3); *Insane* (4); *Overpowered* (5), são as mesmas utilizadas nas versões anteriores do LS (Pedroso, 2018).

Após efetuar a validação da dificuldade sentida, o aluno recebe os XPs que lhe foram atribuídos pelo seu desempenho na *quest*. A partir desta validação por parte dos alunos, o docente pode assim visualizar a dificuldade média sentida pelos alunos em cada SC.

Contents Mappings Rules

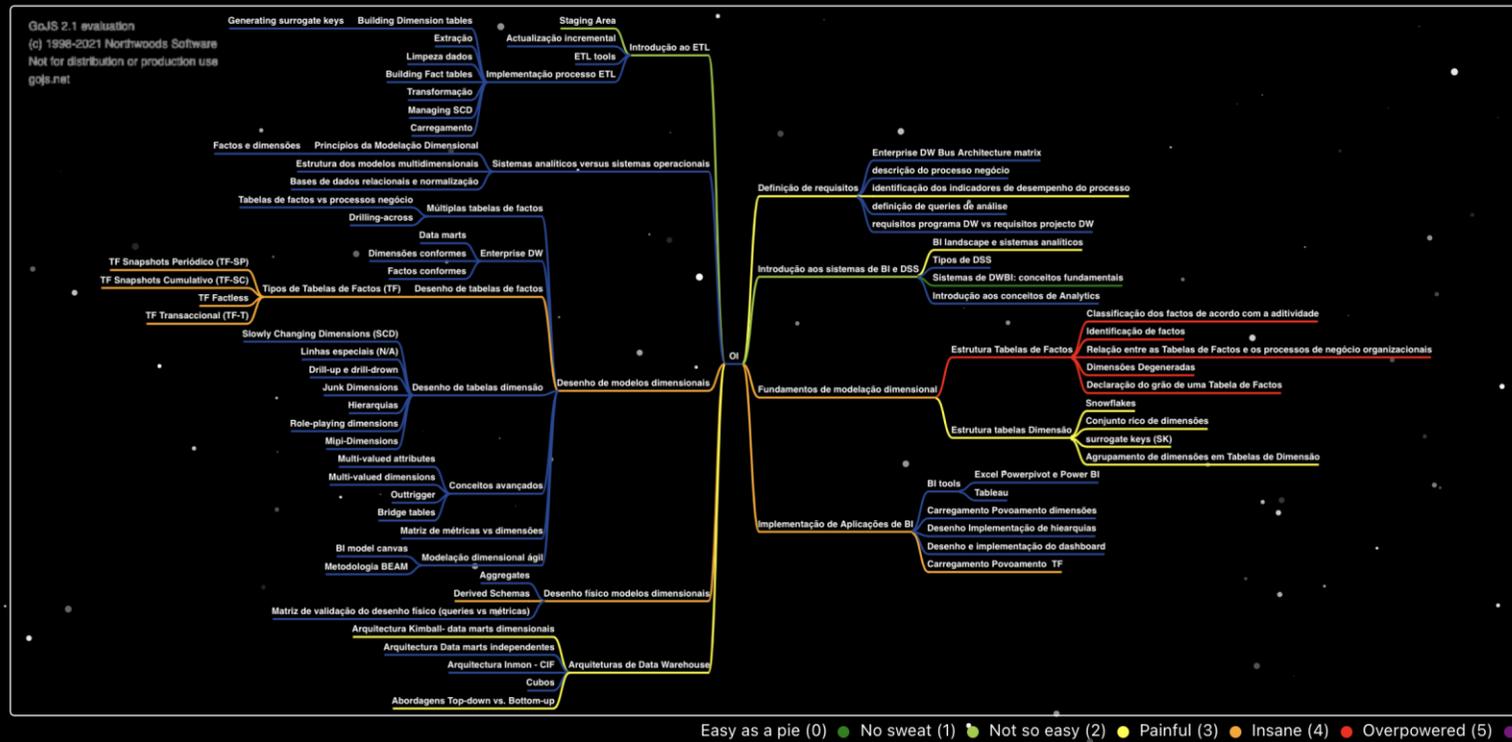


Figura 26: Interface de visualização da dificuldade sentida pelos alunos nos SCs de uma UC

Na Figura 26, é apresentado ao docente o componente *mindmap* do LS 2.0 que alia a visualização hierárquica dos SCs com a dificuldade sentida pelos alunos nos mesmos com o mapeamento da dificuldade feito por cores a ser apresentado no segmento inferior do *mindmap*. Esta visualização existe tanto para o docente (dificuldade média da turma) como para o aluno (dificuldade individual) e deste modo o utilizador pode tomar decisões para melhorar a aprendizagem dos conteúdos com maior dificuldade sentida.

Como apenas se avaliam os conteúdos de nível 2, a dificuldade dos restantes é processada da seguinte maneira: a dificuldade dos conteúdos de nível 1 é calculada pela média das dificuldades dos conteúdos de nível 2 englobados que já foram avaliados, os conteúdos de nível 3, 4, *N* herdam a dificuldade dos conteúdos de nível 2 que os engloba. Resta referir que a dificuldade avaliada pelo aluno num determinado SC pode não ser imutável, caso exista outra *quest* relacionada com o mesmo SC quando o aluno avaliar os SCs dessa *quest* o valor da dificuldade sentida nesse tal SC é atualizado de modo a oferecer dinamismo ao processo de avaliação para que o docente possa também gerir a UC de acordo com as dificuldades que os alunos vão sentido no decorrer da UC.

### 3.4.5 Diferenciação entre vistas de aluno e docente

Nas versões antigas do LS a forma escolhida para diferenciar as vistas de aluno e docente foi inverter as cores da barra de navegação. A barra de navegação antiga continha muita informação a ser apresentada em simultâneo. No LS 2.0, optou-se por colocar certos componentes da barra num menu *dropdown*, embutido no *avatar* do utilizador que surge quando se clica no mesmo.



**Figura 27: Comparação da barra de navegação das vistas de aluno e docente entre a versão 4 e o LS 2.0**

Nesta nova versão, a barra de navegação foi alvo de várias alterações visuais e funcionais. Em vez de inverter por completo as cores da barra de navegação, no LS 2.0 optou-se por minimizar a diferença visual entre as duas vistas, fazendo-o de forma subtil através da inversão do símbolo do LS. É de frisar que esta nova versão inclui uma *breadcrumb*, que é barra de menor dimensão situada abaixo da barra de navegação, esta adição permite ao utilizador navegar facilmente pela plataforma e saber onde o mesmo se encontra. A Figura 27 permite visualizar estas diferenças.

### **3.4.6 Calendar**

Ao contrário das versões antigas que apenas apresentavam o planeamento da UC por semanas, nesta nova versão do LS os alunos podem visualizar a ordem cronológica das *quests* de uma unidade curricular de duas maneiras: através de um calendário ou de uma linha cronológica. Estas duas opções surgiram pelo facto de oferecer, tanto ao aluno como ao docente, diferentes formas de se poderem organizar da maneira que melhor preferirem.

A Figura 28 apresenta estas duas visualizações. Na visualização *Calendar*, o utilizador pode ver o planeamento mensal para a UC que engloba todas as *quests* da mesma. Na visualização *Timeline*, o utilizador tem a opção de ver o seguimento cronológico das *quests* da UC com informação sobre título, sumário e datas de começo e conclusão da *quest*.

### **3.4.7 Leaderboards**

As *leaderboards* permitem aos alunos visualizarem o seu progresso na UC através da quantidade de XPs que os mesmos acumulam durante a execução da UC. O docente também tem acesso a esta visualização, mas de uma forma mais geral. Nas *leaderboards* o utilizador pode escolher entre várias classificações como: total, em *quizzes*, em *exercises* e por *guilds*.

Nesta versão do LS, as *leaderboards* foram desenvolvidas com um design mais subtil. Como podemos observar na Figura 29, a *leaderboard* passa a identificar a posição dos três melhores jogadores através da cor da *border* (ouro, prata e bronze) para além dos avatares estarem posicionados em formato de pódio com a dimensão de cada avatar a ter mais destaque quanto melhor a classificação do aluno. É importante referir que no LS 2.0, o aluno consegue ver a sua posição com maior destaque através da aplicação de uma sombra azulada para o identificar na *leaderboard*.

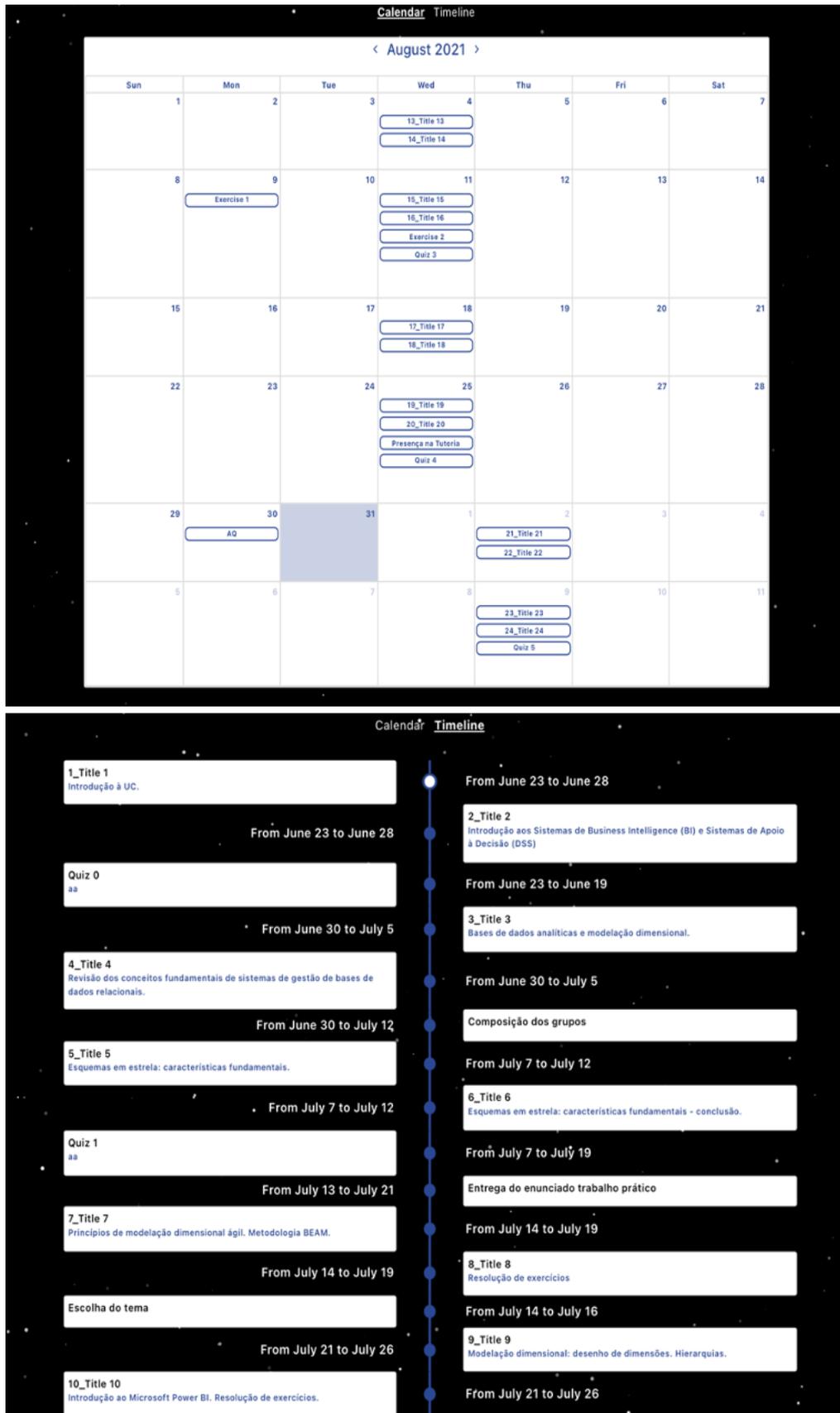


Figura 28: Em cima, o componente *Calendar* e, em baixo, o componente *Timeline* do LS 2.0

ALL QUIZZES EXERCISES GUILDS COMBINED RULES

Position	Name	XP's
1	<b>JoaoPedro</b> Legendary	36150
2	<b>JoaoBone</b> Master	34700
3	<b>f2196</b> Master	33400
4	<b>JoséM</b> Master	32850
5	<b>John Doe</b> Master	31800

All Quiz Exercise Guilds

#	Username	Alliance	Rank	XP
★	miguel_student	MEI	Newbie	0
★	To be defined	MEI	Newbie	0
★	To be defined	IGE	Newbie	0
4	To be defined	ABC	Newbie	0
5	To be defined	SLB	Newbie	0
6	To be defined	MEI	Newbie	0
7	To be defined	IGE	Newbie	0
8	To be defined	ABC	Newbie	0
9	To be defined	SLB	Newbie	0
10	To be defined	MEI	Newbie	0

★ miguel\_student MEI Newbie 0

Figura 29: Leaderboards no LS v4 (em cima) e no LS 2.0 (em baixo)

### 3.4.8 Grades

No LS 2.0 a *tab* Grades fornece ao docente duas funcionalidades: a de submeter as folhas de presenças e a de submeter as notas das *quests* realizadas (ver Figura 30).

