



Sistema de apoio à decisão para gestão das tarefas de avaliação das competências dos alunos

ANTÓNIO JOSÉ MEIRELES DE LIMA

Julho de 2019

Sistema de apoio à decisão para gestão das tarefas de avaliação das competências dos alunos

António José Meireles de Lima

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: Professor Doutor António Constantino Lopes Martins

Coorientador: Professora Doutora Goreti Marreiros

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, julho de 2019

Aos meus pais e à minha irmã, por me terem apoiado até aqui.

À minha namorada, Mélodi Guedes, pelo apoio incansável.

Resumo

Nas duas últimas décadas, o combate ao Abandono Escolar precoce tem sido um dos principais objetivos da União Europeia. No ano de 2009, Portugal e Espanha figuravam na lista de Estados Membros com as taxas de abandono escolar mais elevadas da União Europeia (cerca de 30.9%). Para combater estes problemas, a Comissão Europeia define em 2010, no plano estratégico “Europa 2020”, que, até o ano de 2020, a taxa de abandono escolar deve ser inferior a 10% e pelo menos 40% da geração jovem deve possuir formação superior.

Para ir de encontro aos objetivos estipulados pela Comissão Europeia no plano estratégico “Europa 2020”, surge o projeto piloto “**From birth to adult age – a WBL successful Practice!**”. Este projeto é cofinanciado pelo programa “Erasmus+”, e o seu principal objetivo passa por reformular a forma como as instituições de ensino profissional atuam. Um dos principais resultados desse projeto europeu é uma plataforma informática que auxilie as escolas e as empresas no processo de validação e avaliação das competências dos alunos.

Esta dissertação enquadra-se no desenvolvimento da plataforma informática supramencionada. Para atingir este objetivo os requisitos funcionais e não funcionais foram identificados, assim como várias arquiteturas que permitam responder aos requisitos levantados. Foram estudadas as melhores técnicas de integração para permitir a conexão entre sistemas das várias escolas.

Os requisitos da aplicação foram identificados através da realização de reuniões com as partes interessadas. Na análise das arquiteturas, verificou-se que a alternativa que utiliza um ESB como mecanismo de integração seria a mais útil. Os testes de sistema validaram a conformidade da aplicação com alguns dos requisitos não funcionais. Com os testes de usabilidade concluiu-se que a o sistema desenvolvido corresponde às expectativas dos utilizadores.

Palavras-chave: Educação, LMS, ESB, Integração de Sistemas, Arquitetura de Software, Clean Architecture

Abstract

In the last two decades, combating early school leaving has been one of the main objectives of the European Union. In 2009, Portugal and Spain were on the list of Member States with the highest dropout rates in the European Union (around 30,9%). In order to combat these problems, the European Commission defines in 2010, in the “Europe 2020” strategic plan that, by the year of 2020, the drop-out rate should be less than 10% and at least 40% of the youth generation should have higher education.

In order to meet the goals set by the European Commission in the “Europe 2020” strategic plan, the pilot project “**From birth to adult age – a WBL successful Practice!**” emerge. This project is co-funded by the “Erasmus+” program, and its main goal is to restructure the way in which VET institutions act. One of the main goals of this project is a software that helps schools and companies validate and evaluate students’ competences.

The main goal of this thesis is the development of the aforementioned software solution. To achieve this goal, functional and non-function requirement have been identified, as well as several architectures that allow to meet the specified requirements. Some integration techniques were studied to allow the integration between systems of the various schools.

Application requirements have been identified by bringing together the different stakeholders. In the software design, it was concluded that the alternative that uses an ESB as an integration mechanism would be the most useful. System tests validated the application’s compliance with some of the non-functional requirements. With the usability tests it was concluded that the developed system matches the user’s expectations.

Keywords: Education, LMS, ESB, Systems Integration, Software Architecture, Clean Architecture

Agradecimentos

Em primeiro lugar agradeço aos meus orientadores e coorientadores, Professor Constantino Martins e Professora Goreti Marreiros, pela ajuda e, acima de tudo, por nunca me terem deixado cair ao longo destes meses. Um especial agradecimento também ao Luís Conceição, pela experiência transmitida.

Aos meus pais, irmã e à minha namorada, por sempre terem acreditado que este dia iria chegar.

Aos meus amigos, em especial ao Bernardo Bessa, pelas trocas de ideias que certamente permitiram chegar a este resultado.

E por último, à minha empresa, Armis, e a toda a equipa da Armis ITS, por me terem dado o tempo e espaço necessário para concretizar este objetivo.

Obrigado.

António Lima

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema	2
1.3	Objetivos	3
1.4	Resultados esperados	4
1.5	Estrutura do documento	5
2	Contexto e Estado da Arte	7
2.1	Contexto	7
2.1.1	Educação e Formação 2020 (EF 2020)	7
2.1.2	Erasmus+	8
2.1.3	A importância da cooperação entre as instituições de ensino e as empresas	9
2.1.4	From birth to adult age - a WBL successful Practice	10
2.2	Áreas abrangidas	12
2.2.1	Learning Management Systems (LMS)	12
2.3	Análise de Valor	16
2.3.1	<i>The New Concept Development Model (NCD)</i>	16
2.3.2	Proposta de Valor	17
2.3.3	Modelo Canvas	17
3	Análise e Design	19
3.1	Visão geral	20
3.2	Engenharia de requisitos	20
3.2.1	Atores do sistema	20
3.2.2	Modelo de domínio	21
3.2.3	Requisitos Funcionais	24
3.2.4	Requisitos não funcionais	28
3.3	Tecnologias relevantes	29
3.3.1	Aplicação <i>web</i>	29
3.3.2	Base de dados	30
3.4	Design Arquitetural	30
3.4.1	Clean Architecture	30
3.4.2	A aplicação <i>web</i>	32
3.4.3	Integração de dados entre aplicações	34
3.4.4	Geração de relatório e certificados	38
3.4.5	Vista de dados	40
3.4.6	Vista de Implantação	40
3.4.7	Continuous Integration (CI)	41
4	Implementação	43
4.1	Implementação da aplicação <i>web</i>	43

4.1.1	Camada “SharedKernel”	44
4.1.2	Camada Core	44
4.1.3	Camada de Infraestrutura	46
4.1.4	Camada de apresentação	47
4.1.5	Evidências de boas práticas e do uso de padrões de <i>software</i>	48
4.2	Integração de sistemas	53
4.2.1	Capacidades de um ESB	53
4.2.2	Seleção do ESB	53
4.2.3	Solução a implementar	53
4.2.4	Implementação do pipeline do ESB	54
5	Avaliação	61
5.1	Uma breve introdução aos testes	61
5.2	Implementação dos testes	62
5.2.1	Testes unitários	62
5.2.2	Testes de integração	64
5.2.3	Testes de sistema	64
5.3	Avaliação de usabilidade	66
5.3.1	Metodologia para a sessão de avaliação	66
5.3.2	Resultados e interpretação de resultados	68
5.3.3	Apreciações finais	72
6	Conclusões	73
6.1	Resumo	73
6.2	Objetivos realizados	74
6.3	Limitações e trabalho futuro	75

Lista de Figuras

Figura 1 – Excerto do curriculum do curso de Manutenção Industrial.....	10
Figura 2 – <i>Inputs e outputs</i> da plataforma digital a desenvolver.....	12
Figura 3 – <i>Overview</i> das funcionalidades típicas de um LMS [17]	13
Figura 4 – Logótipo do Moodle [18].....	14
Figura 5 – Logótipo da plataforma OLAT [21]	14
Figura 6 – Logótipo do Blackboard [23]	15
Figura 7 – Modelo New Concept Development (NCD) [25]	16
Figura 8 – Modelo Canvas	18
Figura 9 – Modelo de domínio	23
Figura 10 – Vista funcional do administrador de instância.....	24
Figura 11 – Vista funcional do Administrador de Instância e Diretor de Curso.....	26
Figura 12 - Vista funcional do Diretor de Curso, Professor e Orientador de Empresa	27
Figura 13 – Vista (conceptual) de camadas do estilo arquitetural “Clean Architecture” [34]..	32
Figura 14 – Inserção de testes unitários e de integração na arquitetura [34].....	32
Figura 15 – Vista de alto nível	33
Figura 16 – Componentes da aplicação na arquitetura “Clean Architecture”	34
Figura 17 – Integração entre instâncias (alternativa 1)	34
Figura 18 – Diagrama de sequência da alternativa 1 de integração.....	35
Figura 19 – Integração entre instâncias (alternativa 2)	35
Figura 20 - Diagrama de sequência da alternativa 2 de integração.....	36
Figura 21 – Integração entre instâncias (alternativa 3)	37
Figura 22 - Diagrama de sequência da alternativa 3 de integração.....	37
Figura 23 – Introdução do ESB na arquitetura.....	38
Figura 24 – Possível ponto crítico na geração de certificados	38
Figura 25 – Vista lógica do componente “Educa.Infrastructure”	40
Figura 26 – Vista de implantação	41
Figura 27 – Pipeline de Continuous Integration (CI).....	42
Figura 28 – Organização dos projetos.....	43
Figura 29 – Vista de módulos da camada “SharedKernel”	44
Figura 30 – Vista de módulos da camada Core	45
Figura 31 – Vista de módulos “Educa.Core”	45
Figura 32 – Vista de módulos da camada de Infraestrutura	46
Figura 33 – Vista de módulos da camada de apresentação.....	47
Figura 34 – Vista de módulos “Educa.Web”	47
Figura 35 – Implementação do padrão repositório + unidade de trabalho.....	49
Figura 36 – Implementação do padrão “estratégia”.....	50
Figura 37 – Diagrama de classes do padrão Notificação	52
Figura 38 – Diagrama conceptual da solução a implementar.....	54
Figura 39 – REST API criada no ESB.....	54
Figura 40 – Representação do pipeline implementado no ESB	56

Figura 41 – Análise da Insequence do ESB (passo 1).....	57
Figura 42 – Análise da Insequence do ESB (passo 2).....	57
Figura 43 – Análise da Outsequence do ESB (passo 1).....	58
Figura 44 - Análise da Outsequence do ESB (passo 2)	59
Figura 45 – Mapeamento e cálculo da nota.....	59
Figura 46 – Estrutura de testes do projeto “Educa”	62
Figura 47 – Teste de carga na geração de relatórios.....	65
Figura 48 – Teste de carga na geração de certificados	65
Figura 49 – Cargos dos participantes	68
Figura 50 – Classificação da experiência de utilização	69
Figura 51 – Classificação das funcionalidades de administração.....	69
Figura 52 – Classificação das funcionalidades de gestão de disciplinas.....	70
Figura 53 – Classificação das funcionalidades de gestão de curso	70
Figura 54 – Classificação do processo de avaliação de alunos.....	71
Figura 55 – Avaliação final da plataforma.....	71

Lista de Tabelas

Tabela 1 – Taxas de AEP na União Europeia [3].....	1
Tabela 2 – Entidades envolvidas no projeto	11
Tabela 3 – Comparação dos três produtos apresentados	15
Tabela 4 – Atores do sistema	21
Tabela 5 – Escalas de avaliação.....	22
Tabela 6 – Especificação do caso de uso “UC1. Criar Utilizador”	24
Tabela 7 – Especificação do caso de uso “UC4. Criar Curso”	25
Tabela 8 - Especificação do caso de uso “UC18. Criar Outcome de disciplina”	26
Tabela 9 - Especificação do caso de uso “UC20. Gerar Certificado de aquisição de competências”	27
Tabela 10 - Especificação do caso de uso “UC22. Avaliar aluno”	27
Tabela 11 – Requisitos não funcionais identificados	28
Tabela 12 – Comparação de tecnologias Microsoft.....	30
Tabela 13 – Comparação entre Entity Framework Core e Dapper [36].....	39
Tabela 14 – Especificação do JSON enviado para a REST API do ESB	55
Tabela 15 – Resumo dos testes de carga efetuados.....	66
Tabela 16 – Escala usada no questionário de usabilidade.....	68
Tabela 17 – Sugestões e comentários dos participantes	71
Tabela 18 – Objetivos propostos e percentagem de conclusão	74

Acrónimos e Símbolos

Lista de Acrónimos

AAA	Arrange, Act, Assert
AEP	Abandono Escolar Precoce
API	Application Programming Interface
CMS	<i>Content Management System</i>
ECTS	European Credit Transfer System
ECVETS	European Credit System for Vocational Education and Training
EF2010	Plano Educação e Formação da União Europeia do ano 2010
EF2020	Plano Educação e Formação da União Europeia do ano 2020
EFP	Ensino e Formação Profissional
ESB	Enterprise Service Bus
FURPS	Functionality, Usability, Reliability, Performance, Supportability
GECAD	Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento
IPP	Instituto Politécnico do Porto
LMS	<i>Learning Management System</i>
MVC	<i>Model-View-Controller</i>
ORM	Object-Relational Mapper
SEI	Software Engineering Institute
SQL	Structured Query Language
WBL	<i>Work-based learning</i>
UE	União Europeia
UI	<i>User Interface</i>

1 Introdução

Este capítulo introduz o contexto da dissertação, o problema que esta pretende resolver, os objetivos que se pretendem atingir e os resultados esperados. Complementarmente, é introduzida a análise de valor. Por fim, é detalhada a estrutura do documento.