Para a primeira funcionalidade, é necessário submeter um ficheiro *Excel* em que uma linha corresponde a uma presença e cada linha contém o número do aluno e o título da *Class Attendance*. É importante referir que caso o docente extraia as presenças de outro sistema (como o Fénix por exemplo) tem de certificar-se de que o título da *quest* no LS corresponde ao valor obtido pelo sistema de presenças.

Para a segunda funcionalidade, o docente apenas necessita de preencher um ficheiro *Excel* com linhas compostas por número de aluno e valor (sempre em percentagem) da nota que o aluno teve. Esta funcionalidade permite que o docente possa distribuir as avaliações dos alunos para que estes as possam posteriormente validar e receber os XPs associados às mesmas.

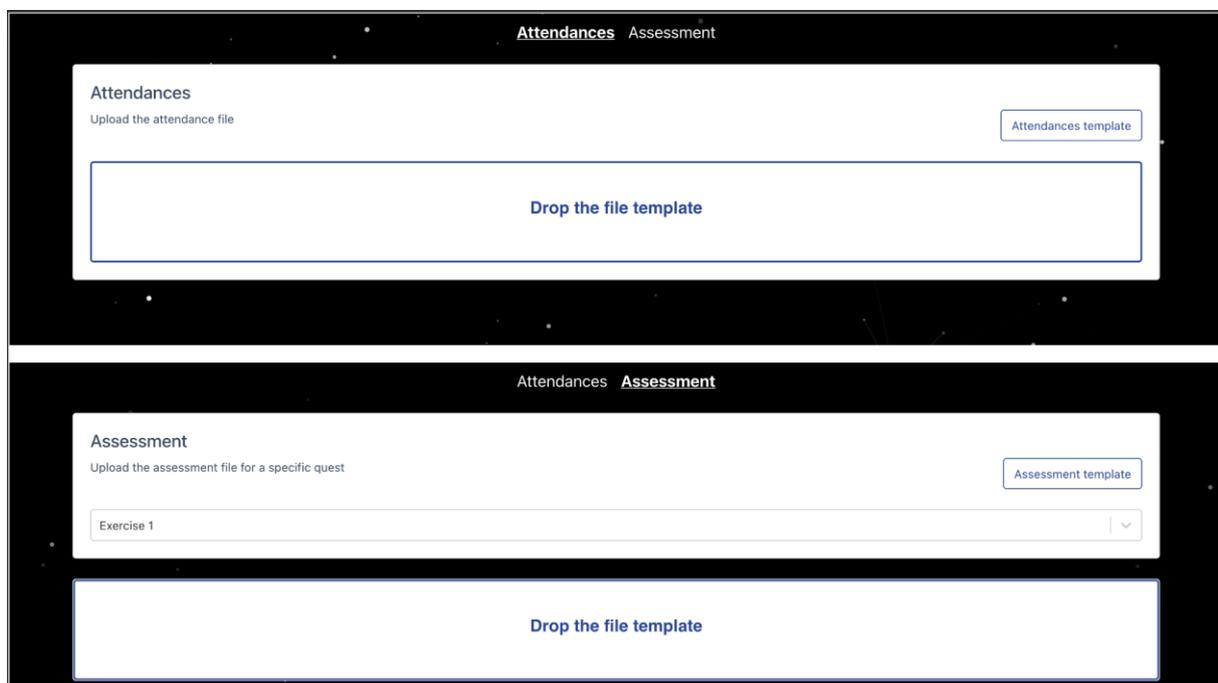
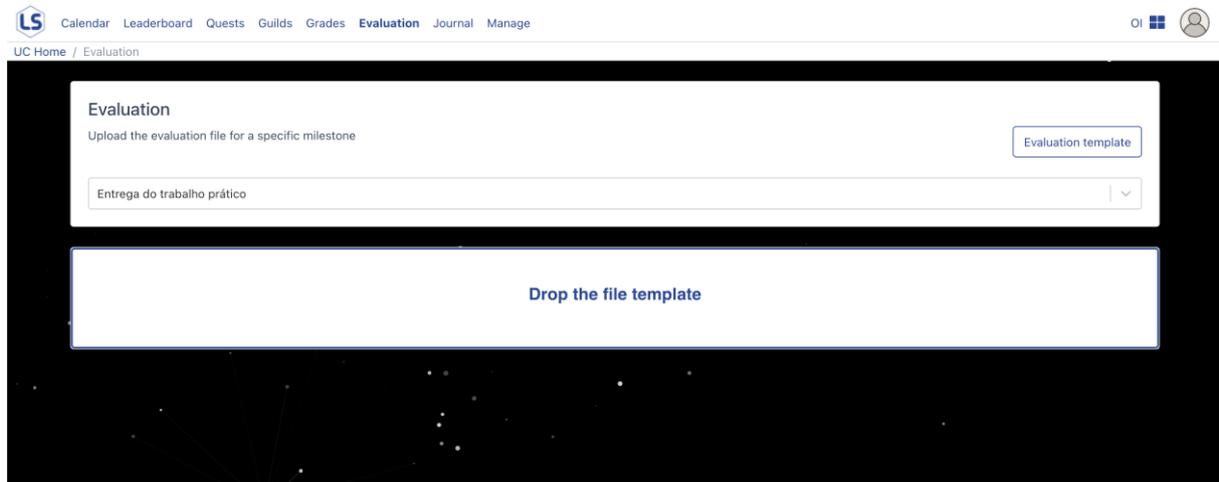


Figura 30: Interface de submissão de presenças (em cima) e de distribuição de notas

### 3.4.9 Evaluation

Como referido acima, o LS contém *quests* obrigatórias. Estas *quests* são na verdade *milestones*, i.e, objetivos importantes a cumprir durante o decorrer da unidade curricular. A particularidade das *milestones*, é que caso o aluno falhe alguma, perde o jogo, deixando de estar em avaliação contínua. No LS 2.0 a funcionalidade de avaliação foi repensada, pois os eventos, que eram na

sua maioria *milestones*, passaram a ser um tipo de *quest*. O docente pode assim seleccionar qual a *milestone* a ser avaliada e através da importação de um ficheiro *Excel* (ver Figura 31) definir quais os alunos que passaram a *game over*, i.e., que não permanecem em avaliação contínua no LS. Este ficheiro *Excel* contém informação sobre o número de aluno e se o mesmo passou ou não na *milestone* (através das letras Y/N – *Yes/No*).



**Figura 31: Interface *Evaluation* do LS 2.0**

### 3.4.10 *Journal*

Tal como nas antigas versões do LS, a funcionalidade *Journal* permite ao docente visualizar todos os XPs atribuídos aos alunos por *quest*, de modo a servir como um “diário” da UC. Nesta versão a interface foi melhorada de modo a ser mais legível e permitir que o docente possa ter uma visão mais detalhada sobre o desempenho de cada aluno numa determinada *quest* (ver Figura 32).

Quiz 2			
Username	XPs	Grade	Validated
miguel_student	1500	100 %	No
To be defined	1425	95 %	No
To be defined	1200	80 %	No
To be defined	750	50 %	No
To be defined	1050	70 %	No

Exercise 2			
Username	XPs	Grade	Validated
miguel_student	2000	A	No
To be defined	1900	A	No
To be defined	1600	B	No
To be defined	1000	C	No
To be defined	1400	C	No

**Figura 32: Interface *Journal* do LS 2.0**

### 3.4.11 Vista de Administração

A vista de Administração é a nova vista no LS, desenvolvida para um novo tipo de utilizador, o administrador. Nesta vista, o utilizador pode criar unidades curriculares bem como efetuar toda a gestão relativa às mesmas. O mesmo acontece para toda a gestão dos utilizadores, docentes e alunos, como também para a restante gestão da plataforma. Esta vista surgiu da necessidade de haver uma entidade responsável pela manutenção da plataforma, de modo a auxiliar alunos e docentes em potenciais situações que necessitam de uma intervenção com mais privilégios de utilização.

#### 3.4.11.1 UCs

Como podemos ver na Figura 33, o administrador pode aceder à *tab* UCs para realizar a gestão de todas as UCs presentes no LS. Em cada linha da tabela das UCs o administrador pode expandir a mesma para revelar mais informação sobre a unidade curricular escolhida como o número de estudantes, docentes, *guilds* e *quests* como também o nome, acrónimo, ano letivo e estado (ativa ou fechada) da mesma. Pode também aceder à página de gestão da unidade curricular de modo a efetuar toda a gestão da mesma. O administrador é o único que pode apagar unidades curriculares, ao contrário do que acontecia nas versões anteriores em que o docente tinha acesso a esta funcionalidade.

The screenshot displays the 'UCs' management interface. At the top, there's a navigation bar with 'LS', 'UCs', 'Teachers', and 'Students'. Below it, a filter input field and 'Items per page: 5' are visible. The main table has columns: Name, Acronym, Academic year, Semester, and Status. The first row shows 'Guild 4' with acronym 'UC10', year '2020/2021', semester '2', and status 'Active'. This row is expanded to show 'UC10 - Guild 4' with statistics: Students: 0, Teachers: 0, Guilds: 0, and Quests: 78. Below the expansion, a list of sub-UCs is shown:

Name	Acronym	Academic year	Semester	Status
DATA WAREHOUSE AND BUSINESS INTELLIGENCE SYSTEMS	DWBIS	2021/2022	1	Active
SDWBI	SDWBI	2021/2022	2	Active
TESTT	TESTT	2020/2021	2	Active
TOP	TOP	2020/2021	2	Active

At the bottom, there's a pagination bar showing page 1 of 2, and an 'Add UC' button.

Figura 33: Interface de gestão de UCs na vista de administração

### 3.4.11.2 Teachers

Na *tab* Teachers o administrador pode efetuar a gestão dos docentes como criar e apagar os mesmos. Nesta *tab* está também disponível a informação sobre o nome do docente e unidades curriculares a que o mesmo está associado.

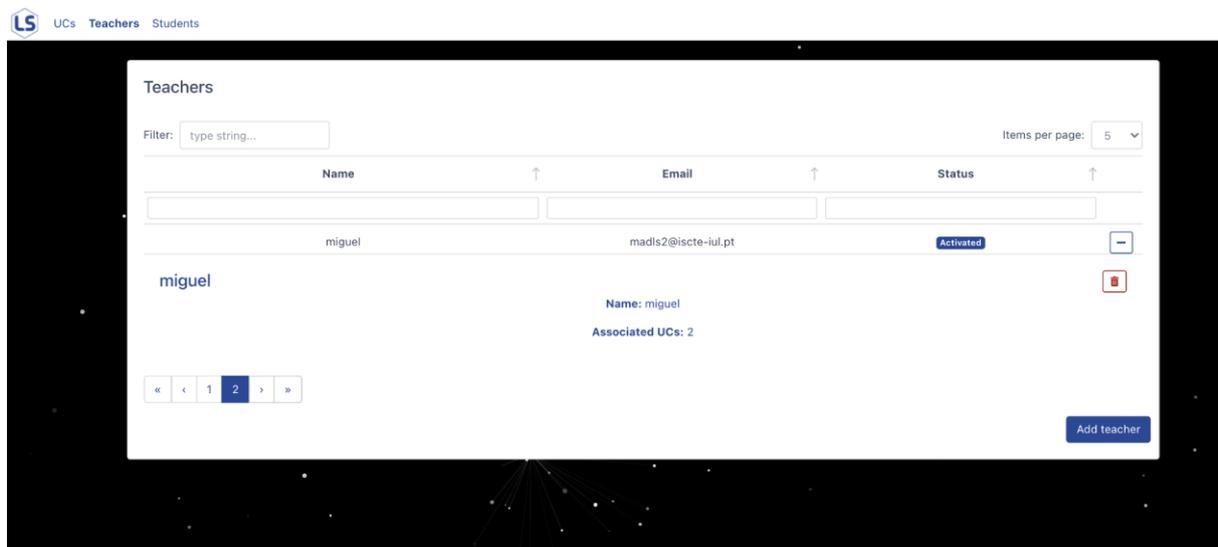


Figura 34: Interface de gestão de docentes na vista de administração

### 3.4.11.3 Students

Na *tab* Students o administrador pode visualizar os alunos existentes em toda a plataforma, podendo eliminá-los. Caso queira criar alunos ele terá que aceder a uma UC através da *tab* UCs e importá-los.

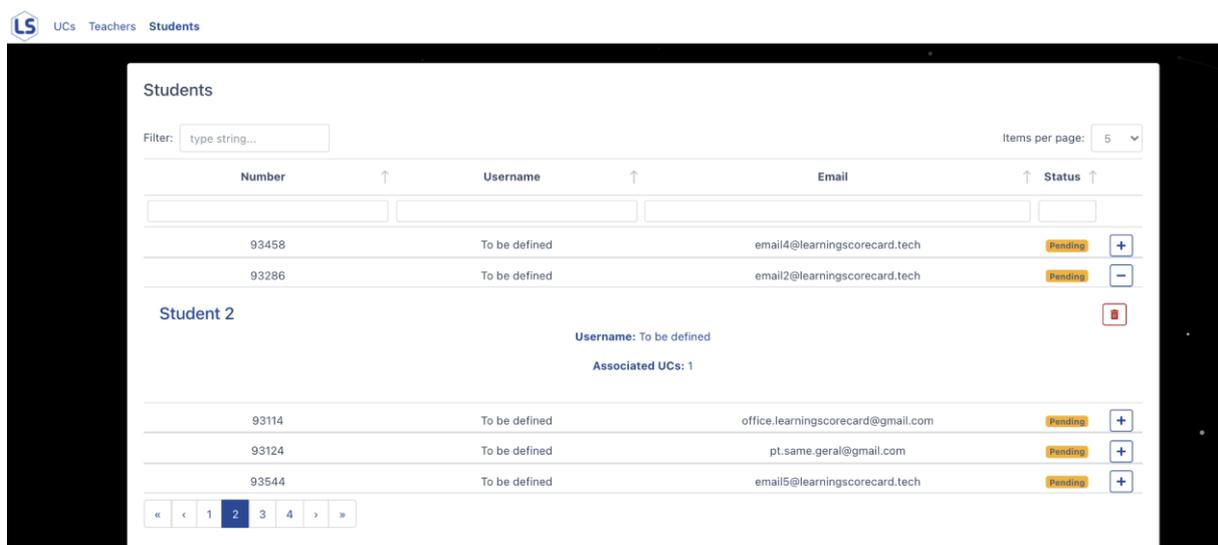


Figura 35: Interface de gestão de alunos na vista de administração



## Capítulo 4 Validação

Este capítulo está relacionado com a etapa *Evaluation* da metodologia DSR. A validação dos artefactos produzidos nesta dissertação foi realizada com recurso a um questionário. É importante referir que um dos objetivos deste questionário passa por introduzir os alunos à plataforma *Learning Scorecard*, são apresentadas as interfaces das novas funcionalidades da mesma e no final os alunos são questionados se a utilizariam no seu dia-a-dia académico. Pretende-se assim utilizar este questionário para auxiliar a resposta às questões de investigação desta dissertação.

De modo a validar todos os desenvolvimentos feitos no âmbito desta dissertação foram questionados 75 alunos através da plataforma *Google Forms*. Neste questionário as questões foram elaboradas com o intuito de compreender o que pensam os alunos sobre as novas funcionalidades da plataforma. O questionário está organizado em quatro secções de perguntas: caracterização dos respondentes, hábitos académicos, experiência de aprendizagem e avaliação da dificuldade de aprendizagem.

Na primeira secção do questionário os alunos respondem a perguntas que nos permitam analisar em detalhe a caracterização dos respondentes. Questões como idade, género, ciclo de estudos que frequenta, o curso e o ano curricular em que se encontram inscritos no semestre de 2021/2022 bem como se pertencem ou não ao Iscte. O questionário foi enviado a alunos que se encontram matriculados num curso de Ensino Superior no ano de 2021/2022.

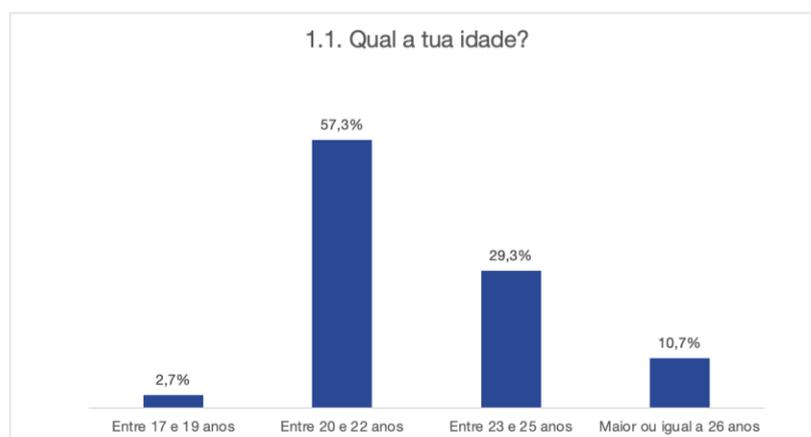
Na segunda secção o objetivo centra-se em entender melhor os hábitos académicos dos alunos, tais como a experiência que têm na utilização dos sistemas académicos (*E-learning*, sistema de gestão de ensino, etc.), a maneira de como os alunos acedem aos conteúdos disponibilizados nas UCs, bem como lidam com os mesmos.

Na terceira secção os alunos são questionados sobre a experiência de aprendizagem nas UCs. É apresentada a interface relativa ao mapeamento dos conteúdos com as *quests* da plataforma LS 2.0 e os alunos avaliam a sua utilidade bem como identificam pontos positivos e negativos que encontrem nesta interface.

Por último, na quarta secção os alunos respondem a perguntas relativas à utilidade da interface para a avaliação e visualização da dificuldade de aprendizagem dos SCs numa UC. São também questionados sobre se utilizariam esta plataforma como auxílio à preparação para as atividades de uma UC.

## 4.1 Caracterização dos respondentes

Na Figura 36 podemos concluir que neste estudo a maioria dos alunos situam-se entre a faixa etária dos 20 aos 22 anos (57.3%), seguida pela faixa entre os 23 e 25 anos (29.3%). Apenas 10.7% dos alunos têm 26 ou mais anos de idade e 2.7% dos alunos situam-se na idade compreendida entre os 17 e 19 anos. Podemos ver que pela idade a maioria dos respondentes já devem de ter tido contacto com outras plataformas de ensino, como por exemplo, *E-learning* ou outros sistemas de gestão de ensino, o que é uma mais-valia para o âmbito desta dissertação.



**Figura 36: Distribuição das idades dos alunos questionados**

No total, 43.2% dos estudantes questionados são do género feminino e 56.8% do género masculino.

Um aspeto positivo deste questionário é que foi respondido por alunos de várias áreas científicas e de outras universidades para além do Iscte. Isto é uma mais-valia, pois em todas as versões anteriores foram apenas questionados alunos das UCs de Sistemas Informáticos de Apoio à Decisão I e II do Iscte. Isto permite-nos ter acesso a opiniões mais variadas de alunos de vários cursos já que o LS 2.0 tem como objetivo ser utilizado por qualquer curso.

Nesta dissertação foram questionados alunos de diferentes ciclos de estudos, nomeadamente do primeiro e segundo ciclo (licenciaturas e mestrados respetivamente). Relativamente às licenciaturas, a área de estudo com mais repostas é Ciências e Tecnologias de Informação com 44% dos alunos, seguindo-se a área de Ciências Sociais e Humanas com 14.7% dos alunos. Já Economia e Gestão constitui 5.3% do número total de estudantes questionados.

Para os mestrados a área predominante foi a de Ciências e Tecnologias de Informação com 22.7% de inquiridos seguida por Economia e Gestão com 8%. Neste ciclo a área de Ciências Sociais e Humanas integra 5.3% das respostas dos alunos questionados. Podemos observar melhor esta distribuição através da Tabela 3. É importante referir que de todas as respostas recebidas 60% são alunos do Iscte.