1.1 Contexto

Nas duas últimas décadas, o combate ao Abandono Escolar Precoce (AEP, daqui em diante) tem sido um dos principais objetivos da União Europeia. A Comissão Europeia define o AEP como “uma falha na conclusão do ensino médio, a incapacidade de concluir a escolaridade obrigatória ou a falta de qualificações ou certificados” [1]. De acordo com a estratégia “Europa 2020¹”, até o ano 2020, a percentagem de AEP deve ser inferior a 10% e pelo menos 40% da geração jovem deve possuir formação superior [2]. A Tabela 1 apresenta os cinco estados membros com as maiores taxas no ano de 2009.

Tabela 1 – Taxas de AEP na União Europeia [3]

País	Taxa de AEP (%)
Espanha	30.9
Portugal	30.9
Malta	25.7
Islândia	21.3
Itália	19.1

Não existe um motivo específico que explique a razão pelo qual um indivíduo decide abandonar o ensino numa fase prematura. Na verdade, trata-se de uma decisão meramente pessoal. No entanto, ao nível social, o AEP segue alguns padrões, tornando possível a definição de um perfil geral destes indivíduos.

¹ Plano estratégico de crescimento (a cumprir até o ano 2020) implementado pela União Europeia

Assim, de uma forma geral, um individuo inserido neste perfil é [1]:

- proveniente de uma família de classe baixa, onde, geralmente, o nível de ensino dos pais é baixo;
- pertencente a grupos vulneráveis, tais como mães adolescentes ou pessoas com limitações físicas ou/e cognitivas;
- responsável por contribuir para o orçamento familiar, tendo desde cedo de assumir funções de adulto.

O AEP contribui diretamente para a taxa de desemprego jovem. No entanto, este nem sempre leva ao desemprego. Frequentemente, o AEP é motivado pela possibilidade de uma oportunidade de emprego [4].

Com isto, e segundo um estudo da McKinsey&Company, atualmente, os Governos e as empresas deparam-se com um problema pertinente: existem altas taxas de desemprego jovem e, ao mesmo tempo, existe uma elevada escassez de candidatos a empregos que requerem competências críticas. Claramente, existe a necessidade de as empresas trabalharem diretamente com as instituições de ensino, por forma a que os alunos aprendam as competências necessárias para terem sucesso na posição que ocupam [5]. Estas medidas são importantes e permitem facilitar e agilizar o processo de transição entre o meio educacional e o meio profissional, por forma a combater o AEP e a desmotivação dos alunos no ensino.

1.2 Problema

A presente dissertação insere-se no contexto de um projeto europeu, intitulado “**From birth to adult age – a WBL² successful Practice!**”³. Este projeto, cofinanciado pelo programa europeu Erasmus+, tem o objetivo de ir ao encontro dos objetivos definidos na estratégia europeia para o ensino e combater os problemas identificados na secção 1.1. Aliando indústrias do setor da Metalurgia, Mecânica, Mecatrónica e Manutenção Industrial, e as instituições de ensino profissional, pretende capacitar os jovens com as competências necessárias para atuarem nestas áreas. É também um objetivo do mesmo reformular a forma como estas instituições de ensino atuam, já que seguindo os planos curriculares oficiais não conseguem atingir os objetivos propostos pelas indústrias, nem preparar tecnicamente os alunos de forma eficaz. Assim, um dos *outcomes* esperados é um curriculum baseado em competências para cada uma das áreas industriais acima mencionadas.

Este projeto é uma parceria entre o IPP (Instituto Politécnico do Porto), aqui representado pelo GECAD (Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento), três escolas de ensino profissional, três indústrias que atuam nas áreas acima mencionadas e três órgãos públicos de três países (Portugal, Espanha e Itália).

Um outro *outcome* esperado é o desenvolvimento de uma solução de *software* que permita auxiliar as escolas e as indústrias no processo de validação das competências adquiridas pelos

² WBL – *Work-based learning*

³ Sítio na web: <http://wblsp.aeva.eu/>

alunos. Também é expectável que realize a análise de dados (apresentação de indicadores) e a geração de relatórios, por forma a auxiliar no processo de tomada de decisão.

1.3 Objetivos

O objetivo desta dissertação é desenvolver a solução de *software* que irá auxiliar as escolas e indústrias nos processos de creditação de alunos e de tomada de decisão. Pretende-se também apoiar na gestão dos *outcomes*⁴ de cada Unidade Curricular, avaliar os alunos face a cada *outcome* e auxiliar os docentes e organizações a gerir as tarefas de avaliação das competências dos alunos.

Neste sentido, e para cumprir com o objetivo mencionado, foram definidas um conjunto de tarefas específicas, tais como:

- Efetuar o levantamento e especificação de requisitos;
- Estudar bibliografia referente ao tema do projeto;
- Documentar as decisões tomadas sobre o design arquitetural;
- Desenvolvimento de protótipo que:
 - Permita a gestão de acessos;
 - Forneça mecanismos de autenticação;
 - Permita a gestão de alunos, cursos, turmas e disciplinas;
 - Permita a gestão de *outcomes* de um curso;
 - A cada unidade curricular, permitir e apoiar na realização da avaliação dos alunos em termos de resultados da aprendizagem e objetivos educacionais do Curso. A plataforma deve permitir o rastreamento dessas avaliações;
 - Para um aluno finalista, permita apresentar automaticamente como complemento ao diploma, os *outcomes* atingidos;
 - Possibilite a apresentação automática de relatórios por curso e unidade curricular;
 - Possibilite a integração automática de dados (nas suas escolas de origem) de alunos que participam em programas de mobilidade;
 - Forneça mecanismos para apoiar a avaliação do programa/currículo, que determina a medida em que os resultados dos alunos e os objetivos educacionais do curso/currículo estão sendo alcançados;

⁴ Os *outcomes* caracterizam os conhecimentos e competências que os alunos devem reunir na conclusão do curso.

- Preparar a aplicação para suportar várias línguas (português, inglês, italiano e espanhol);
- Desenvolvimento de *User Interface* (UI) responsiva.
- Realizar testes:
 - Testes unitários;
 - Testes de integração;
 - Testes de sistema;
 - Testes de usabilidade com uma amostra aleatória de utilizadores do sistema.

1.4 Resultados esperados

No final desta dissertação esperam-se atingir os seguintes resultados:

1. **Solução de *software* desenvolvida e completamente funcional:** espera-se que no final do projeto o processo de gestão de utilizadores, de *outcomes*, de cursos, de alunos e da avaliação dos mesmo esteja totalmente funcional. A plataforma será responsiva e suportará várias línguas.
2. **Geração de certificados funcional:** espera-se que o processo de geração de certificados de aquisição de competências esteja totalmente funcional;
3. **Geração de relatórios funcional:** espera-se que o processo de geração de relatórios esteja funcional;
4. **Apresentação de um *dashboard* com indicadores:** espera-se que o processo de análise dos dados que compõem a plataforma esteja funcional, sendo os mesmos apresentados em *dashboards* , o que irá contribuir para o processo de tomada de decisão.
5. **Integração de dados entre aplicações:** espera-se que a integração de dados entre as várias escolas que utilizam a plataforma esteja funcional.
6. **Realização de testes:** espera-se que a plataforma a desenvolver seja validada através de testes unitários, de integração, de sistema e de usabilidade.

1.5 Estrutura do documento

Este documento encontra-se dividido em seis capítulos, sendo estes:

1. **Introdução:** apresenta o contexto da dissertação e a área de negócio em que este se insere, assim como o problema, objetivos e resultados esperados;
2. **Contexto e Estado da Arte:** detalha o contexto e o problema da dissertação, apresentando definições de conceitos de negócio. Apresenta também a análise de valor, identificando o processo de negócio e inovação, o valor para o cliente, o valor percebido, a proposta de valor e o modelo Canvas.
3. **Análise e Design:** especifica os requisitos funcionais e não funcionais identificados para o desenvolvimento da plataforma. É realizada uma análise detalhada dos requisitos por forma a desenvolver uma arquitetura que os suporte. Todas as decisões tomadas são justificadas e, sempre que possível, são apresentadas alternativas às soluções apresentadas.
4. **Implementação:** apresenta os principais detalhes de implementação, com as devidas justificações.
5. **Avaliação:** apresenta a estratégias de testes adotada e os resultados dos mesmos. São também apresentados os resultados dos questionários dos testes de usabilidade.
6. **Conclusão:** apresenta as conclusões desta dissertação, limitações e trabalho futuro.

2 Contexto e Estado da Arte

Por forma a facilitar o entendimento do projeto por parte do leitor, é importante introduzir e detalhar alguns conceitos essenciais. Após a introdução desses conceitos, é apresentada a análise de valor realizada (numa perspetiva de negócio). Finalmente, é apresentado o estado da arte em soluções existentes e tecnologia relevante.

2.1 Contexto

Tal como referido na secção 1.1, o aumento do abandono escolar precoce é um problema identificado na União Europeia. Ao mesmo tempo, as empresas deparam-se com uma escassez de pessoas com as competências necessárias para desempenharem tarefas específicas. Um pouco por todos os Estados Membros, várias medidas têm sido tomadas para combater estes problemas, tais como o plano **Educação e Formação 2020**⁵ (sucessor do plano Educação e Formação 2010) e o **Erasmus+**⁶.

2.1.1 Educação e Formação 2020 (EF 2020)

Cada Estado Membro é responsável pelo seu próprio sistema de ensino e formação. No entanto, é responsabilidade da União Europeia (UE) apoiar e complementar as ações que melhoram e modernizam os sistemas de educação nacionais.

Nesse sentido, o quadro estratégico para a cooperação europeia no domínio da educação e da formação (EF 2020), sucessor do plano Educação e Formação 2010 (EF 2010), é um fórum que visa facilitar a cooperação entre os países da UE tendo em vista a criação de boas práticas [6]. Este fórum é formado por vários grupos de trabalho que colaboram entre si, com o objetivo de ajudar os Estados Membros a desenvolver as suas políticas através de aprendizagem partilhada e identificando boas práticas em matéria de educação e formação [7].

⁵ Sítio na web: https://ec.europa.eu/education/policies/european-policy-cooperation/et2020-framework_pt

⁶ Sítio na web: <https://erasmusmais.eu/>

Criado e reformulado no ano de 2009, após retiradas as conclusões do EF 2010, o EF 2020 mantém os seguintes quatro objetivos comuns [6]:

- tornar realidade a mobilidade e a aprendizagem ao longo da vida;
- melhorar a qualidade e eficácia da educação e da formação;
- promover a igualdade, a coesão social e a cidadania ativa;
- incentivar a criatividade e a inovação, nomeadamente o empreendedorismo, em todos os níveis da educação e formação.

Para além dos quatro objetivos acima apresentados, o EF 2020 também contribui de forma significativa para o cumprimento dos seguintes objetivos presentes do plano “Europa 2020” [6]:

- pelo menos, 95% das crianças deverão frequentar o ensino pré-escolar;
- a percentagem de jovens de 15 anos com níveis de competências insuficientes em leitura, matemática e ciências deve ser inferior a 15%;
- a taxa de abandono escolar entre os jovens dos 18 aos 24 anos deve baixar para menos de 10%;
- pelo menos, 40% da população entre os 30 e os 34 anos com uma qualificação inicial de formação profissional deverão ter realizado um período de estudo ou de formação no estrangeiro;
- pelo menos, 15% dos adultos deverão participar em ações de aprendizagem;
- a percentagem de jovens dos 20 aos 34 anos que tenham concluído, pelo menos, o ensino secundário e encontrado emprego no espaço de um a três anos deve chegar aos 82%.