**Tabela 3: Distribuição das respostas por ciclo e área de estudo**

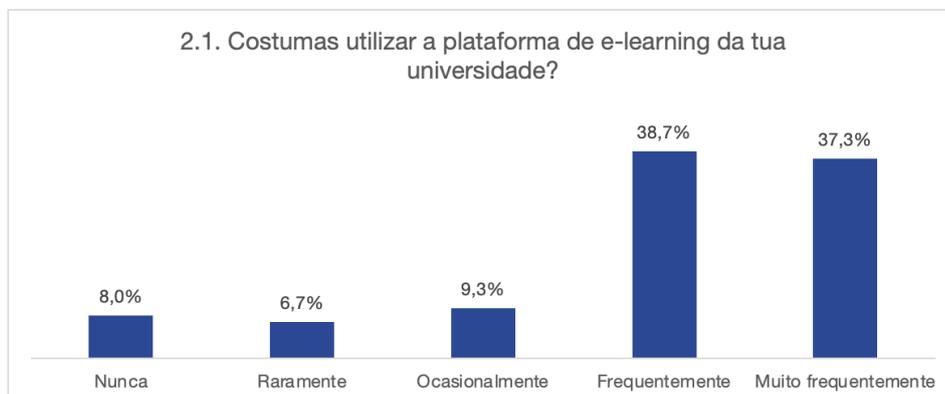
Ciclo de Estudos	Área de Estudo	Nº de Respostas	% de Respostas
1º ciclo (licenciatura)	Ciências e Tecnologias de Informação	33	44,0%
	Ciências Sociais e Humanas	11	14,7%
	Economia e Gestão	4	5,3%
2º ciclo (mestrado)	Ciências e Tecnologias de Informação	17	22,7%
	Ciências Sociais e Humanas	4	5,3%
	Economia e Gestão	6	8,0%

## 4.2 Hábitos académicos

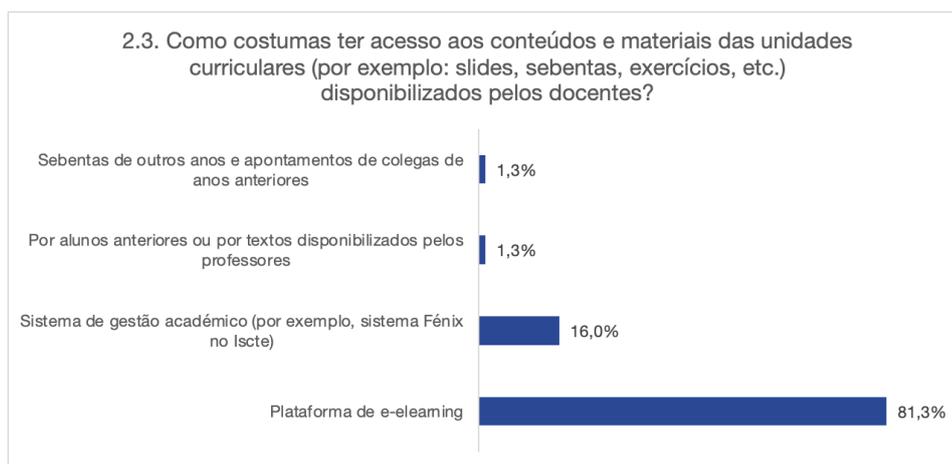
Relativamente à utilização de plataformas de *E-learning*, mais propriamente à frequência, podemos dizer que os alunos na sua maioria utilizam frequentemente as plataformas (38.7%), os que utilizam muito frequentemente compõem 37.3% das respostas. 9.3% dos participantes afirmam utilizar ocasionalmente a plataforma de *E-learning*, já em minoria estão os alunos que raramente (6.7%) ou nunca (8%) utilizam as plataformas. Na Figura 37 são demonstradas as respostas dos participantes.

Quanto ao que cada aluno sente que falta na plataforma da sua universidade as opiniões convergem com base na frequência de utilização. Os alunos que utilizam frequentemente e muito frequentemente o *E-learning* sentem falta de uma melhor interação aluno-docente bem como uma plataforma mais intuitiva, duas coisas que o LS 2.0 pretende resolver. Necessidades como uma melhor organização da plataforma, apresentação de conteúdos e acesso a notas foram também opiniões frequentes nesta categoria. Os alunos que utilizam a plataforma de *E-learning* ocasionalmente referem mais a necessidade de um aumento de conteúdos disponibilizados na mesma e reforçam a necessidade de uma melhor interação entre alunos e docentes. Relativamente aos que raramente ou nunca utilizam, estes atribuem essa decisão à falha do funcionamento da plataforma, em termos de disponibilidade e rapidez.

Com isto podemos desde já perceber que as necessidades dos alunos correspondem a algumas melhorias efetuadas no LS 2.0. Em particular, o novo ambiente tecnológico do LS que permite aumentar a rapidez e disponibilidade, uma melhoria na interação entre o docente e o aluno através da avaliação de dificuldade dos conteúdos programáticos (SCs) e mapeamento entre *quests* e SCs. É importante referir que alguns alunos do Iscte sentem que se devia utilizar apenas uma plataforma na universidade em vez de duas, *E-learning* e Fénix, em que essa plataforma incluía informação relativa às duas.



**Figura 37: Frequência de utilização das plataformas de *E-learning***



**Figura 38: Distribuição dos acessos a materiais das unidades curriculares**

A Figura 38 demonstra a distribuição do acesso a conteúdos e materiais disponibilizados pelos docentes em que a maioria dos alunos (81.3%) tem acesso a esta informação via a plataforma *E-learning*, já 16% utilizam o sistema de gestão académico e os que utilizam sebatas e apontamentos de anos anteriores ocupam em conjunto 2.6% das respostas.

Esta questão revela que os alunos inquiridos dependem da plataforma de *E-learning* para terem acesso a este tipo de conteúdos e materiais sendo esta mais uma razão para a utilização do LS 2.0 já que oferece funcionalidades para que os alunos possam compreender melhor os conteúdos de uma UC.

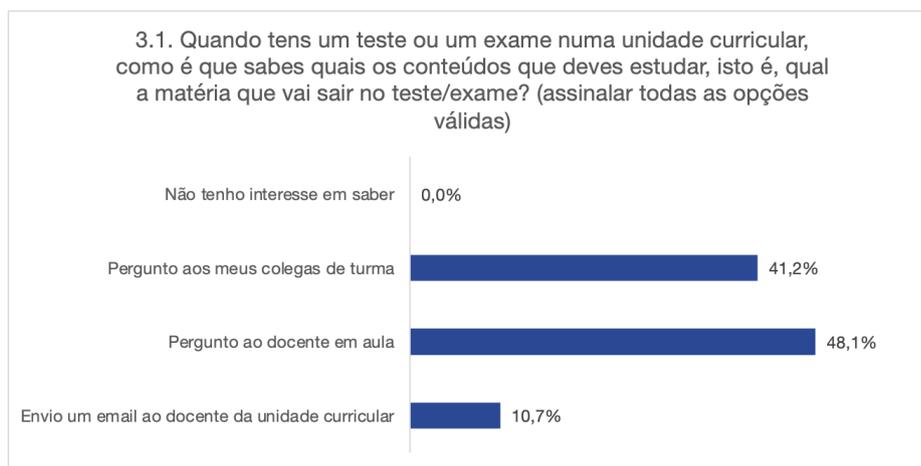
Quando questionados sobre como ficam a par da matéria dada numa aula quando faltam, os alunos na sua maioria (41%) recorrem à plataforma de *E-learning* para compensarem pela falta, o que demonstra mais uma vez a importância que o LS 2.0 pode ter na experiência académica dos alunos ao fornecer-lhes informação relativa ao mapeamento dos conteúdos por *quest* (aula, teste, exercício, etc.). A partilha de matéria entre alunos através de apontamentos também desempenha um papel relevante já que 36.8% dos inquiridos pedem apontamentos aos colegas quando não comparecem às aulas, apenas 6.3% dos alunos comunicam com os docentes a respeito deste assunto e 0.7% afirmam não fazer nada quando faltam (ver Figura 39).



Figura 39: Distribuição do acompanhamento da matéria quando não se comparece a uma aula

### 4.3 Experiência de aprendizagem

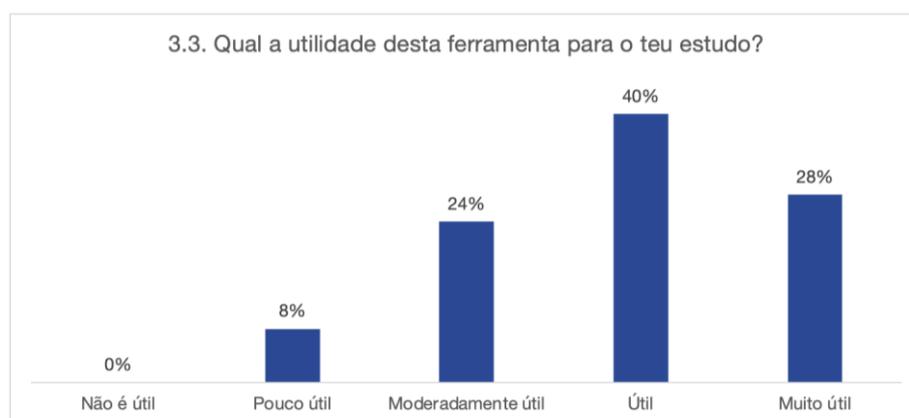
Os alunos foram também questionados sobre como sabem quais os conteúdos que devem estudar para um teste sendo que 58.8% (48.1% em aula, 10.7% por email) obtêm esta informação pelo docente e 41.2% pelos colegas (ver Figura 40). A interação entre aluno e docente é algo muito referenciado pelos alunos inquiridos e nesta nova versão do LS os docentes aproximam-se mais da experiência de aprendizagem dos alunos não só ao partilhar conhecimento relativo aos conteúdos e atividades (*quests*) como também a acompanharem mais detalhadamente aspetos como a dificuldade sentida em cada conteúdo pelos alunos.



**Figura 40: Distribuição dos alunos sobre como sabem quais os conteúdos para um teste**

Relativamente a se gostariam de ter acesso a uma ferramenta que identificasse qual a matéria relevante para todos os instrumentos de avaliação de uma unidade curricular que pudesse ser consultada de forma autónoma, 100% dos alunos respondeu que sim, o que permite averiguar que a funcionalidade do mapeamento de conteúdos do LS 2.0 desenvolvida nesta dissertação é de interesse para os alunos.

Quanto à utilidade dessa mesma ferramenta para auxiliar o seu estudo, 40% dos alunos consideram-na útil, 28% muito útil e 24% moderadamente útil, apenas 8% dos alunos sente que é pouco útil (ver Figura 41). Mais de metade dos alunos mostra-se interessado em utilizar esta ferramenta para o estudo, isto demonstra que apesar da imensa oferta de ferramentas de aprendizagem os alunos sentem necessidade de saber o que realmente devem de estudar para cada atividade na unidade curricular.

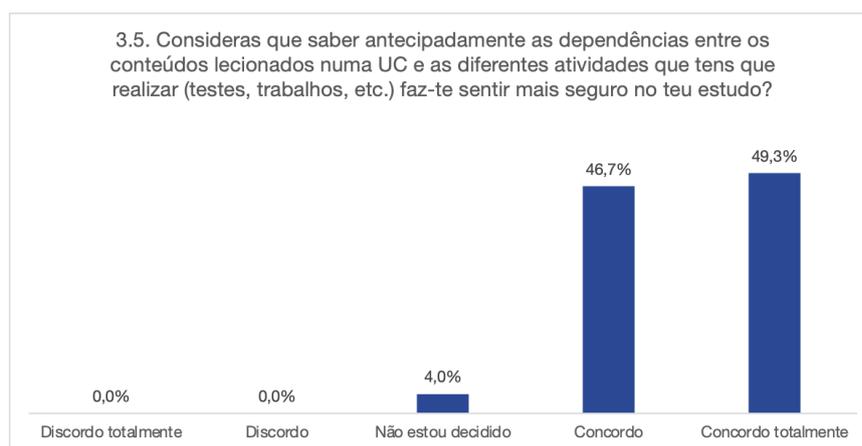


**Figura 41: Utilidade do LS 2.0 para o estudo dos alunos**

De seguida, foi apresentada aos alunos a interface representada na Figura 24, i.e., a visualização do mapeamento entre conteúdos da UC e uma *quest*, para que pudessem avaliar a interface apresentada e assinalarem os pontos positivos e negativos que encontraram na mesma.

Na sua maioria os alunos acharam a interface simples, intuitiva e de fácil entendimento, destacando a versatilidade, organização e eficiência no auxílio ao estudo. Para os alunos a interface apresenta uma visualização bem detalhada, abrangente e objetiva relativamente à matéria. Vários alunos realçam a possibilidade de ver toda a matéria que precisam de saber no mesmo sítio como um ponto positivo, outros valorizam o desenho da estrutura ao apresentar a divisão de tópicos por ramos, como também o contraste das cores que permite aos alunos terem a noção do que devem estudar e por isso consideram esta interface importante para o estudo autónomo dos mesmos. Quanto aos pontos negativos da interface, a grande maioria respondeu que não tinha nada a apontar. No entanto, alguns alunos consideram a interface um pouco complexa à primeira vista por ser extensa devido a representar todos os conteúdos pertencentes a uma UC (pelo que um aluno sugeriu criar uma opção para abrir e fechar ramos da interface), já outros referem-se ao facto do fundo escuro juntamente com as cores da interface dificultarem a visualização.

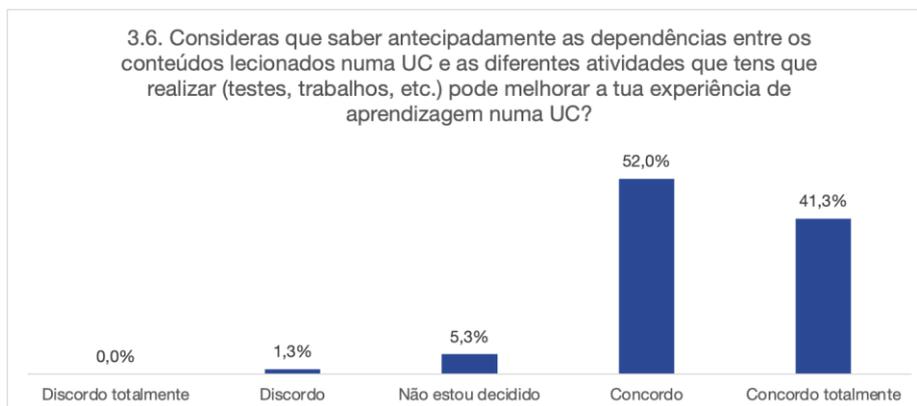
Com base nas respostas chegamos à conclusão de que a interface de visualização do mapeamento entre conteúdos e uma *quest* foi bem aceite pelos alunos, pois para além de a considerarem simples e bem detalhada, estes valorizam a possibilidade de saberem quais os conteúdos aos quais se devem dedicar mais para cada atividade na unidade curricular, pois não só auxilia, como promove o estudo autónomo por parte dos alunos.



**Figura 42: Opinião dos alunos relativamente à sua segurança no estudo**

Ao serem questionados se ao saberem antecipadamente as dependências entre os conteúdos lecionados e as diferentes atividades que têm de realizar, como por exemplo, testes ou exercícios, sentem-se mais seguros no seu estudo, os alunos foram bastante claros, 49,3% concordam totalmente, 46,7% concordam e apenas 4% não estão decididos (ver Figura 42).

De seguida, a mesma questão foi posta aos alunos, não em termos de segurança no estudo, mas agora relativamente a poder melhorar a experiência de aprendizagem na UC. Nesta questão 41.3% dos alunos concordam totalmente, 52% concordam, apenas 5.3% não estão decididos e 1.3% discordam (ver Figura 43).



**Figura 43: Opinião dos alunos relativamente à experiência de aprendizagem numa UC**

Com base nas respostas obtidas nestas duas questões podemos concluir que esta ferramenta tem potencial para realmente ajudar os alunos a estudar bem como melhorar a sua experiência de aprendizagem nas unidades curriculares. Através da funcionalidade do mapeamento entre conteúdos e *quests*, os alunos podem recorrer ao LS 2.0 para obter mais informação relativamente à UC e às suas atividades em vez de sobrecarregarem o docente com questões sobre as mesmas.

Os alunos foram depois questionados sobre a importância do LS 2.0 na concretização de quatro objetivos (ver Figura 44):

- melhorar a experiência global de aprendizagem na UC,
- melhorar a satisfação em geral com a UC,
- melhorar a satisfação com o seu desempenho individual na UC,
- melhorar o desempenho individual na UC, i.e., a nota na UC.

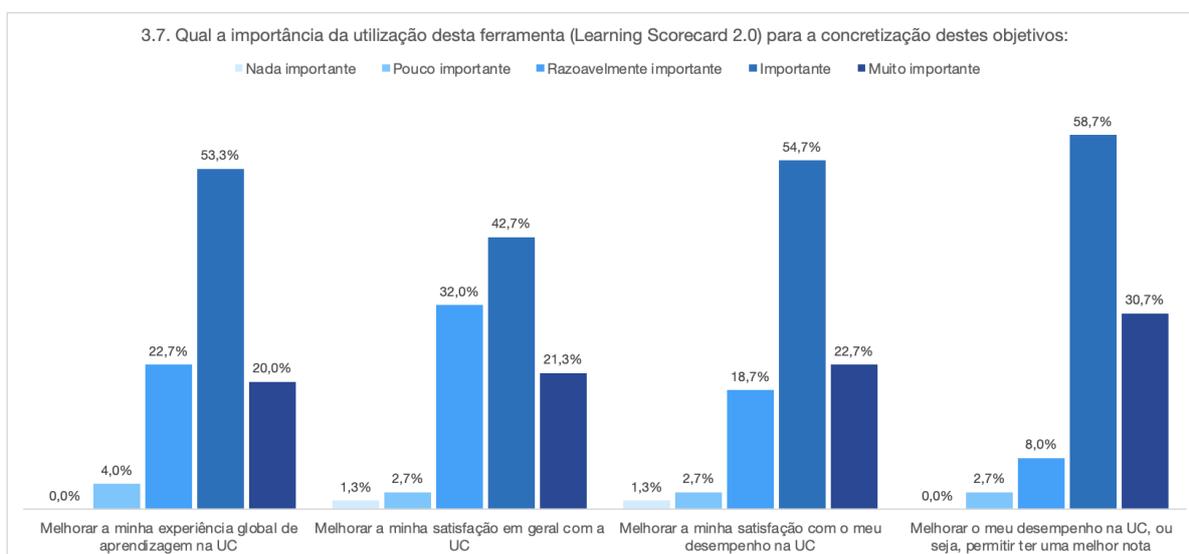
Para concretizar o primeiro objetivo, a maioria (53.3%) dos alunos considera a ferramenta importante, 20% dos alunos consideram-na muito importante, 22.7% consideram-na razoavelmente importante e apenas 4% consideram o LS 2.0 pouco importante para melhorar a experiência global de aprendizagem numa UC.

Relativamente ao segundo objetivo (melhorar a sua satisfação em geral com a UC), 42.7% dos alunos consideram a ferramenta importante, 21.3% muito importante, 32% razoavelmente importante e apenas 2.7% e 1.3% consideram-na pouco e nada importante, respetivamente.

Já para o terceiro objetivo (melhorar a satisfação com o seu próprio desempenho na UC), a maioria dos alunos considera o LS 2.0 importante, 22.7% muito importante, 18.7% razoavelmente importante, 2.7% e 1.3% consideram a ferramenta pouco e nada importante, respetivamente.

Para a concretização do quarto objetivo (obter uma melhor nota na UC), a opinião dos alunos é mais expressiva: 58.7% dos alunos consideram o LS 2.0 como importante, seguido por 30.7% a considerarem-no muito importante. Apenas 8% consideram o LS 2.0 razoavelmente importante, e 2.7% consideram esta ferramenta como pouco importante.

A partir da análise desta questão podemos concluir que, em geral, os alunos consideram a nova versão do LS como uma ferramenta importante para concretizar os objetivos apresentados, bem como ajudá-los a melhorar o seu desempenho individual e, conseqüentemente, melhorar a nota numa unidade curricular.



**Figura 44: Distribuição da importância do LS 2.0 para concretizar os quatro objetivos mencionados**

## 4.4 Avaliação da dificuldade de aprendizagem

Sobre se os alunos consideram que os docentes estão cientes ou a par das suas reais dificuldades, as respostas obtidas estão quase como que distribuídas entre concordo, discordo e não decidido. A maior parte dos alunos (30.7% concorda, 6.7% concorda totalmente) sente que os docentes estão a par das suas dificuldades, 32% dos alunos (25.3% discorda, 6.7% discorda totalmente) discorda desta afirmação e 30.7% afirma não estar decidido relativamente à mesma. Quando questionados sobre o porquê da resposta que forneceram, as opiniões dos alunos assemelham-se consoante a concordância, discordância ou indecisão.

Os alunos que concordam com esta afirmação atribuem aos anos de experiência do docente, seja a lidar com a matéria lecionada seja a interagir com os alunos, o que lhes permite ter uma noção mais concreta do que por norma é uma dificuldade na UC que o mesmo leciona. Estes alunos afirmam também que os docentes estão cientes das suas dificuldades através do acompanhamento ativo dos alunos e dedicam mais tempo a certas partes da matéria como também através de perguntas aos alunos e da análise do docente nos momentos de avaliação.

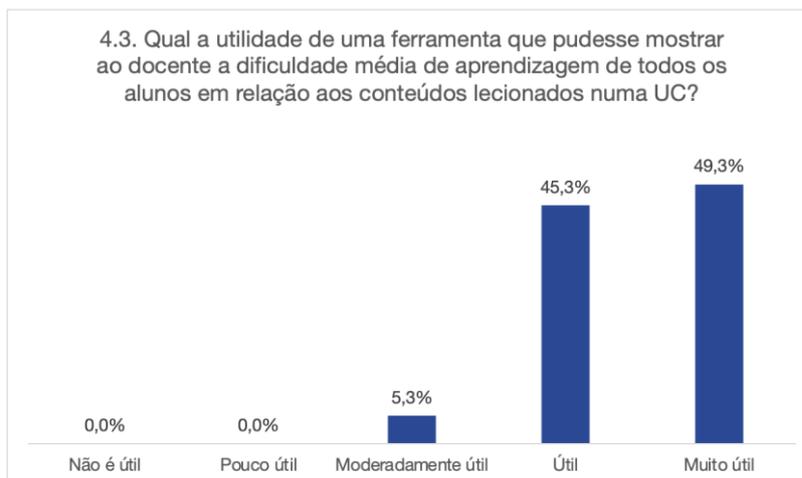
Para os alunos que discordam desta afirmação, as respostas incluem várias justificações como a existência de alunos menos participativos que não dão o *feedback* necessário aos docentes sobre as suas dúvidas em aula, alunos que tenham vergonha de por vezes demonstrar as mesmas ou até mesmo falta de interesse por parte dos alunos. Outros alunos mencionaram o facto de não haver a proximidade necessária entre o docente e os alunos ou de um docente ter vários alunos e resultar em sobrecarga do mesmo. Outras justificações referem que por vezes os docentes não adaptam a matéria consoante as dificuldades dos alunos ou que apenas veem as dificuldades mais acentuadas através dos resultados obtidos ao invés das mais subtis. Alguns alunos sentem que os docentes não têm a noção concreta das dificuldades sentidas pelos alunos ou que não mostram disponibilidade para ajudar ou adaptar os seus métodos de ensino.

Os alunos que estão indecisos quanto à questão apresentada na maioria afirmam depender do docente; existem docentes que realmente se interessam em ajudar e outros que não demonstram qualquer interesse, sentem que se deve ao facto de serem demasiados alunos para um docente. Referem também a falta de esforço ou vontade suficiente por parte dos docentes para entender as dificuldades, bem como a falta de clarificação e interação entre ambas as partes, pois acham que certos docentes não compreendem o quão realmente difíceis certos aspetos das UCs são e que estes “*deviam tentar colocar-se na posição dos alunos*”.