O plano estipula que estes objetivos têm de ser cumpridos até o ano 2020, aplicando-se uma série de ferramentas e instrumentos, tais como a criação dos grupos de trabalho (já referidos anteriormente), a cooperação entre os Estados Membros em atividades de aprendizagem, avaliação e aconselhamento, e o financiamento de atividades e projetos inovadores através do programa Erasmus+ [6], apresentado na subsecção seguinte.

2.1.2 Erasmus+

O Erasmus+ é o programa da UE que atua nos domínios da educação, da formação, da juventude e do desporto para o período de 2014-2020 [8]. Com um orçamento de 14,7 milhões de euros, este destina-se a apoiar os esforços dos países do programa no sentido de utilizarem o potencial do talento e ativos sociais numa perspetiva de aprendizagem ao longo da vida de forma eficiente, indo de encontro aos objetivos definidos na estratégia “Europa 2020”. O programa também destaca as oportunidades de cooperação e mobilidade junto dos países que o compõem [9].

Para atingir os objetivos propostos, o programa leva a cabo um conjunto de três ações, que segue a seguinte estrutura:

- **Ação-chave 1** – Mobilidade individual;
- **Ação-chave 2** – Cooperação para a inovação e o intercâmbio de boas práticas;
- **Ação-chave 3** – Apoio à reforma das políticas.

No contexto desta dissertação, importa realçar a Ação-chave 2, tal como será explicado posteriormente. Dentro das várias iniciativas apoiadas, esta ação apoia parcerias e alianças estratégicas entre instituições de ensino e empresas, que procuram promover a inovação, o empreendedorismo, a criatividade, a empregabilidade, o intercâmbio de conhecimentos e/ou o ensino e a aprendizagem multidisciplinares [9].

2.1.3 A importância da cooperação entre as instituições de ensino e as empresas

Na secção 1.1 foi apresentado um problema com que os governos e as empresas se deparam hoje em dia: existem altas taxas de desemprego jovem e, ao mesmo tempo, existe uma elevada escassez de candidatos a empregos que requerem competências críticas. Assim, é importante que as empresas e as instituições de ensino trabalhem lado a lado no sentido de formarem os alunos com as competências necessárias para um determinado posto de trabalho.

Atualmente, a qualidade do capital humano é crucial para o êxito da Europa. Para garantir a qualidade do capital humano, é necessário que os planos curriculares dos cursos estejam enquadrados com a realidade atual e que os alunos estejam a adquirir as competências necessárias. Segundo um comunicado da Comissão Europeia, é também essencial o contínuo investimento na educação e na formação para o desenvolvimento de competências, para estimular o crescimento e a competitividade: as competências determinam a capacidade da Europa para aumentar a produtividade. A longo prazo, as competências podem desencadear inovação e crescimento, fazer subir a produção na cadeia de valor, estimular a concentração de competências de nível mais elevado na UE e modelar o mercado de trabalho no futuro [10].

As instituições de Ensino e Formação Profissional (EFP⁷) são um elemento fundamental dos sistemas de aprendizagem ao longo da vida, permitindo às pessoas adquirirem as competências necessárias a uma determinada profissão [11]. O plano estratégico “Europa 2020” realça a importância de reforçar a capacidade de atração deste tipo de ensino e, entre vários objetivos, define o de “promover a aprendizagem em contexto laboral, envolvendo nesse processo os parceiros sociais, as empresas, as câmaras de comércio e os prestadores de EFP, incentivando a inovação e o empreendedorismo” [11].

Assim, existindo a cooperação entre estas instituições e as empresas, e adotando uma abordagem de ensino baseada em trabalho, contribui-se para que os jovens adquiram as competências e os conhecimentos necessários para um primeiro emprego, facilitando a transição entre a escola e o mundo do trabalho [12].

⁷ A sigla EFP é muitas vezes representada na literatura portuguesa pela sigla inglesa VET (*Vocational Education and Training*).

2.1.4 From birth to adult age – a WBL successful Practice

Tal como referido na secção 1.2, a presente dissertação insere-se no contexto de um projeto piloto europeu, intitulado “**From birth to adult age – a WBL successful Practice!**”.

Nas subsecções anteriores foram referidos os problemas do abando escolar precoce e do desemprego jovem, e também as dificuldades sentidas pelas empresas em recrutar pessoas com competências específicas para um determinado posto de trabalho. Foram também referidos os objetivos da União Europeia para combater estes problemas, através do plano estratégico EF 2020, que refere a importância da colaboração entre as instituições de ensino e as empresas para a formação de jovens com competências.

Este projeto piloto surge com o objetivo de cumprir com os objetivos traçados no plano EF 2020, criando uma cooperação entre as instituições de ensino e indústrias do setor da Metalurgia, da Mecânica, da Mecatrónica e Manutenção Industrial. Usando uma metodologia de ensino baseada em trabalho, pretende-se capacitar os alunos com competências para atuarem nestes setores, facilitando a transição entre o ensino e o mundo do trabalho. Também se pretende que os alunos em algum momento do curso participem em programas de mobilidade e realizem parte do ano curricular num país envolvido no piloto.

O projeto é financiado pelo programa Erasmus+, inserindo-se na **Ação-chave 2** (ver subsecção 2.1.2), e tem os seguintes quatro *outputs*:

1. *Curriculum* baseado em competências⁸ para o setor da Manutenção Industrial;
2. *Curriculum* baseado em competências para o setor da Mecatrónica;
3. *Curriculum* baseado em competências para os setores da Mecânica e Metalurgia;
4. Plataforma digital que auxilie no processo de validação de competências.

A Figura 1 representa um excerto de um *curriculum* baseado em competências para o curso de Manutenção Industrial.

COMPONENTE DE ENSINO	DISCIPLINA	ANO E HORAS & ECVETS	OUTCOMES	NOTA *	A	B	C	D	E	F	G	H	I	J	ADQUIRIDO NA (ESCOLA / EMPRESA)		
Tecnológico	Tecnologia & Processos	1º (150h = 9 ECVETS)	Usa instrumentos de medida	15	X						X			X	X		
			a) Conhece os vários instrumentos de medida	15													
			b) Distingue instrumentos de medida, identificando as suas características	15													
			c) Revela habilidades para observar	15													

Figura 1 – Excerto do *curriculum* do curso de Manutenção Industrial

No excerto da Figura 1 encontram-se listados parte dos *outcomes* esperados que um aluno atinja na disciplina de Tecnologia & Processos, do 1º ano do curso de Manutenção Industrial. Um aluno é avaliado face a um *outcome*, tal como podemos verificar pela coluna “Nota”. Um

⁸ Um *curriculum* baseado em competências foca-se nas competências que um aluno deve atingir, em vez daquilo que se espera que os alunos aprendam em termos de conteúdos.

outcome contribuí para a aquisição de *outcomes* do curso (letras de “A” a “J”) e pode ser adquirido na escola, ou na empresa onde o aluno está a realizar o estágio.

No desenvolvimento do projeto participam entidades ligadas à área do ensino, à área empresarial e governamental. Estas entidades pertencem a três países distintos – Portugal, Espanha e Itália. A Tabela 2 apresenta cada uma das entidades envolvidas e os *outputs* associados.

Tabela 2 – Entidades envolvidas no projeto

Entidade	Área de atuação	País	Output (nº)
AEVA – Associação para a educação e valorização da região de Aveiro	Ensino	Portugal	1
A. Silva Matos Metalomecânica	Indústria	Portugal	1
Município de Sever do Vouga	Governamental	Portugal	1
IPP – Instituto Politécnico do Porto (através do GECAD)	Ensino	Portugal	4
Politeknika Ikastegia Txorierrri	Ensino	Espanha	3
TKNIKA	Governamental	Espanha	3
Gestamp Technology Institute	Indústria	Espanha	3
ITIS “E. Mattei”	Ensino	Itália	2
Provincia Di Pesaro e Urbino	Governamental	Itália	2
Benelli Armi SPA	Indústria	Itália	2

O tema desta tese centra-se no desenvolvimento do *output* nº4 do projeto, a plataforma digital que auxilia o processo de validação de competências. Assim, é esperado que esta plataforma permita parametrizar currículos baseados em competências (ver Figura 1). Depois de parametrizados, esta terá de permitir que os professores e orientadores das empresas realizem as avaliações dos alunos. Concluídas as avaliações, e caso um aluno seja finalista e conclua o curso com sucesso, terá de ser possível emitir um certificado de aquisição de competências, que será anexado ao diploma de conclusão de curso. Independentemente deste último cenário, terá de ser possível a qualquer momento gerar relatórios por curso, para tornar possível a análise do seu desenvolvimento.

Tal como foi referido inicialmente, os alunos poderão participar em programas de mobilidade em algum momento do curso. Assim, é esperado que as notas que os alunos obtêm durante o programa de mobilidade sejam automaticamente integradas nas suas escolas de origem.

A imagem da Figura 2 resume os objetivos da plataforma digital a desenvolver, representado os *inputs* e *outputs* esperados.

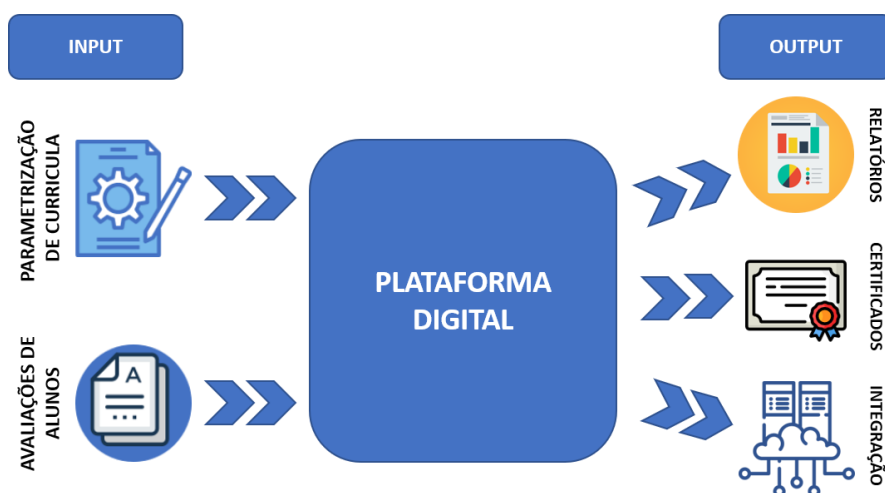


Figura 2 – *Inputs* e *outputs* da plataforma digital a desenvolver

2.2 Áreas abrangidas

A plataforma a desenvolver tem como principal objetivo auxiliar tanto os professores como os formadores das empresas na validação e avaliação de competências dos alunos. Esta é uma plataforma ligada à área do ensino, estando relacionada com a área dos *Learning Management Systems* (LMS), conceito introduzido na subsecção seguinte.

2.2.1 Learning Management Systems (LMS)

Learning Management System (Sistema de Gestão de Aprendizagem) trata-se de uma plataforma de ensino online projetada a partir de uma metodologia pedagógica que tem como objetivo promover e disseminar a educação através da modalidade de ensino à distância [13]. Segundo Hall, um LMS é um software que automatiza a administração de eventos formativos. Todos os LMS gerem o acesso (login) dos utilizadores registados, gerem catálogos de cursos, registam dados dos estudantes e fornecem relatórios à gestão [14]. Estas ferramentas constituem o suporte tecnológico ao e-learning.

As funcionalidades comuns de um LMS são [15]:

- criação de conteúdo;
- comunicação;
- avaliação;
- administração.

A criação de conteúdo permite aos professores criar conteúdo num curso, associando ficheiros, apresentações, imagens, áudio e vídeo. Podem também ser criados *links* para outros *websites* ou para documentação externa ao sistema [16].

As ferramentas típicas de comunicação permitem criar canais de comunicação entre os professores e os alunos, seja através de anúncios colocados nos próprios cursos, envio de emails, fóruns de discussão, *chats* de mensagens, entre outros [16].