É importante referir que o LS 2.0 aborda alguns problemas acima referidos, tais como a falta de feedback fornecida pelo aluno. Através das funcionalidades de avaliação de dificuldade dos conteúdos da UC e de visualização do mapeamento da matéria com as *quests* é possível melhorar a interação docente-alunos, aproximando os alunos do docente e vice-versa, pois o docente ao ficar informado das dificuldades dos alunos pode ajustar o método de ensino.

Quando questionados sobre a utilidade de uma ferramenta que pudesse mostrar ao docente a dificuldade média de aprendizagem de todos os alunos em relação aos conteúdos lecionados numa UC os alunos foram bastante claros na resposta, 49.3% consideram-na muito útil, 45.3% útil e 5.3% moderadamente útil (ver Figura 45).

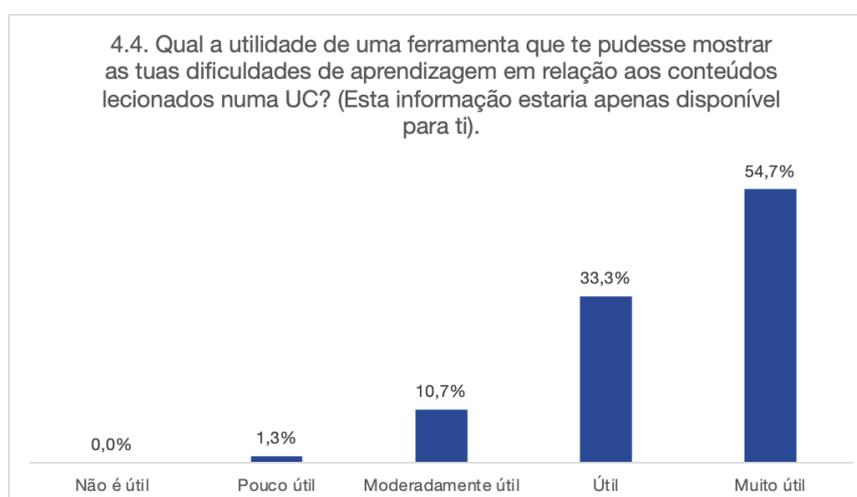
Podemos assim observar que esta nova versão do LS apresenta funcionalidades de interesse para os alunos, permitindo aos mesmos avaliar todos os conteúdos de nível 2 para que o docente possa ter uma ideia mais concreta sobre quais os tópicos que devem de ser aprofundados para que os alunos possam realmente ter melhor proveito nas unidades curriculares.



**Figura 45: Utilidade da visualização média de aprendizagem dos alunos numa UC**

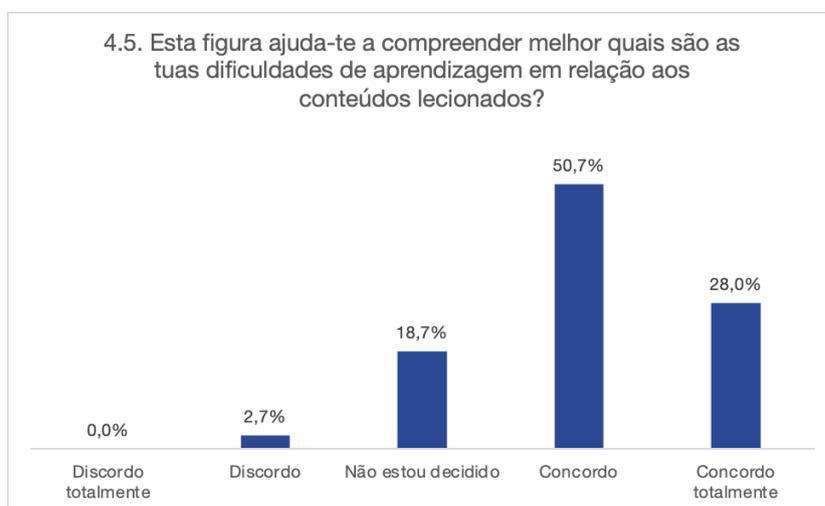
Quanto à utilidade de uma ferramenta que mostre as próprias dificuldades do aluno em relação aos conteúdos lecionados numa UC, os inquiridos na sua maioria (54.7%) avaliaram a ferramenta como muito útil, um terço (33.3%) dos inquiridos avaliou como útil e 10.7% consideraram-na moderadamente útil; apenas 1.3% acha a ferramenta pouco útil (ver Figura 46).

Dito isto, vemos que para além de interessados em fornecer informação ao docente sobre as suas dificuldades, os alunos valorizam uma ferramenta que os ajude a identificar quais as áreas em que sentem mais dificuldades, que por vezes não são claras para os próprios.



**Figura 46: Utilidade da visualização das dificuldades do aluno numa UC**

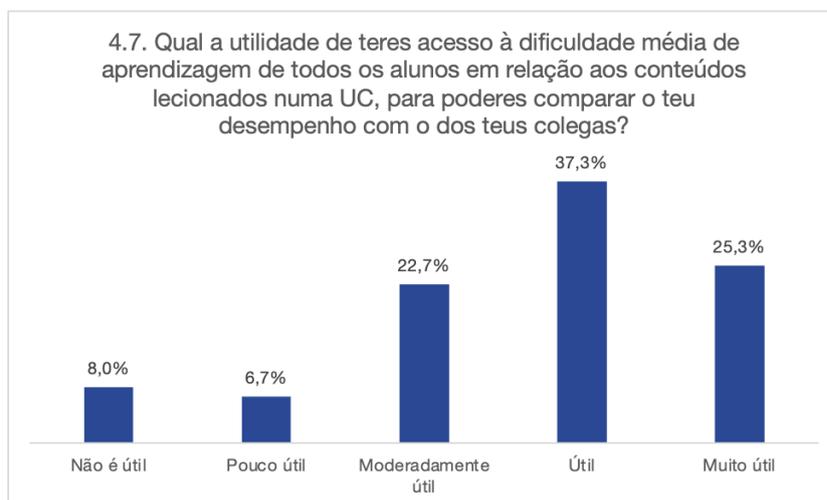
Seguidamente, foi apresentada aos alunos a interface representada na Figura 26 desta dissertação, que permite visualizar a dificuldade sentida pelos alunos nos SCs de uma UC. Quando questionados se a interface os ajuda a compreender melhor as suas dificuldades de aprendizagem em relação aos conteúdos lecionados, a maioria (50.7%) dos alunos concordam, 28% concordam totalmente, 18.7% não estão decididos e apenas 2.7% discordam (ver Figura 47).



**Figura 47: Opinião dos alunos relativa à interface de visualização de dificuldade**

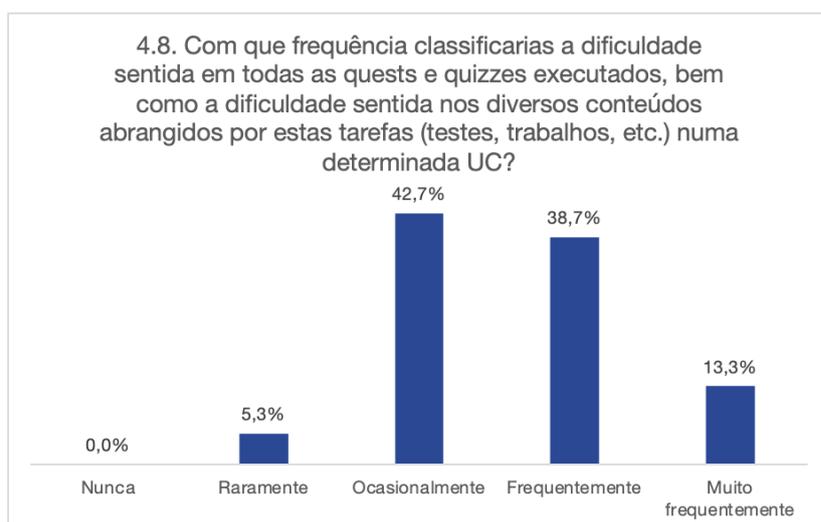
Foi depois pedido aos alunos que assinalassem os pontos positivos e negativos que encontraram na interface. Relativamente aos pontos positivos, a maioria dos alunos destaca a utilização das cores para representar os diferentes níveis de dificuldade, a legenda de cores, a simplicidade e clareza de apresentação, o entendimento claro das dificuldades sentidas nos conteúdos, e consideram a interface intuitiva, bem estruturada e detalhada. Os alunos valorizam bastante a facilidade de visualizar os próprios pontos fracos numa UC sem ter que estudar a matéria toda, realçam que esta interface permite ter uma noção mais realista de onde teriam que dedicar mais tempo e estudo, uma ferramenta útil para saberem quais as matérias que exigem mais deles de modo a organizar melhor o próprio estudo e esforço ao longo da UC. Já em relação aos pontos negativos a vasta maioria não teve nada a apontar, apenas alguns alunos referem o fundo preto e o tamanho das letras. Esta interface foi alvo de várias sugestões como a possibilidade de ordenar os conteúdos de maior para menor dificuldade ou alterar o número de níveis de 6 para 5 de modo a ser mais intuitiva a compreensão.

Em relação à utilidade de terem acesso à dificuldade média de aprendizagem de todos os alunos em relação aos conteúdos lecionados numa UC para que possam comparar o próprio desempenho com o dos colegas, os alunos na sua maioria consideram útil (37.3%), 25.3% consideram essa oportunidade como muito útil e 22.7% consideram-na moderadamente útil. Já 6.7% consideram pouco útil e 8% não veem utilidade nesta opção (ver Figura 48). Podemos observar que alguns alunos preferem realmente focar-se mais no seu desempenho e evitar comparações com os restantes colegas.



**Figura 48: Utilidade da visualização da dificuldade média de todos os alunos**

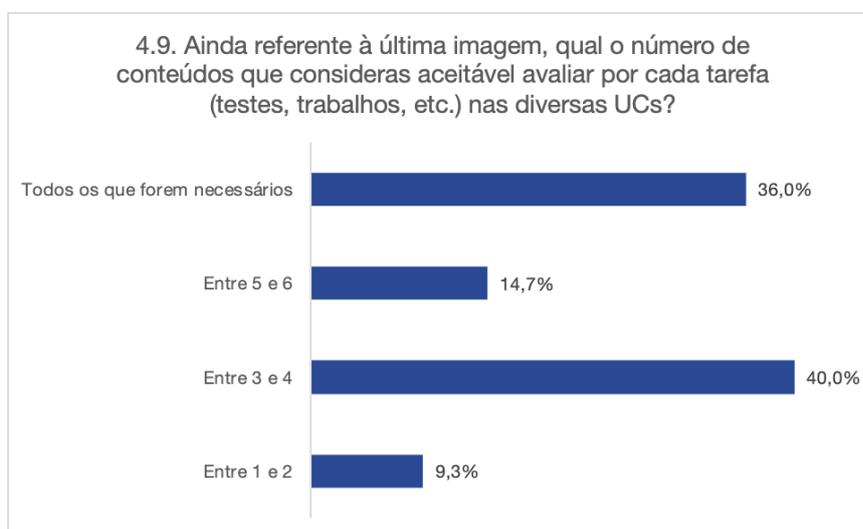
Na questão seguinte foi apresentada a interface de validação de *quests*, ilustrada na Figura 25 desta dissertação. Os alunos foram questionados sobre com que frequência classificariam a dificuldade sentida em todas as *quests*, bem como a dificuldade sentida nos diversos conteúdos abrangidos por estas tarefas (testes, trabalhos, etc.) numa determinada UC. Nesta funcionalidade, apesar da maioria (52%) prevalecer na utilização frequente (38.7% frequentemente e 13.3% muito frequentemente), a opção ocasionalmente foi a mais escolhida com 42.7% das respostas e apenas 5.3% raramente utilizariam (ver Figura 49). Estes resultados levam-nos a concluir que esta interface pode ser melhorada (em trabalho futuro) para se tornar mais apelativa aos alunos de modo a aumentar a frequência de utilização.