As ferramentas de avaliação fornecem aos professores mecanismos para testar e acompanhar os conhecimentos dos alunos. As ferramentas típicas incluem a gestão de exames, gestor de questões e gestor de avaliação de alunos [16].

Por fim, as ferramentas de administração incluem as funcionalidades de gestão e customização de cursos, criação de cursos e utilizadores e monitorização do sistema [16].

Assim, os sistemas LMS focam-se principalmente nos aspetos de gestão de cursos, deixando de lado, em regra geral, o aspeto de gestão/produção de conteúdo.

A Figura 3 apresenta um resumo das funcionalidades típicas de um LMS.

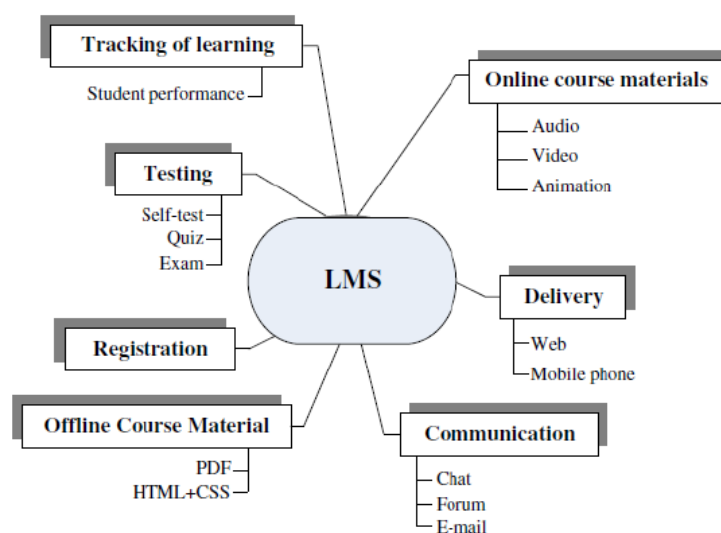


Figura 3 – *Overview* das funcionalidades típicas de um LMS [17]

Existem várias soluções LMS no mercado desenvolvidas para dar suporte ao e-learning. Os mais utilizados na Europa são o Moodle⁹, o OLAT¹⁰ e o Blackboard¹¹. Em seguida são apresentadas cada uma das plataformas mencionadas.

2.2.1.1 Moodle

O Moodle (Modular Object-Oriented Dynamic Learning Environment) é uma plataforma LMS desenvolvida desde 2002, na Austrália. O seu desenvolvimento iniciou-se como uma alternativa às plataformas comerciais existentes na altura.

⁹ Sítio na web: <https://moodle.org>

¹⁰ Sítio na web: <https://olat.org>

¹¹ Sítio na web: <https://blackboard.com>



Figura 4 – Logótipo do Moodle [18]

A plataforma é *open-source* e tem como base as tecnologias PHP e MySQL. A facilidade de instalação de ferramenta faz com que seja um dos LMS mais usados no mundo, e aquele que mais tem desenvolvido [16].

Sendo o Moodle um LMS, é de esperar que este possua as funcionalidades típicas de um sistema deste tipo. No entanto, o Moodle tem um aspeto mais social do que a grande maioria dos LMS existentes, focando-se bastante na facilidade de comunicação e interação social entre os vários utilizadores. Os conceitos “pessoas”, “eventos” e “últimas notícias” têm sempre um principal destaque nas páginas dos cursos.

Uma das mais recentes funcionalidades do Moodle é o ensino baseado em competências. Esta funcionalidade permite ao administrador criar quadros de competências e planos de aprendizagem. O professor poderá, posteriormente, associar determinadas competências às suas disciplinas e também às suas atividades. Os alunos serão avaliados de acordo com o seu grau de proficiências nas competências presentes nos seus planos de aprendizagem [19].

2.2.1.2 OLAT

OLAT é outra plataforma LMS *open-source* existente no mercado. O desenvolvimento desta solução iniciou-se em 1999, na Universidade de Zurique (Suíça). A primeira versão foi instalada em 2000, na própria universidade.

Tal como o Moodle, o OLAT também tem como base a tecnologia PHP, tendo sido atualizada para Java em 2004, por forma a responder a novos requisitos de performance [20].



Figura 5 – Logótipo da plataforma OLAT [21]

Foi realizada uma extensa revisão na literatura e não se identificou nenhuma evidência de que o OLAT suporte o ensino baseado em competências.

2.2.1.3 Blackboard

O Blackboard é um LMS proprietário que foi criado pela Blackboard, Inc, com sede em Washington D.C, USA. A empresa foi fundada em junho de 1997 e domina o setor da educação dos Estados Unidos, sendo detentora de cerca de 57% do mercado educacional [22]. Este

produto acabou por dominar o mercado devido à política que a empresa adotou de adquirir os produtos concorrentes [16, p. 9]. Várias instituições de todo o mundo utilizam este produto, como por exemplo Georgetown University, Arizona State University e Singapore Polytechnic.



Blackboard

Figura 6 – Logótipo do Blackboard [23]

Da pesquisa realizada sobre a plataforma, foram encontradas evidências de que o Blackboard suporta o ensino baseado em competências, permitindo [24]:

- Associar competências a cursos;
- Associar competências de um curso às suas atividades.

No entanto, é importante voltar a realçar que esta plataforma não é gratuita.

2.2.1.4 Comparação das plataformas

A tabela compara as plataformas analisadas quanto às tarefas típicas de um LMS, assim como ao suporte de ensino baseado em competências.

Tabela 3 – Comparação dos três produtos apresentados

Funcionalidade	Moodle	Blackboard	OLAT
<i>Open-source</i>	sim	não	sim
Gestão de cursos	sim	sim	sim
Gestão de utilizadores	Sim	sim	sim
Gestão de disciplinas	sim	sim	sim
Gestão de competências	sim (desde 2016)	sim	não
Geração de relatórios	sim	sim	sim
Emissão de certificados baseados nas competências adquiridas	não	sim	não

Analisando a Tabela 3 verifica-se que o Blackboard é a solução mais completa e a que fornece um maior suporte ao ensino baseado em competências. No entanto, este produto é pago.

Verifica-se também que o Moodle suporta o ensino baseado em competências desde 2016, mas ainda não permitem a emissão de certificados baseados em competências adquiridas. O facto de apenas suportar esta funcionalidade desde 2016 pode ser uma evidência de que a funcionalidade ainda se encontra numa fase prematura.

Sobre o OLAT, o produto é *open-source*, no entanto não fornece qualquer tipo de suporte para o ensino baseado em competências.

2.3 Análise de Valor

Nesta secção é apresentada a análise de valor da presente dissertação. Inicialmente é apresentado o processo de inovação segundo o modelo de Peter Koen, sendo em seguida apresentado a proposta de valor e o modelo Canvas.

2.3.1 The New Concept Development Model (NCD)

Nesta subsecção apresentam-se as respostas a cada um dos pontos definidos pelo modelo NCD.

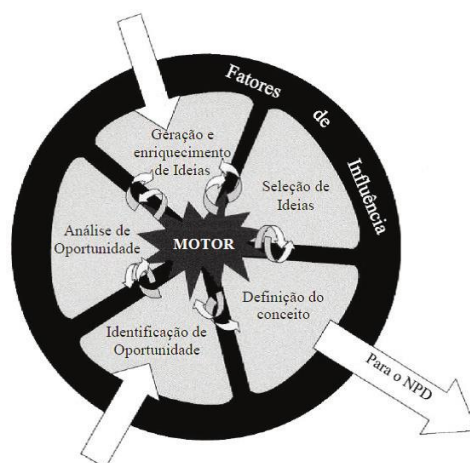


Figura 7 – Modelo New Concept Development (NCD) [25]

Identificação da Oportunidade

Tal como apresentado anteriormente, a identificação da oportunidade surge após serem publicados os objetivos para a educação e formação da estratégia “Europa 2020”. Esses objetivos especificam a importância do combate ao abandono escolar precoce, a importância do desenvolvimento de competências e da cooperação das instituições de ensino com as empresas, e do papel fundamental que as instituições EFP têm neste processo.

Análise da Oportunidade

Identificada a oportunidade verifica-se que as indústrias dos setores da Metalurgia, Mecânica, Mecatrónica e Manutenção Industrial da zona sul da Europa sentem dificuldades em encontrar pessoas com as competências necessárias para desempenhar uma determinada profissão.

Geração da ideia

Com isto, surge a ideia do desenvolvimento de parcerias entre empresas destes setores e de instituições de ensino EFP.

Seleção da ideia

Nesta parceria pretende-se desenvolver os currículos para cada um destes setores, especificando que competências cada aluno deve adquirir. Será necessária uma plataforma informática para validar estas competências.

Desenvolvimento/Definição de Conceito

Depois de selecionada a ideia, é necessário realizar uma análise sobre como a plataforma será implementada, bem como identificar todos os requisitos e especificações que irão garantir a satisfação do problema.

2.3.2 Proposta de Valor

O sistema a desenvolver trata-se de um *outcome* de um projeto europeu, logo o valor gerado para todos os intervenientes no projeto é inquestionável. Para além disso, trata-se de um LMS desenvolvido orientado ao ensino profissional e ao ensino baseado em competências, trazendo um grande valor para todas as entidades envolvidas nestas áreas de ensino. O facto de este ter sido desenvolvido orientado a uma área de ensino específica não significa que o mesmo seja exclusivo a essa a área. O sistema poderá ser usado noutras áreas de ensino, ou até mesmo empresariais.

2.3.3 Modelo Canvas

A Figura 8 apresenta o modelo Canvas do projeto.

Parcerias Principais IPP Empresas	Atividades Chave Desenvolvimento de software Desenvolvimento de plataforma de apoio à decisão Desenvolvimento de plataforma de validação de competências de alunos	Proposta de Valor Apoio na gestão de cursos Apoio na avaliação de alunos Apoio na validação de competência Plataforma independente da área de ensino/setor empresarial	Relacionamento com Clientes Suporte Site oficial	Segmentos de Clientes Instituições de ensino Empresas
	Recursos Principais Equipa de desenvolvimento		Canais Fóruns ligados ao ensino Site oficial	
Custos Alojamento da plataforma web Material informático Equipa de desenvolvimento		Receitas Depois da fase de piloto poderá ser comercializada e ter uma fonte de receita através da venda de licenças de uso		

Figura 8 – Modelo Canvas

3 Análise e Design

“Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.”

Eoin Woods

Os sistemas de *software* são, em regra, construídos para satisfazer os objetivos de negócio de uma organização. É importante que, desde o começo dos projetos, a arquitetura se aproxime e tente responder aos requisitos levantados.

Com o capítulo de Análise e Design pretende-se apresentar o domínio da aplicação a desenvolver e analisar e especificar os requisitos funcionais e não. Após essa especificação, são apresentadas várias arquiteturas que prometem responder aos requisitos identificados. Em cada uma das arquiteturas é realizada uma análise detalhada das suas vantagens e desvantagens, por forma a concluir qual a que melhor se adequa ao sistema a desenvolver.

As diferentes arquiteturas apresentadas foram documentadas seguindo a metodologia *Views and Beyond* (V&B), desenvolvida pelo Software Engineering Institute¹² (SEI). Este metodologia utiliza o conceito de vista (*view*) como o “princípio fundamental para organizar a documentação de uma arquitetura” [26, p. 19].

¹² Sítio na web: <https://www.sei.cmu.edu>

3.1 Visão geral

A aplicação “Educa”¹³ insere-se na área da educação e surge como um resultado do projeto europeu apresentado na subsecção 2.1.4. O seu principal objetivo é auxiliar os diretores de curso, professores e orientadores das empresas no processo de avaliação e validação de competências dos alunos, permitindo a parametrização de cursos e disciplinas (através dos currícula), a geração de relatórios por curso, a emissão de certificados de aquisição de competências e a integração automática das notas dos estudantes que frequentam programas de mobilidade.

O projeto envolve três escolas de ensino profissional situadas em países distintos (Portugal, Espanha e Itália). Assim, espera-se que cada escola tenha uma instância da aplicação a desenvolver e que todas as instâncias comuniquem entre si para a troca de dados.

Os *stakeholders* do projeto são todas as partes envolvidas no projeto europeu, tais como as escolas, empresas e as instituições governamentais. Sendo este projeto financiado pelo programa Erasmus+, também se pode considerar Comissão Europeia como uma parte interessada.