**Figura 49: Frequência da classificação de dificuldade das *quests***

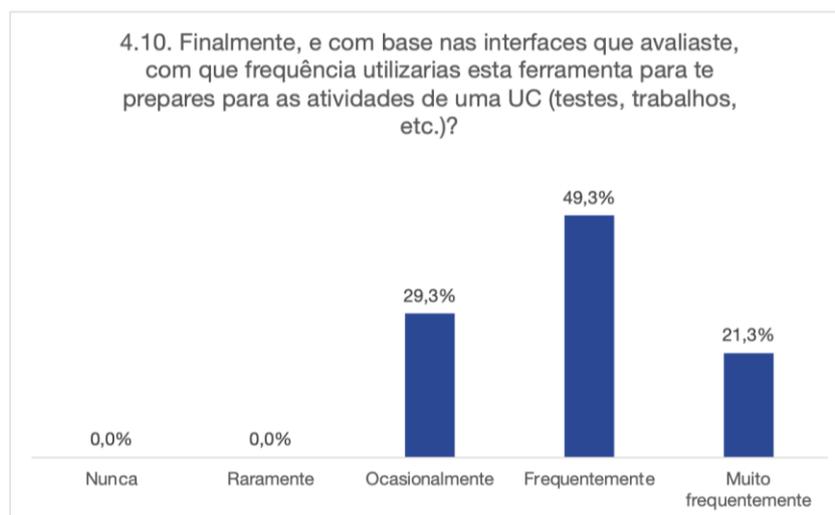
Os alunos foram ainda questionados relativamente ao número de conteúdos que considerariam aceitável avaliar por cada *quest*. A maioria (40%) respondeu entre três e quatro conteúdos, 14.7% respondeu entre cinco e seis conteúdos e 9.3% respondeu entre um e dois conteúdos (ver Figura 50).

A grande surpresa foi saber que 36% dos alunos aceitaria avaliar todos os conteúdos que fossem necessários, pois apesar das respostas à questão anterior suscitarem algumas questões para trabalho futuro podemos concluir que parte significativa dos alunos considera importante poder gerar essa informação, não só para eles, mas também para que o docente possa ter uma ideia mais concreta das reais dificuldades de aprendizagem dos alunos.



**Figura 50: Número aceitável de conteúdos para avaliar por cada *quest***

Por último, e com base nas interfaces que lhes foram apresentadas ao longo do questionário, foi perguntado aos alunos com que frequência utilizariam o LS 2.0 para se prepararem para as atividades de uma UC. A vasta maioria (70.6%) afirmou que utilizaria frequentemente e muito frequentemente (49.3% e 21.3% respetivamente) e 29.3% utilizaria ocasionalmente (ver Figura 51). É de notar que nenhum aluno respondeu que utilizaria raramente ou nunca, o que por si só é um ponto muito positivo para o trabalho desenvolvido nesta dissertação.



**Figura 51: Frequência de utilização da ferramenta LS 2.0**

Podemos atribuir diversas razões para explicar este resultado, numa primeira instância podemos destacar a grande aceitação por parte dos alunos às interfaces apresentadas do LS 2.0, bem como a resposta positiva à utilidade das diversas funcionalidades representadas neste questionário. Outra razão para a obtenção destes resultados deve-se ao LS responder *a priori* a uma grande parte das necessidades identificadas pelos alunos em relação aos docentes e às suas experiências nas unidades curriculares.

Em suma, a resposta à nova versão do LS foi muito positiva e permitiu ter uma noção mais concreta da opinião dos alunos em relação ao que realmente valorizam ao longo da experiência numa unidade curricular, as suas necessidades e prioridades como também hábitos e modos de estudo.



## Capítulo 5 Conclusão

O *Learning Scorecard* (LS) é uma plataforma de gestão de desempenho que pretende proporcionar aos alunos do Ensino Superior um ambiente analítico que possibilite o acompanhamento do seu desempenho numa unidade curricular (UC) contribuindo para a valorização da sua experiência de aprendizagem. O LS foca-se nos alunos e docentes ao fornecer vistas para ambos com informações relativas ao desempenho e aprendizagem dos alunos.

A nova versão da ferramenta *Learning Scorecard* tem o propósito de promover a melhoria contínua do processo de acompanhamento da aprendizagem dos alunos cujo principal objetivo consiste em incorporar uma ontologia com o potencial que o *Learning Scorecard* oferece em relação ao acompanhamento educacional, criando novas formas de estruturação e visualização dos elementos de uma unidade curricular como também permite inferir novo conhecimento relativo aos mesmos.

Nesta dissertação, a plataforma LS foi integralmente refeita, tanto a nível de *frontend* como de *backend* e armazenamento de dados. De notar que o *backend* constitui uma API REST, o que garante o isolamento e a independência tecnológica entre as diferentes camadas. O LS 2.0 aproveita os conceitos desenvolvidos nas versões anteriores e oferece novas funcionalidades (listadas na Tabela 2), suportadas por uma ontologia desenvolvida para o LS.

Numa primeira instância foram desenvolvidas as funcionalidades base do LS como a criação de utilizadores, UCs, *guilds* e *quests*. De seguida, foi criada a ontologia, definindo os conceitos pertencentes ao domínio do LS e respetivas relações que vão suportar as novas funcionalidades. A partir da etapa anterior tornou-se possível desenvolver funcionalidades relativas aos conteúdos (SCs), como o mapeamento entre conteúdos lecionados numa UC e *quests*, a avaliação da dificuldade sentida nesses mesmos conteúdos pelos alunos, as visualizações do mapeamento e das dificuldades (para o docente a dificuldade média sentida pelos participantes da UC, para o aluno a dificuldade sentida pelo próprio). Após o desenvolvimento da plataforma, procedeu-se à criação e posterior análise de um questionário relativo aos hábitos académicos dos alunos inquiridos e à aceitação das novas funcionalidades da plataforma por parte dos mesmos.

A aceitação em geral desta plataforma foi bastante positiva, tendo 70% dos alunos inquiridos (num total de 75) manifestado a intenção de utilização frequente<sup>12</sup> do LS. Estes valores são motivantes para continuar a aprimorar esta ferramenta e a desenvolver novas funcionalidades para a mesma.

## 5.1 Questões de Investigação

Esta dissertação propôs-se a responder a duas questões de investigação:

- **Será possível desenvolver uma ontologia para mapear e gerir os conteúdos programáticos relacionando-os com as atividades de uma unidade curricular?** A ontologia do LS 2.0 foi desenvolvida de forma a ser extensível e abrangente, permitindo organizar e mapear os conteúdos programáticos (SCs) de uma UC com as suas atividades (*quests*). Adicionalmente, fornece aos alunos e docentes uma visualização da dificuldade sentida pelos alunos em cada um dos SCs da UC, estando preparada para suportar outras visualizações. Deste modo, podemos obter e criar informação útil para ambos, tanto que através do questionário enviado aos alunos, a grande maioria expressou a satisfação com esta funcionalidade, sendo que 28% dos alunos considera-a muito útil, 40% útil e 24% moderadamente útil. É importante referir também que a utilização desta ontologia para o mapeamento entre SCs e *quests* não só promove a experiência de aprendizagem dos alunos numa UC, em que 93% dos alunos concordam (52% concordam e 41% concordam totalmente) com esta afirmação, como também 96% dos alunos concordam (47% concordam e 49% concordam totalmente) que esta funcionalidade os ajuda a sentirem-se mais seguros no seu estudo.
- **Poderá a ontologia ser integrada de forma útil no *Learning Scorecard*, sendo usada pelos alunos e docentes para a monitorização da aprendizagem numa UC?** Como referido acima, a ontologia suporta também a funcionalidade que fornece aos alunos e docentes uma visualização da dificuldade sentida pelos alunos. Esta visualização mapeia diferentes níveis de dificuldade com cores para que os alunos possam ter a perceção das suas dificuldades e para que os docentes possam ter um conhecimento mais concreto de quais os conteúdos de uma UC que suscitam mais dificuldades nos alunos. No questionário realizado, os alunos demonstraram-se interessados com esta funcionalidade, quando questionados sobre qual a utilidade da mesma para que o docente possa ter conhecimento das suas dificuldades 94% considerou-a útil (45% útil

---

<sup>12</sup> 49% dos alunos responderam que utilizariam o LS frequentemente e 21% muito frequentemente.

e 49% muito útil). Os alunos também foram inquiridos relativamente à utilidade de uma ferramenta que pudesse mostrar as suas próprias dificuldades em que 99% dos inquiridos consideram-na útil (55% muito útil, 33% útil e 11% moderadamente útil). A integração da ontologia no LS 2.0 promoveu a aceitação da plataforma pelos alunos em relação à monitorização da sua aprendizagem, com cerca de 70% a afirmarem que a utilizariam frequentemente (49% frequentemente e 21% muito frequentemente).

## 5.2 Limitações

Esta dissertação encontrou algumas limitações no curso do seu desenvolvimento. A limitação mais sentida foi no desenvolvimento da interface de visualização, o componente *mindmap*. Foi realizada uma pesquisa abrangente sobre componentes de utilização gratuita que pudessem ser utilizados para criar esta interface. Esta pesquisa revelou-se bastante difícil devido à falta de componentes que satisfaçam os requisitos pretendidos, ora porque uns componentes encontravam-se demasiado “*hardcoded*” o que por sua vez dificultava a integração dos mesmos com a plataforma ora devido à falta de potencial a nível funcional para o que era pretendido desenvolver. Foi decidido utilizar a ferramenta *goJS* para este efeito, visto oferecer a possibilidade aos programadores de editarem o código fonte para ajustarem a mesma às funcionalidades que são pretendidas. Esta ferramenta permitiu criar um *mindmap* com as funcionalidades suportadas pela ontologia, de forma a criar um componente semântico único para ser utilizado no LS. Apesar de ser uma ferramenta paga, foi-nos concedida autorização para utilizar o *goJS* no âmbito desta dissertação.

Outra limitação encontrada foi na integração do *GraphDB* (base de dados em grafo) onde está armazenada a ontologia do LS. Para manter uma instância deste *software* a funcionar foi necessária a criação de uma máquina virtual na plataforma *Azure* da Microsoft. Contudo, este *software* que corre sobre JVM (*Java Virtual Machine*) necessita de mais de 2GB de RAM para ser executada sem limitações. As máquinas virtuais de uso gratuito no *Azure* apenas contêm 1GB de RAM, o que impossibilitava a utilização desta ferramenta pelo LS pois a instância do *GraphDB* parava por falta de memória com pouca utilização por parte do LS. A solução encontrada adveio de uma oferta da Microsoft para estudantes, que oferecia um crédito de 100 dólares para utilizar na plataforma *Azure*, com isto foi possível requisitar uma máquina virtual com recursos suficientes para providenciar um ambiente de teste capaz de suportar o desenvolvimento desta nova versão do LS durante a realização desta dissertação.

### 5.3 Trabalho Futuro

Esta versão do LS englobou muitas das funcionalidades presentes nas versões anteriores da plataforma. Como trabalho futuro são propostas algumas novas funcionalidades bem como melhorias relativamente às funcionalidades já desenvolvidas:

- **Adição dos *Learning Outcomes* no LS** – os objetivos de aprendizagem (*Learning Outcomes*) estão relacionados com os SCs e com as *quests* de uma UC. Apesar de já estarem definidos na ontologia bem como as suas relações, ainda não foram desenvolvidas funcionalidades que englobem os LOs, que têm potencial para podermos mapear *quests* e SCs com características como *skills* e conhecimentos obtidos, o que permitirá fornecer aos docentes informação útil sobre o desempenho dos alunos.
- **Implementação de dashboards no LS 2.0** – devido ao desenvolvimento já extenso desta versão do LS, algumas funcionalidades já existentes nas versões anteriores, como os *dashboards*, ficaram fora do âmbito deste trabalho. De notar que os *dashboards* serão alvo de estudo noutra dissertação de mestrado com início em setembro de 2021.
- **Melhorar algumas interfaces do LS 2.0** – com base no questionário percebeu-se que alguns alunos acham que as decisões tomadas relativamente ao desenho das interfaces do LS 2.0 não são apelativas. Por exemplo, a utilização do fundo preto, comum em ambientes mais tecnológicos (e.g., para desenvolvedores de software), não é tão apelativo para perfis de utilizadores menos técnicos. Uma solução pode ser proporcionar aos utilizadores a possibilidade de escolha entre dois *layouts*: um fundo branco ou preto.
- **Melhorar a infraestrutura do LS 2.0** – como esta versão do LS foi uma prova de conceito, a infraestrutura apenas está preparada para utilização leve, neste caso apenas para desenvolver e testar o LS. Para que esta ferramenta possa ser utilizada pelos alunos e docentes deve de haver uma melhoria na infraestrutura existente, nomeadamente aumentar os recursos das máquinas virtuais que hospedam esta plataforma.

## Referências Bibliográficas

- Al-yahya, M., George, R., & Alfaries, A. (2015). Ontologies in E-Learning : Review of the Literature Publication Growth for the word " Ontology " Publication Growth for the words " Ontology " and " E-Learning ". *International Journal of Software Engineering and Its Applications*, 9(2), 67–84.
- Aroyo, L., Dolog, P., Houben, G. J., Kravcik, M., Naeve, A., Nilsson, M., & Wild, F. (2006). Interoperability in personalized adaptive learning. *Educational Technology and Society*, 9(2), 4–18.
- Bittencourt, I. I. M., Costa, E., Isotani, S., & Mizoguchi, R. (2008). Towards a Reference Model to Semantic Web-based Educational Systems. *Cscmuedu*, 1–10. <http://www.cs.cmu.edu/~sisotani/artigos/Towards a Reference Model to Semantic Web-based.pdf>
- Bogoslov, I. A. (2018). *Future Research Directions on Web-Based Educational Systems*. 9–20.
- Bravo, M., Reyes, L. F. H., & Reyes Ortiz, J. A. (2019). Methodology for ontology design and construction. *Contaduria y Administracion*, 64(4), 1–24. <https://doi.org/10.22201/FCA.24488410E.2020.2368>
- Cardoso, E. (2018). *Learning Analytics and Gamification. EUNIS Learning Analytics Workshop: Using data and analytics to enhance learning and teaching practice. Helsinki, Finland 11-12 October 2018*.
- Cardoso, E., Costa, D., & Santos, D. (2017). Introducing the Learning Scorecard: a tool to improve the student learning experience. *European Journal of Higher Education IT*, May.
- Cardoso, E., Santos, D., Costa, D., Caçador, F., Antunes, A., & Ramos, R. (2016). Learning Scorecard: Monitor and Foster Student Learning through Gamification . In *2nd International Workshop on Educational Knowledge Management* (Vol. 1780, pp. 39–50). <http://ceur-ws.org/Vol-1780/>
- Cardoso, E., Santos, D., Costa, D., Caçador, F., Antunes, A., & Ramos, R. (2017). Learning scorecard: monitor and foster student learning through gamification. In *Knowledge Engineering and Knowledge Management. EKAW 2016. Lecture Notes in Computer Science* (Vol. 10180, pp. 55–68). [https://doi.org/10.1007/978-3-319-58694-6\\_5](https://doi.org/10.1007/978-3-319-58694-6_5)
- Cimiano, P. (2006). Ontologies. *Ontology Learning and Population from Text*, 9–17. [https://doi.org/10.1007/978-0-387-39252-3\\_2](https://doi.org/10.1007/978-0-387-39252-3_2)
- Costa, D. (2017). *Learning Scorecard: Plataforma Para a Monitorização Da Experiência De Aprendizagem De Alunos No Ensino Superior Aplicando Técnicas De Business Intelligence E Gamificação* [ISCTE-IUL]. <https://repositorio.iscte-iul.pt/handle/10071/15604>
- Croitoru, M., & Dinu, C.-N. (2016). A critical analysis of learning management systems in higher education. *Economy Informatics*, 16(1), 5–18. <https://search-proquest-com.libproxy.csun.edu/docview/1988422260/fulltextPDF/62F85A3E5FFB4CA7PQ/1?accountid=7285>
- Dicheva, D. (2008). Ontologies and Semantic Web for E-Learning. *Handbook on Information Technologies for Education and Training*, 47–65. [https://doi.org/10.1007/978-3-540-74155-8\\_3](https://doi.org/10.1007/978-3-540-74155-8_3)
- Dicheva, D., Sosnovsky, S., Gavrilova, T., & Brusilovsky, P. (2005). *Ontological Web Portal for Educational Ontologies*. 19–27. <http://www.win.tue.nl/SW-EL/2005/swel05-aied05/proceedings/4-Dicheva-final-full.pdf>
- Draft, W. (2003). *RDF / XML Syntax Specification ( Revised )*. October, 1–56.
- Falquet, G., Métral, C., Teller, J., & Tweed, C. (2011). *Ontologies in Urban Development Projects* (Vol. 1, Issue

- March 2016). <https://doi.org/10.1007/978-0-85729-724-2>
- Gandon, F., Krummenacher, R., Han, S.-K., & Toma, I. (2011). *The Resource Description Framework and its Schema. Handbook of Semantic Web Technologies* (Issue January).  
<http://www.springer.com/us/book/9783540929123>
- Ghirardini, B. (2011). E-learning methodologies: A guide for designing and developing e-learning courses. In *Food and Agriculture Organization of the United Nations (FAO)*. <https://doi.org/10.1111>
- Gómez-Pérez, A., Fernández-López, M., Corcho, O., & Gomez-Perez, A. (2010). Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. In *with examples from the areas of knowledge management, e-commerce and the Semantic Web* (Issue March 2016).
- Grimm, S. (2010). *Knowledge Representation and Ontologies BT - Scientific Data Mining and Knowledge Discovery: Principles and Foundations* (M. M. Gaber (ed.); pp. 111–137). Springer Berlin Heidelberg.  
[https://doi.org/10.1007/978-3-642-02788-8\\_6](https://doi.org/10.1007/978-3-642-02788-8_6)
- Guarino, N. (1997). Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1299, 139–170. [https://doi.org/10.1007/3-540-63438-x\\_8](https://doi.org/10.1007/3-540-63438-x_8)
- Heflin, J. (2007). An introduction to the owl web ontology language. *Lehigh University. National Science Foundation (NSF)*, 7.
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science research in information systems. *MIS Quarterly*, 28(1), 75–105.
- IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos, CA: *IEEE Computer Society*. <https://doi.org/10.1109/IEEESTD.1990.101064>
- Lassila, O., & Swick, R. R. (1999). *Resource description framework (RDF) model and syntax specification. World Wide Web Consortium Recommendation. October*. <http://www.w3.org/TR/REC-rdf-syntax>
- Pedroso, T. (2018). *Gamificação para a melhoria da experiência de aprendizagem dos alunos: caso estudo Learning Scorecard* [ISCTE-IUL]. <http://hdl.handle.net/10071/18415>
- Pedroso, T., Cardoso, E., Rações, F., Baptista, A., & Barateiro, J. (2019). *Learning Scorecard Gamification: Application of the MDA Framework BT - Information Systems for Industry 4.0* (I. Ramos, R. Quaresma, P. Silva, & T. Oliveira (eds.); pp. 29–46). Springer International Publishing.
- Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24, 45–77.
- Prot, U., Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C., Jupp, S., Moulton, G., Drummond, N., & Brandt, S. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. *Matrix*, 0–107.  
[http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4\\_v1\\_3.pdf](http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf)
- Rações, F. (2018). *Learning Scorecard: Visualização Da Experiência De Aprendizagem Em Estudantes Do Ensino Superior* [ISCTE-IUL]. <https://repositorio.iscte-iul.pt/handle/10071/17294>
- Randhawa, S. (2008). *Open source software and libraries. January 2008*.  
<http://eprints.rclis.org/handle/10760/13172>

- Simões, T. M. C., Rodrigues, J. J. P. C., & de la Torre, I. (2013). Personal learning environment box (PLEBOX): A new approach to E-learning platforms. *Computer Applications in Engineering Education*, 21, E100–E109. <https://doi.org/10.1002/cae.20537>
- Smith, B. (2002). Ontology and Information Systems. *Science*, 1964, 1–97. [http://ontology.buffalo.edu/ontology\\_long.pdf](http://ontology.buffalo.edu/ontology_long.pdf)
- Snae, C., & Brückner, M. (2007). Ontology-Driven E-Learning System Based on Roles and Activities for Thai Learning Environment. *Interdisciplinary Journal of E-Skills and Lifelong Learning*, 3, 001–017. <https://doi.org/10.28945/382>
- Stancin, K., Poscic, P., & Jaksic, D. (2020). Ontologies in education – state of the art. *Education and Information Technologies*, 25(6), 5301–5320. <https://doi.org/10.1007/s10639-020-10226-z>
- Stojanovic, L., Staab, S., & Rudi, S. (2001). eLearning based on the Semantic Web. *WebNet2001 - World Conference on the WWW and Internet, Orlando, Florida, Neidl*, 23–27. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.295&rep=rep1&type=pdf>
- Tapia-Leon, M., Rivera, A. C., Chicaiza, J., & Lujan-Mora, S. (2018). Application of ontologies in higher education: A systematic mapping study. *IEEE Global Engineering Education Conference, EDUCON, 2018-April(May 2019)*, 1344–1353. <https://doi.org/10.1109/EDUCON.2018.8363385>
- Valaski, J., Malucelli, A., & Reinehr, S. (2012). Ontologies application in organizational learning: A literature review. *Expert Systems with Applications*, 39(8), 7555–7561. <https://doi.org/10.1016/j.eswa.2012.01.075>
- Whitelock, D., & Anderson, T. (2004). The educational semantic web: Visioning and practicing the future of education. *Journal of Interactive Media in Education*, 2004, 1–15. <http://www-jime.open.ac.uk/2004/1/editorial-2004-1-disc-paper.html>
- Wilson, R. (2004). The role of ontologies in teaching and learning. *TechWatch Reports, August*. [http://www.jisc.org.uk/uploaded\\_documents/ACF11AC.pdf](http://www.jisc.org.uk/uploaded_documents/ACF11AC.pdf)
- Zem-Lopes, A. M., & Isotani, S. (2018). Abordagem de Avaliação da Qualidade de Sistemas Educacionais baseados em Web Semântica. *Anais Dos Workshops Do VII Congresso Brasileiro de Informática Na Educação (CBIE 2018)*, 1(October), 12. <https://doi.org/10.5753/cbie.wcbie.2018.12>
- Zhang, N., & Bao, H. (2010). Research on E-learning with digital technology in distance education. *IC4E 2010 - 2010 International Conference on e-Education, e-Business, e-Management and e-Learning*, 299–302. <https://doi.org/10.1109/IC4E.2010.126>