3.2 Engenharia de requisitos

A engenharia de requisitos é uma atividade que se preocupa em identificar e comunicar o propósito de uma aplicação, e o contexto em que a mesma se irá inserir. Assim, esta funciona como uma atividade crucial para se perceber aquilo que os principais *stakeholders* da aplicação pretendem [27]. Por vezes, uma má engenharia de requisitos pode ditar no fracasso de um projeto, e, por esse motivo, foi dada uma elevada relevância a essa tarefa durante o desenvolvimento deste projeto.

Para perceber o domínio do problema e identificar os requisitos, foram realizadas várias reuniões com alguns dos *stakeholders*, assim como com alguns professores da Escola Profissional de Aveiro. Nas reuniões com os professores procurou-se perceber como funcionam os processos atuais de avaliação nos sistemas informáticos existentes até à data.

Nas subsecções seguintes são apresentados os resultados destas reuniões.

3.2.1 Atores do sistema

Identificar e classificar os atores do sistema numa fase preliminar do projeto é essencial para o sucesso da fase de identificação de requisitos. A tabela lista os diferentes atores do sistema e as suas responsabilidades.

¹³ Nome dado à plataforma digital a desenvolver

Tabela 4 – Atores do sistema

Ator	Descrição
Administrador	Representa o administrador de todo o ecossistema de aplicações em execução. Tem a responsabilidade de parametrizar uma nova instância da aplicação.
Administrador de instância	Representa o administrador de uma instância específica da aplicação. Tem a responsabilidade de configurar acessos de utilizadores e de criar cursos.
Diretor de curso	Responsável pela gestão e configuração de um curso. Tem a responsabilidade de parametrizar o currículo do curso, de criar as turmas e de gerar certificados.
Professor	Responsável pela avaliação dos alunos.
Orientador da empresa	Responsável pela avaliação dos alunos.

3.2.2 Modelo de domínio

O modelo de domínio é um artefacto que permite descrever e modelar entidades do mundo real e as relações entre elas, que, coletivamente, descrevem o espaço de domínio do problema. Estando fortemente acoplado aos conceitos do negócio, este modelo torna-se num dos mais indicados para comunicar com *stakeholders* não técnicos [28, p. 6]. Os conceitos representados num modelo de domínio são classes candidatas, que podem, ou não, transitar para um modelo de classes. A Figura 9 mostra o modelo de domínio definido para este problema. De notar que este modelo foi construído ao longo das primeiras fases do projeto, à medida que as reuniões com os *stakeholders* se iam desenvolvendo.

Para enquadrar o leitor nos conceitos apresentados no modelo de domínio da Figura 9, apresentam-se em seguida as suas definições, que são:

- **Escola:** representa a escola para uma instância em específico da aplicação “Educa”. Possui um código, que é único no ecossistema de instâncias da aplicação. Uma Escola é composta por diretores de curso, professores, estudantes, cursos e empresas.
- **Diretor de Curso:** representa um Diretor de Curso. Este está associado ao Curso, pois é responsável sua gestão.
- **Professor:** representa um Professor. Um professor pode lecionar várias disciplinas, em vários cursos e turmas.
- **Empresa:** representa a Empresa onde os estudantes realizam os estágios. Esta está associada à escola. Uma empresa é composta por vários Orientadores de Empresa, sendo estes os orientadores que acompanham os alunos nos estágios.
- **Estudante:** representa um estudante da escola. Este pode ser um estudante externo caso esteja a frequentar algum programa de mobilidade.

- **Curso:** representa um determinado Curso da escola. Todos os cursos são agrupados por **Ano Letivo**, o que permite criar um histórico no sistema. Um curso tem um **Curriculum**, é composto por várias **Turmas** e pode ter vários **Outcomes**. Um *outcome* é aquilo que se espera que um aluno adquira caso conclua o curso com sucesso. Um curso tem um **Estado**, sendo que este pode ser:
 - **Draft** – o curso está criado no sistema, mas apenas visível para o Diretor de Curso. O diretor pode realizar todas as tarefas de configuração do curso que apenas ele o visualiza;
 - **Criado** – o curso encontra-se completamente configurado e visível para todos os professores e orientadores de empresa que pertencem ao curso.
- **Turma:** representa uma Turma de um curso. Uma turma agrega vários Estudantes.
- **Curriculum:** representa o curriculum de um Curso. Este curriculum é composto pelas várias **Disciplinas** do curso. Das várias propriedades que definem uma disciplina, importa realçar os ECVETS, o European Credit System for Vocational Education and Training. Fazendo uma analogia com o sistema de Ensino Superior, este sistema é equivalente aos ECTS (European Credit Transfer System). Uma disciplina é composta por **Outcomes de Disciplina**, que é aquilo que se espera que um aluno adquira caso seja aprovado com sucesso numa disciplina. Um Outcome de Disciplina é composto por um conjunto de **Aprendizagens Essenciais**, que significam aquilo que um aluno deve atingir para adquirir o Outcome de Disciplina.
- **Certificado:** representa o certificado de aquisição de competências de um estudante, que pode ser emitido quando um estudante termina um curso com sucesso.
- **Avaliação:** representa uma avaliação de um estudante numa disciplina específica. No processo de avaliação podem participar professores, orientadores de empresa e diretor de curso. Visto que esta aplicação pode funcionar em vários países, e que cada país possui a sua escala de notas, uma avaliação possui uma **Escala da Avaliação**. Para a versão a desenvolver foram definidas três escalas, que se encontram especificadas na Tabela 5.

Tabela 5 – Escalas de avaliação

Escala	Nota mínima	Nota máxima	Condição de aprovação
Base	0	20	nota \geq 10
Espanhola	0	10	nota \geq 5
Italiana	0	10	nota \geq 6

Um estudante é avaliado a cada uma das aprendizagens essenciais, de cada um dos *outcomes* da disciplina em avaliação. A nota final do *outcome* é o resultado da média das notas de todas as aprendizagens essenciais. Já a nota final da disciplina é o resultado da média de todas as notas finais dos *outcomes*.

A nota final do curso é calculada após o estudante ter sido avaliado a todas as disciplinas. O seu resultado é a média das notas finais de todas as disciplinas do curso.

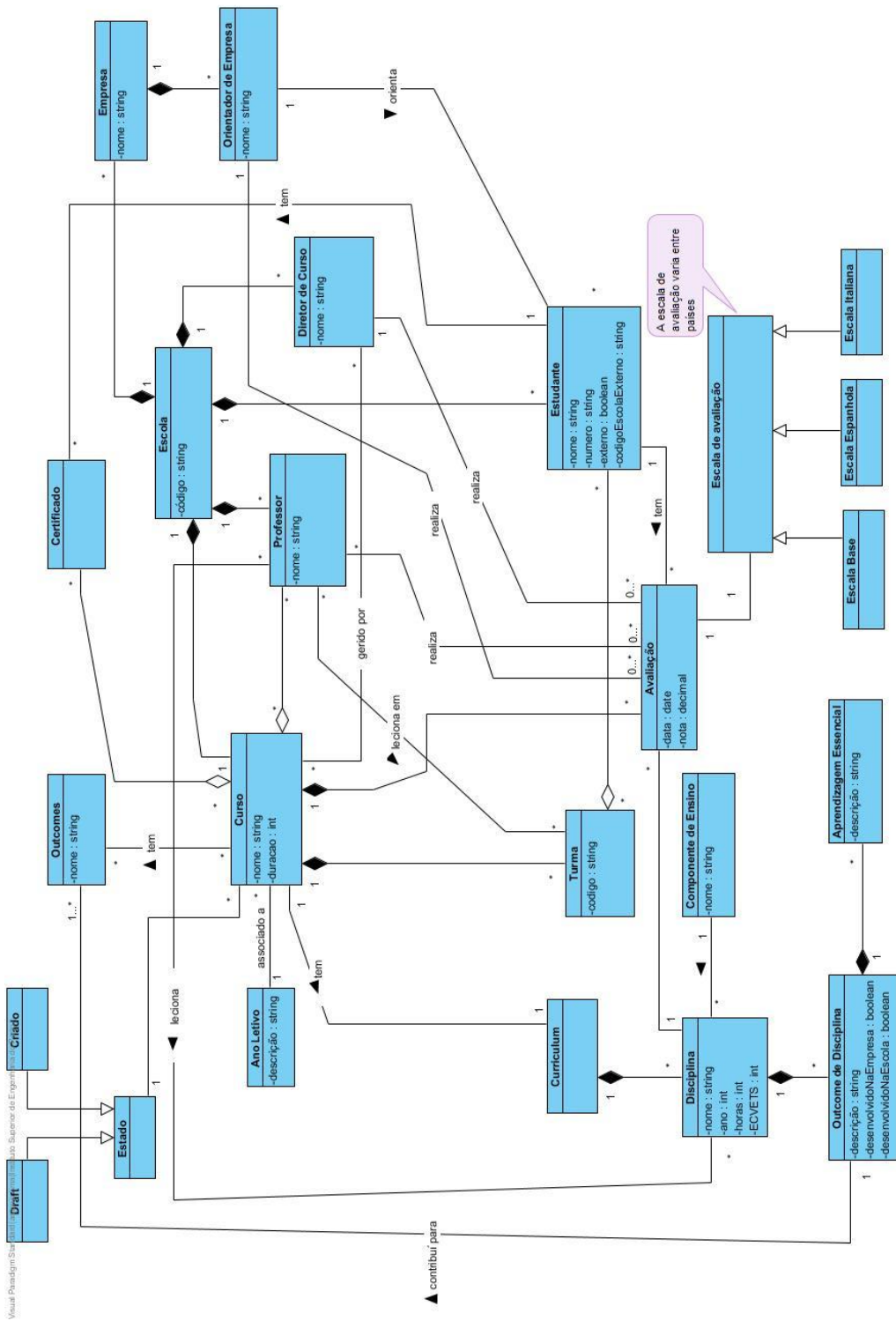


Figura 9 – Modelo de domínio

3.2.3 Requisitos Funcionais

Os requisitos funcionais são requisitos que, tipicamente, especificam um comportamento ou uma funcionalidade [27, p. 23]. Os vários requisitos funcionais identificados encontram-se representados nas vistas funcionais que se seguem. Os requisitos foram divididos em várias vistas funcionais para facilitar a apresentação. Visto que o âmbito deste documento não é a especificação de requisitos, apenas os mais relevantes serão detalhados.

A Figura 10 apresenta parte dos casos de uso do Administrador de Instância, relacionados com funcionalidades de administração.

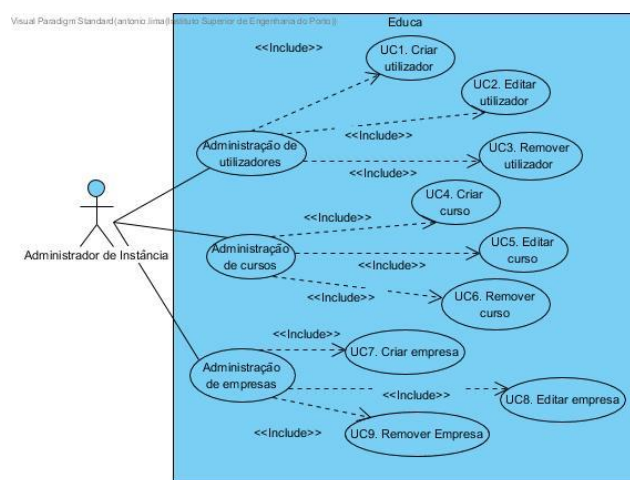


Figura 10 – Vista funcional do administrador de instância

As duas tabelas seguintes (Tabela 6 e Tabela 7) detalham o caso de uso “UC1. Criar Utilizador” e “UC4. Criar Utilizador”, respetivamente.

Tabela 6 – Especificação do caso de uso “UC1. Criar Utilizador”

Código: UC1	Ator: Administrador de instância
Pré-condições O utilizador está autenticado	
Pós-condições O novo utilizador é registado no sistema	
Cenário de sucesso principal (fluxo básico) <ol style="list-style-type: none"> 1. O administrador inicia no sistema o processo de criação de um novo utilizador. 2. O sistema solicita o preenchimento do primeiro e último nome, email, <i>password</i> e cargo. 3. O administrador preenche a informação solicitada. 4. O sistema valida a informação. 5. O sistema regista o novo utilizador. 	
Extensões (fluxos alternativos) 2a. O administrador seleciona o cargo de “Orientador de Empresa”. <ol style="list-style-type: none"> 1. O sistema solicita o preenchimento do campo “Empresa”. 2. O administrador seleciona a empresa. 4a. O sistema deteta informação inválida. <ol style="list-style-type: none"> 1. O sistema informa o utilizador através de uma mensagem de erro. 	

<p>Requisitos especiais</p> <ul style="list-style-type: none"> • Caso o administrador seleciona o cargo “Orientador de Empresa”, têm de existir empresas no sistema para o caso de uso prosseguir. • A <i>password</i> tem de cumprir o seguinte formato: mínimo de 6 caracteres, mínimo de 1 letra maiúscula, mínimo de 1 número, mínimo de 1 caracter especial. • Os campos “Primeiro Nome”, “Cargo”, “Email” e “Password” são de preenchimento obrigatório.

Tabela 7 – Especificação do caso de uso “UC4. Criar Curso”

Código: UC4	Ator: Administrador de instância
<p>Pré-condições</p> <p>O utilizador está autenticado; Existe pelo menos um Diretor de Curso registado no sistema</p>	
<p>Pós-condições</p> <p>O novo curso é registado no sistema</p>	
<p>Cenário de sucesso principal (fluxo básico)</p> <ol style="list-style-type: none"> 1. O administrador inicia no sistema o processo de criação de um novo curso. 2. O sistema solicita o preenchimento do nome do curso, número de anos e ano letivo. 3. O administrador preenche a informação solicitada. 4. O sistema valida a informação. 5. O sistema solicita o preenchimento do campo “Diretor de Curso” e “Professores”. 6. O administrador preenche a informação. 7. O administrador carrega um ficheiro com os alunos do curso. 8. O sistema valida a informação. 9. O sistema regista o curso. 	
<p>Extensões (fluxos alternativos)</p> <p>4a. O sistema deteta informação inválida.</p> <ol style="list-style-type: none"> 1. O sistema informa o utilizador através de uma mensagem de erro. <p>7a. O ficheiro não é válido.</p> <ol style="list-style-type: none"> 1. O sistema informa o utilizador e solicita um novo carregamento. <p>7b. O utilizador não carrega o ficheiro de alunos.</p> <ol style="list-style-type: none"> 1. O caso de uso prossegue. <p>8a. O sistema deteta informação inválida.</p> <ol style="list-style-type: none"> 1. O sistema informa o utilizador através de uma mensagem de erro. 	
<p>Requisitos especiais</p> <ul style="list-style-type: none"> • A informação dos estudantes colocada no ficheiro tem de respeitar o formato solicitado. • Os campos nome do curso, número de anos, ano letivo e diretor de curso são de preenchimento obrigatório. • Pode ser selecionado mais do que um Diretor de Curso. 	

A Figura 11 apresenta os restantes casos de uso do Administrador de Instância e parte dos casos de uso do Diretor de Curso.

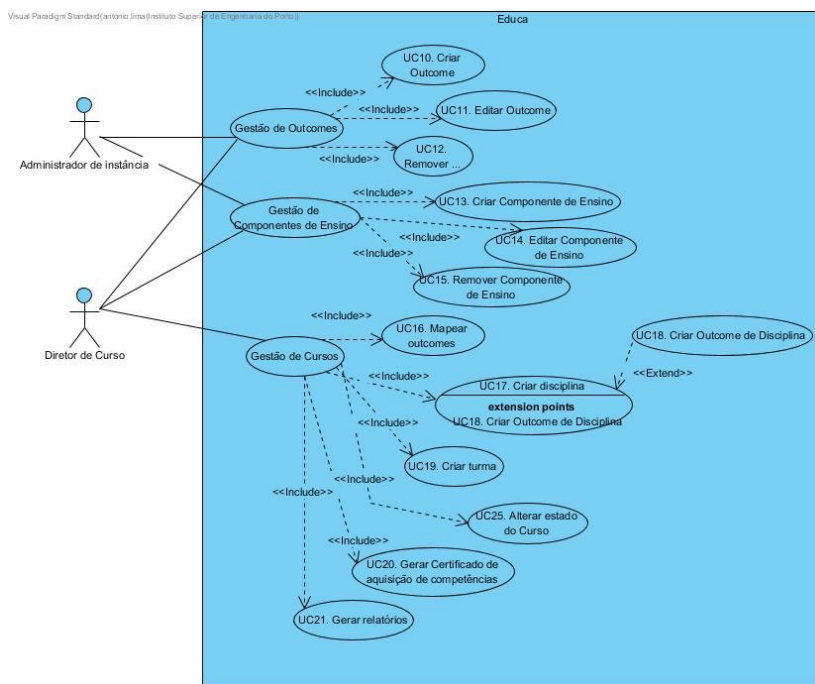


Figura 11 – Vista funcional do Administrador de Instância e Diretor de Curso

As duas tabelas seguintes (Tabela 8 e Tabela 9) detalham os casos de uso “UC18. Criar Outcome de Disciplina” e “UC20. Gerar Certificado de aquisição de competências”, respetivamente.

Tabela 8 - Especificação do caso de uso “UC18. Criar Outcome de disciplina”

Código: UC18	Ator: Diretor de Curso
Pré-condições O utilizador está autenticado;	
Pós-condições O <i>outcome</i> é associado à disciplina e registado no sistema.	
Cenário de sucesso principal (fluxo básico) <ol style="list-style-type: none"> 1. O diretor de curso inicia no sistema o processo de criação de um novo <i>outcome</i>. 2. O sistema solicita o preenchimento do nome do <i>outcome</i>, das aprendizagens essenciais, do(s) <i>outcome(s)</i> do curso para o qual este <i>outcome</i> contribuí e do local de aquisição (escola ou empresa). 3. O administrador preenche a informação solicitada. 4. O sistema valida a informação. 5. O sistema regista o <i>outcome</i>. 	
Extensões (fluxos alternativos) 4a. O sistema deteta informação inválida. <ol style="list-style-type: none"> 1. O sistema informa o utilizador através de uma mensagem de erro. 	
Requisitos especiais <ul style="list-style-type: none"> • Todos os campos são de preenchimento obrigatório 	

Tabela 9 - Especificação do caso de uso “UC20. Gerar Certificado de aquisição de competências”

Código: UC20	Ator: Diretor de Curso
Pré-condições O utilizador está autenticado;	
Pós-condições n/a	
Cenário de sucesso principal (fluxo básico) <ol style="list-style-type: none"> 1. O diretor de curso inicia no sistema o processo de geração de certificado de aquisição de competências. 2. O sistema lista todos os alunos elegíveis para geração de certificado. 3. O diretor de curso seleciona um aluno. 4. O sistema gera o certificado. 	
Extensões (fluxos alternativos) <ol style="list-style-type: none"> 2a. Não existem alunos elegíveis para geração de certificado. <ol style="list-style-type: none"> 1. O caso de uso termina. 	
Requisitos especiais <ul style="list-style-type: none"> • n/a 	

A Figura 12 apresenta os restantes casos de uso do Diretor de Curso e os casos de uso do Professor e Orientador de Empresa.

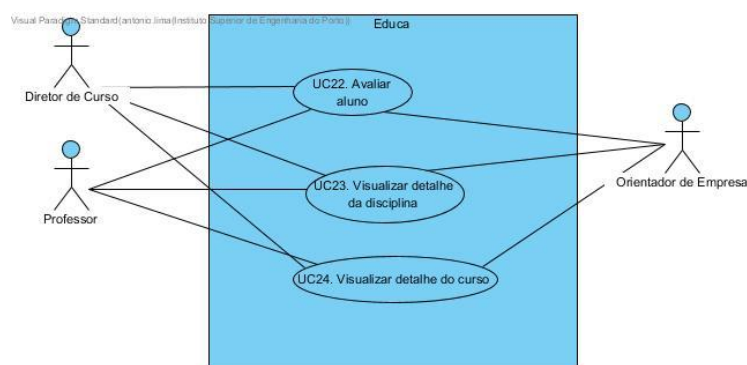


Figura 12 - Vista funcional do Diretor de Curso, Professor e Orientador de Empresa

A Tabela 10 detalha o caso de uso “UC.22 Avaliar aluno”.

Tabela 10 - Especificação do caso de uso “UC22. Avaliar aluno”

Código: UC22	Ator: Diretor de Curso, Professor e Orientador de Empresa
Pré-condições O utilizador está autenticado;	
Pós-condições A avaliação é registada no sistema.	
Cenário de sucesso principal (fluxo básico) <ol style="list-style-type: none"> 1. O utilizador inicia no sistema o processo de avaliação de um aluno. 2. O sistema lista as turmas do utilizador. 3. O utilizador seleciona uma turma. 4. O sistema lista os alunos da turma selecionada. 5. O utilizador seleciona um aluno. 	

<ol style="list-style-type: none"> 6. O sistema carrega o formulário de avaliação. 7. O utilizador preenche a nota das aprendizagens essenciais do <i>outcome</i> que pretende avaliar. 8. O utilizador submete o formulário de avaliação. 9. O sistema valida o formulário. 10. O sistema guarda a informação.
<p>Extensões (fluxos alternativos)</p> <p>9a. O sistema deteta informação inválida.</p> <ol style="list-style-type: none"> 1. O sistema informa o utilizador através de uma mensagem de erro.
<p>Requisitos especiais</p> <ul style="list-style-type: none"> • O Diretor de Curso pode avaliar todos os <i>outcomes</i> da disciplina; • O Professor apenas pode avaliar os <i>outcomes</i> adquiridos na escola; • O Orientador de Empresa apenas pode avaliar os <i>outcomes</i> adquiridos na empresa; • Todos os <i>outcomes</i> da disciplina são listados para avaliação, mas se o utilizador não tiver permissão para o avaliar, o <i>outcome</i> terá de ser listado em modo de leitura; • O Diretor de Curso tem um campo adicional para comentários.

3.2.4 Requisitos não funcionais

Contrariamente aos requisitos funcionais, o objetivo dos requisitos não funcionais não passa por encontrar funcionalidades, mas sim por especificar a forma como o sistema se deve comportar. Um requisito não funcional pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como uma restrição geral do sistema [29, p. 141]. Para capturar este tipo de requisitos foi utilizado o modelo FURPS+, desenvolvido pela Hewlett-Packard. A Tabela 11 apresenta todos os requisitos identificados.

Tabela 11 – Requisitos não funcionais identificados

Id	Descrição	Categoria
RNF01	O sistema deve permitir que as notas dos estudantes que participam em programas de mobilidade sejam automaticamente integradas nas suas escolas de origem	Funcionalidade
RNF02	O sistema deve fornecer um mecanismo de autenticação seguro	Funcionalidade
RNF03	O sistema deve implementar mecanismos de autorização	Funcionalidade
RNF04	Todas as tecnologias usadas têm de ser <i>open source</i>	Implementação
RNF05	A UI do sistema tem de ser responsiva	Usabilidade
RNF06	A UI do sistema tem de ser intuitiva	Usabilidade
RNF07	A UI do sistema tem de suportar as línguas portuguesa, inglesa, espanhola e italiana	Suportabilidade
RNF08	Os relatórios e os certificados têm de ser gerados, no máximo, em 3 segundos	Performance
RNF09	O sistema tem de ser facilmente testado	Suportabilidade
RNF10	O sistema tem de ter uma boa resposta em épocas de avaliação	Confiabilidade

Id	Descrição	Categoria
RNF11	O sistema tem de ser extensível	Suportabilidade
RNF12	O acesso à aplicação deve ser feito através de um <i>browser</i>	Design
RNF13	O sistema tem de ser de fácil manutenibilidade	Suportabilidade
RNF14	O Sistema de Base de Dados a usar é o MySQL	Design

3.3 Tecnologias relevantes

O primeiro passo para selecionar as tecnologias do projeto foi analisar os requisitos não funcionais levantados e identificar os que influenciam a seleção. Analisando a Tabela 11, destacam-se os requisitos RNF04 e RNF14. Estes indicam, respetivamente, que “todas as tecnologias usadas têm de ser *open source*” e que “o sistema de base de dados a usar é o MySQL”.

O segundo passo foi refletir sobre os conhecimentos tecnológicos do autor, tendo em consideração o tempo disponível para o desenvolvimento da solução. Após esta reflexão, conclui-se que o autor domina a *stack* de tecnologias Microsoft, sendo essa a *stack* de tecnologias a explorar.

3.3.1 Aplicação web

A Microsoft oferece um conjunto de tecnologias para o desenvolvimento de aplicações *web*, que são:

1. ASP.NET Web Forms;
2. ASP.NET MVC;
3. ASP.NET Core.

Apesar de todas serem usadas para o desenvolvimento de aplicações *web*, nem todas podem ser aplicadas no contexto deste projeto. As duas primeiras tecnologias (ASP.NET Web Forms e ASP.NET MVC) apenas executam em servidores Windows Server. Segundo a informação disponível na página do produto¹⁴, a licença não é gratuita, o que, seguindo o requisito não funcional RNF04, obriga a que estas duas opções sejam descartadas.

ASP.NET Core é uma *framework open-source* que permite o desenvolvimento de aplicações para a *cloud* ou *on-premises*¹⁵, podendo ser desenvolvida e executada em Windows, Linux ou macOS [30]. A *framework* pode ser utilizada tanto para o desenvolvimento de aplicações *web*,

¹⁴ Sítio na *web*: <https://www.microsoft.com/pt-pt/cloud-platform/windows-server-pricing>

¹⁵ Aplicação que é instalada e gerida na infraestrutura do cliente ou da entidade responsável pelo desenvolvimento

como de APIs. Esta *framework* também contém um sistema de autenticação seguro, o ASP.NET Core Identity. O IDE utilizado para o desenvolvimento pode ser o novo Visual Studio Code¹⁶, que é gratuito e executa em ambientes Linux e macOS.

Após esta análise torna-se evidente que a tecnologia ASP.NET Core será a escolha final, já que entre as tecnologias analisadas, esta é a única que respeita os requisitos não funcionais RNF02 e RNF04. A Tabela 12 resume a análise efetuada.

Tabela 12 – Comparação de tecnologias Microsoft

	Ano de lançamento	Open-source	IDE gratuito	Executa em Linux
ASP.NET Web Forms	2002	não	não	não
ASP.NET MVC	2007	sim	não	não
ASP.NET Core	2016	sim	sim	sim

O desenvolvimento da interface do utilizador será realizado recorrendo a HTML5, CSS3, JQuery e Bootstrap 3, sendo que esta última é uma *framework* bastante útil no desenvolvimento de interfaces responsivas.

3.3.2 Base de dados

A Microsoft oferece o Microsoft SQL Server como o sistema de base de dados. No entanto, neste projeto, o cliente impôs o MySQL como sistema de base de dados a usar, sendo essa a opção a seguir.

Na subsecção anterior foi referido que a tecnologia ASP.NET Core será usada para o desenvolvimento da aplicação. Assim, o ORM (Object-relational mapper) utilizado será o Entity Framework Core. Esta ferramenta também é *open-source* e pode executar em Windows, Linux ou macOS [31].

3.4 Design Arquitetural

A arquitetura de um sistema dita se os requisitos de qualidade (não funcionais) serão aplicados com sucesso, o que torna a disciplina de *design* arquitetural numa das fases mais importantes no desenvolvimento de um *software* [32, p. 41].

Esta secção apresenta a arquitetura definida para o desenvolvimento do *software*. Sempre que possível são apresentadas e analisadas diferentes alternativas.

3.4.1 Clean Architecture

Para respeitar os requisitos não funcionais RNF09 e RNF11, optou-se por usar os princípios da arquitetura “Clean Architecture”. Este estilo arquitetural promove a separação de

¹⁶ Sítio na web: <https://code.visualstudio.com/>

responsabilidades, favorece o baixo acoplamento e alta coesão, permitindo construir aplicações com as seguintes características [33, p. 201]:

- Independente de *frameworks*. A arquitetura não depende de nenhuma *framework*, simplificando a sua substituição.
- Testável. As regras de negócio são facilmente testáveis sem o uso da UI, base de dados ou do servidor *web*.
- Independente da UI. A interface do utilizador pode alterar sem afetar o sistema.
- Independente da base de dados. O sistema de base de dados pode ser alterado sem afetar o restante sistema. As regras de negócio não estão acopladas à base de dados.
- Independente de sistemas externos. As regras de negócio também não estão acopladas às interfaces com sistemas externos.

Neste estilo arquitetural, a lógica de negócio e o modelo aplicacional são o centro da aplicação (*Application Core*). Assim, a lógica de negócio não depende nem da lógica de acesso a dados, nem da lógica de acesso a sistemas externos, ficando o sentido de dependências invertido: a lógica da UI, de acesso a dados e de sistemas externos depende do *core* da aplicação [34, p. 28]. A inversão do sentido da dependência é possível definindo abstrações, ou interfaces, no *Application Core*, que são depois implementadas por tipos definidos nas outras camadas.

O diagrama da Figura 13 mostra um exemplo deste estilo arquitetural, onde podemos verificar a existência de três grandes camadas:

- *User Interface*, onde se encontra implementada a lógica de apresentação.
- *Application Core*, onde se encontram implementados a lógica de negócio e os modelos aplicacionais.
- *Infrastructure*, onde se encontra implementada a lógica de acesso a dados e a sistemas externos.

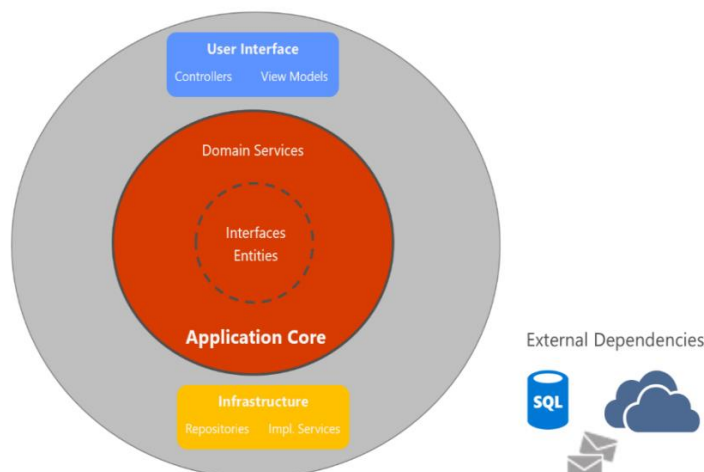


Figura 13 – Vista (conceptual) de camadas do estilo arquitetural “Clean Architecture” [34]

No diagrama da Figura 13 as dependências circulam para o centro do círculo. É possível verificar que os modelos aplicativos e a lógica de negócio encontram-se no centro, sem se verificar a existência de qualquer tipo de dependência.

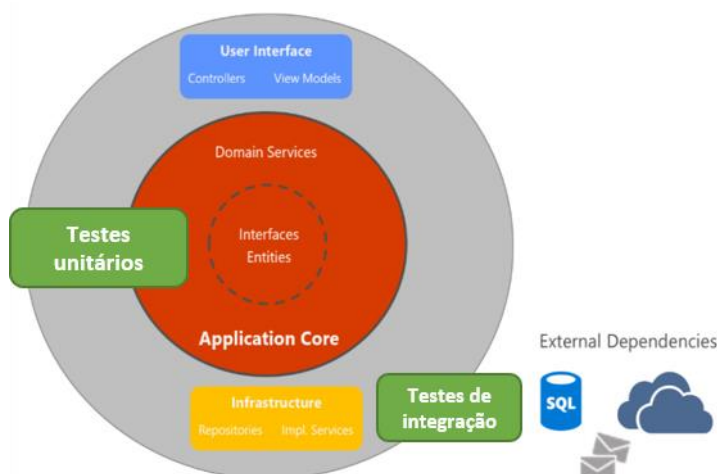


Figura 14 – Inserção de testes unitários e de integração na arquitetura [34]

A Figura 14 representa o “encaixe” dos testes unitários e de integração nesta arquitetura. No primeiro caso, os testes unitários simplificam-se visto que a camada *Application Core* não tem qualquer dependência. O mesmo acontece com os testes de integração, já a camada de UI não tem uma dependência direta da camada de infraestrutura.

3.4.2 A aplicação web

Analisando os requisitos não funcionais identificados, verifica-se que o requisito não funcional RNF12 indica que o acesso à aplicação é realizado através de um *browser*. Assim, estamos perante uma aplicação que irá executar num ambiente *web*. A Figura 15 apresenta, através de um diagrama de componentes, a vista de alto nível da aplicação *web* a desenvolver.

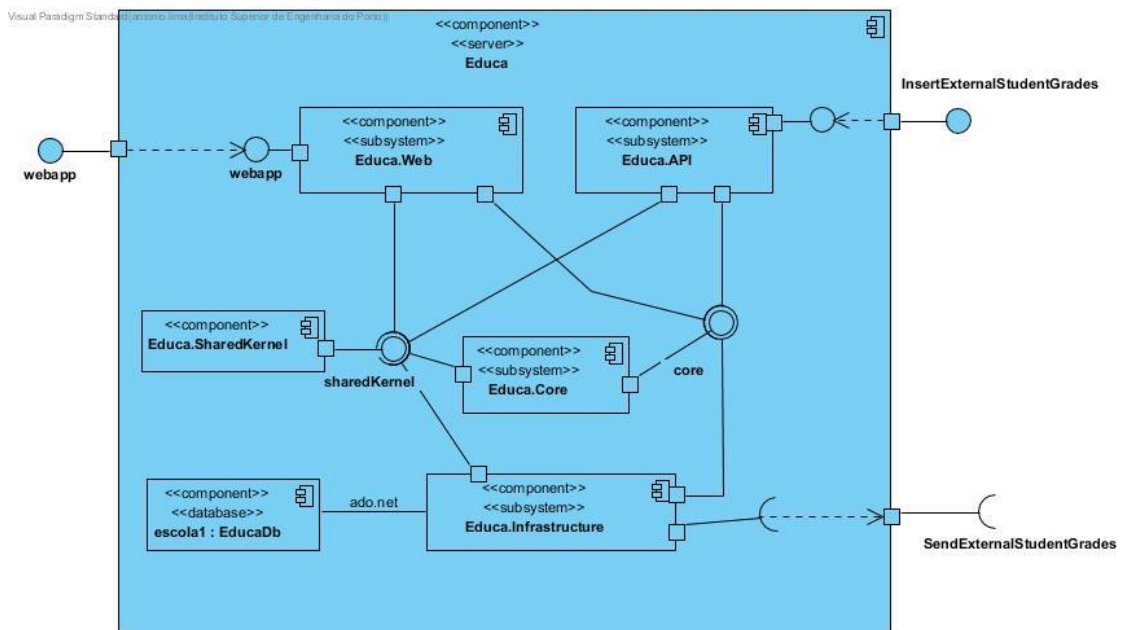


Figura 15 – Vista de alto nível

Analisando o diagrama da Figura 15 verificamos a existência de seis componentes, que são:

- **Educa.Web** – representa a aplicação *web*. É este componente que recebe os pedidos do *browser*, sendo o principal ponto de entrada do sistema. A lógica da UI encontra-se implementada neste ponto.
- **Educa.API** – representa a API da aplicação. É este componente que recebe os pedidos provenientes de outras instâncias (escolas) da aplicação, para integrar dados de alunos que participaram em programas de mobilidade (conforme o requisito não funcional RNF01).
- **Educa.SharedKernel** – componente que contém lógica relacionada com as características transversais da aplicação, tais como a informação sobre o utilizador autenticado (para efeitos de validações de segurança). Este componente não tem dependências de outros componentes.
- **Educa.Core** – componente que contém a lógica de negócio e os modelos aplicacionais. A única dependência deste componente é o “Educa.SharedKernel”. Todos os outros componentes dependem deste.
- **Educa.Db** – representa a base de dados da aplicação.

Visto que estão a ser seguidos os princípios do estilo arquitetural “Clean Architecture”, a Figura 16 mostra onde é que os componentes anteriormente apresentados “encaixam” no diagrama conceptual da Figura 13.

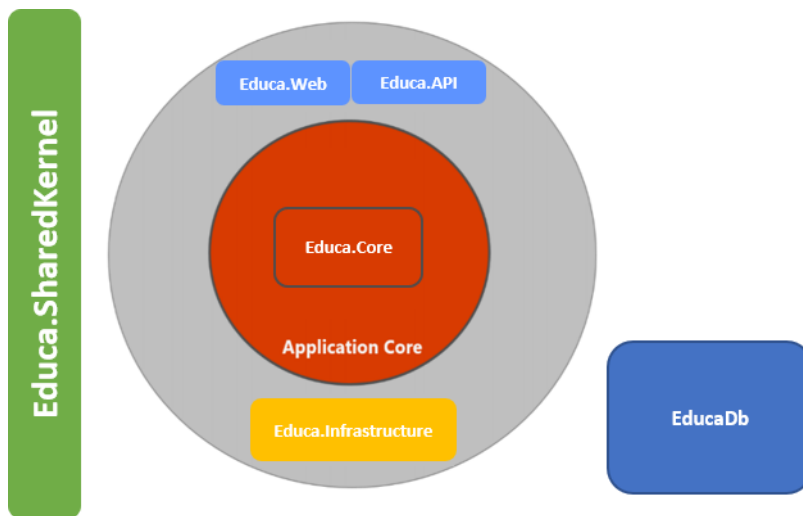


Figura 16 – Componentes da aplicação na arquitetura “Clean Architecture”

3.4.3 Integração de dados entre aplicações

O requisito não funcional RNF01 indica que “o sistema deve permitir que as notas dos estudantes que participam em programas de mobilidade sejam automaticamente integradas nas suas escolas de origem”. Existem várias formas de se cumprir este requisito, sendo as várias alternativas apresentadas em seguida. Para cada uma das alternativas usaram-se, como exemplo, duas instâncias da aplicação.

3.4.3.1 Alternativa 1

O diagrama da Figura 17 apresenta a primeira alternativa possível para respeitar o requisito da integração.

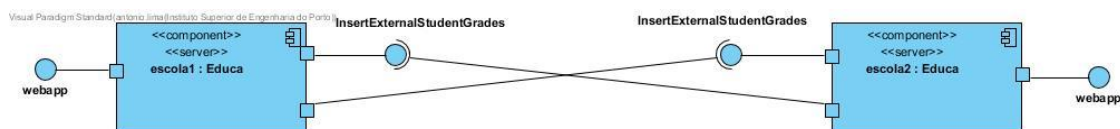


Figura 17 – Integração entre instâncias (alternativa 1)

O diagrama de sequência da Figura 18 mostra um exemplo da integração de dados, seguindo esta arquitetura.

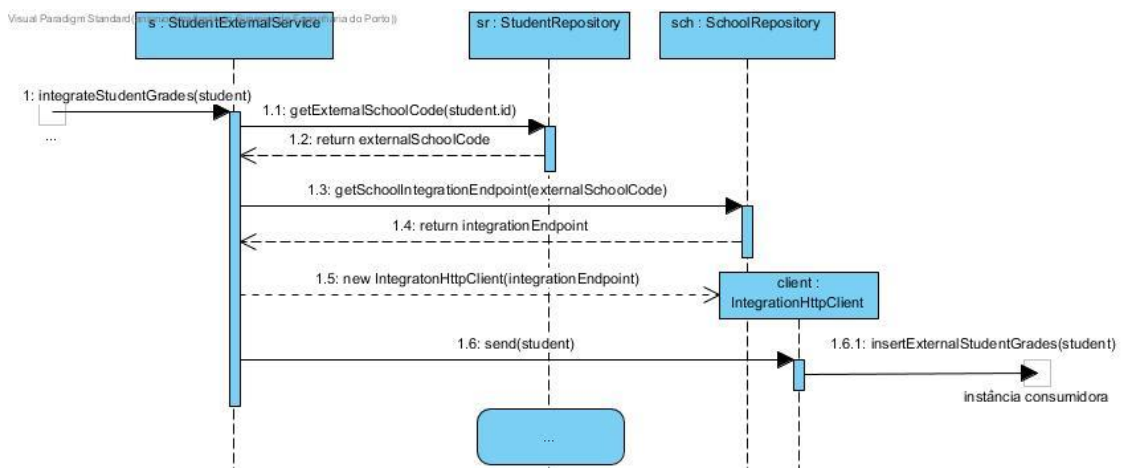


Figura 18 – Diagrama de sequência da alternativa 1 de integração

Nesta alternativa, existe uma conexão direta entre todas as instâncias (escolas) existentes, o que significa que cada instância necessita de conhecer todas as instâncias em execução, por forma a tornar possível a comunicação. Sempre que for adicionada uma nova instância ao ecossistema de aplicações “Educa”, todas as instâncias já existentes vão ter de ser alteradas para acrescentar a nova aplicação.

Ora, num projeto piloto onde, no máximo, existirão três instâncias em execução, este cenário não é problemático e é perfeitamente gerível. No entanto, este viola os requisitos não funcionais RNF11 e RNF13, já que a extensibilidade e a manutenibilidade do sistema são claramente afetadas seguindo esta solução. Assim, esta alternativa é descartada.

3.4.3.2 Alternativa 2

O diagrama da Figura 19 apresenta a segunda alternativa possível para respeitar o requisito da integração.

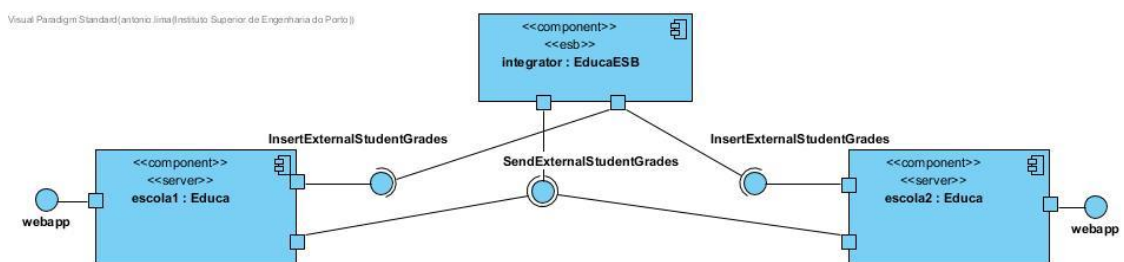


Figura 19 – Integração entre instâncias (alternativa 2)

Analisando o diagrama da Figura 19 verifica-se que as instâncias em execução são ignorantes quanto à existência de outras instâncias. Essa responsabilidade está agora presente no componente “EducaESB” que, tal como o nome indica, trata-se de um Enterprise Service Bus (ESB). Um ESB funciona como um *middleware* que permite que diversas aplicações, serviços e sistemas comuniquem entre si através de um barramento de comunicação comum, independentemente dos tipos de dados ou protocolos [35].

O diagrama de sequência da Figura 20 mostra um exemplo da integração de dados, seguindo esta arquitetura.

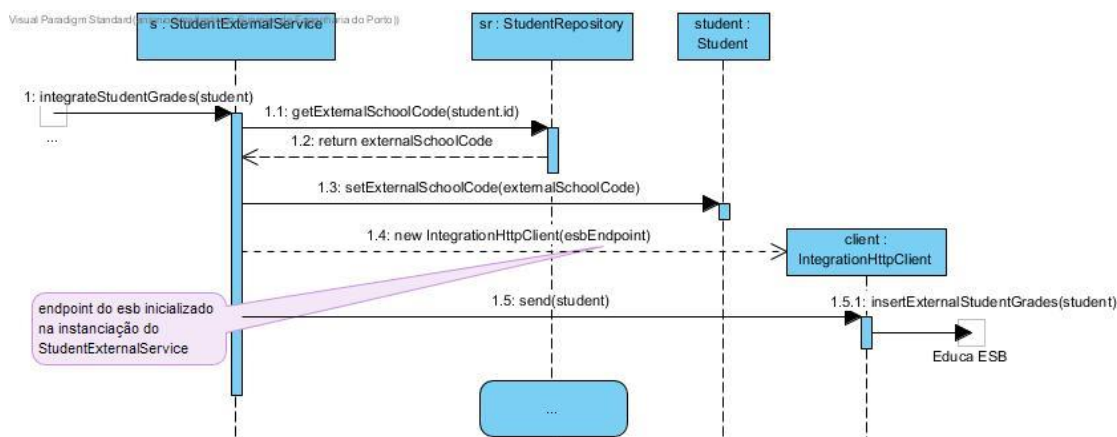


Figura 20 - Diagrama de sequência da alternativa 2 de integração

Comparando este diagrama de sequência com o diagrama de sequência da alternativa 1, verifica-se que a lógica ficou mais simples e que a responsabilidade de determinar o *endpoint* da instância de destino foi eliminada, passando agora para o ESB. Neste cenário, sempre que for necessário acrescentar uma nova instância, só é necessário alterar o ESB, pois só este necessita de conhecer os novos *endpoints*.

No entanto, prevendo que no futuro várias escolas começam a aderir à plataforma “Educa”, esta abordagem torna-se também pouco flexível, já que o ESB terá de ser alterado sempre que for necessário parametrizar uma nova instância. Mais uma vez, e apenas neste cenário, os requisitos não funcionais RNF11 e RNF13 são violados. Assim, esta alternativa é descartada.

3.4.3.3 Alternativa 3

O diagrama da Figura 21 apresenta a segunda alternativa possível para respeitar o requisito da integração.

Esta nova alternativa acrescenta um novo componente ao sistema, o “EducaAdmin”. Este é constituído por uma API, “EducaAdmin.API”, e uma base de dados, “EducaAdminDB”. A responsabilidade desta API é fornecer informação sobre as escolas (*endpoints* e escalas de avaliação, nesta versão), sendo esta informação parametrizada através de uma UI desenvolvida para esse propósito. O utilizador responsável por essa parametrização é o Administrador (ver Tabela 4 – Atores do sistema).

Com este novo componente, sempre que for criada uma instância da aplicação “Educa”, deixa de ser necessário alterar o ESB. Agora, basta adicionar a informação do *endpoint* e da escala de avaliação na base de dados “EducaAdminDB”, ficando imediatamente disponível para consulta por parte do ESB.

As escalas de avaliação também são fornecidas pois, caso a troca de dados seja entre duas escolas que usam escalas diferentes, o ESB automaticamente converte as notas do aluno para a escala da escola de origem (do aluno).

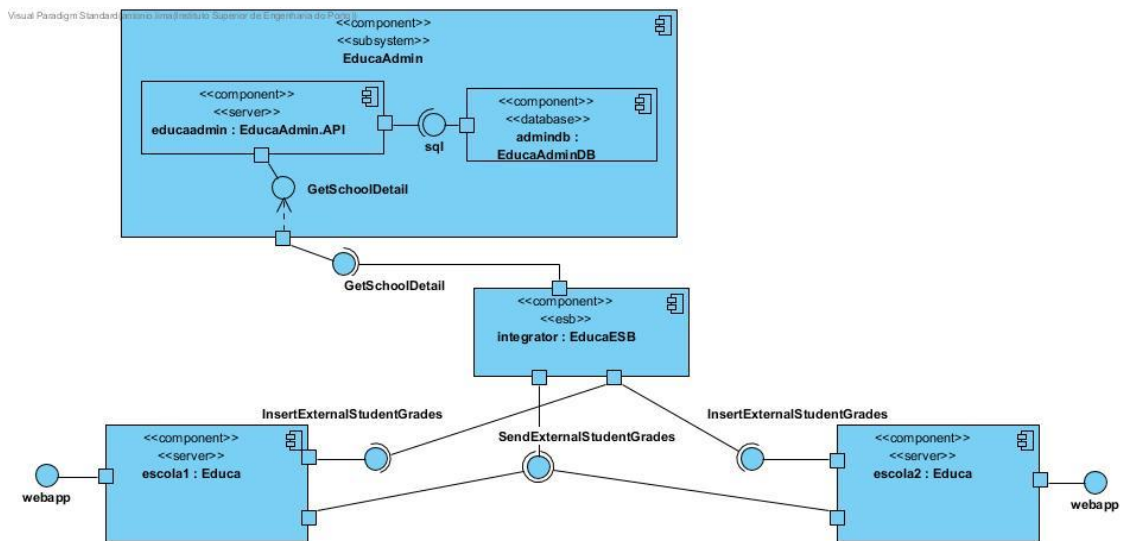


Figura 21 – Integração entre instâncias (alternativa 3)

O diagrama de seqüência da Figura 22 mostra um exemplo da integração de dados, seguindo esta arquitetura.

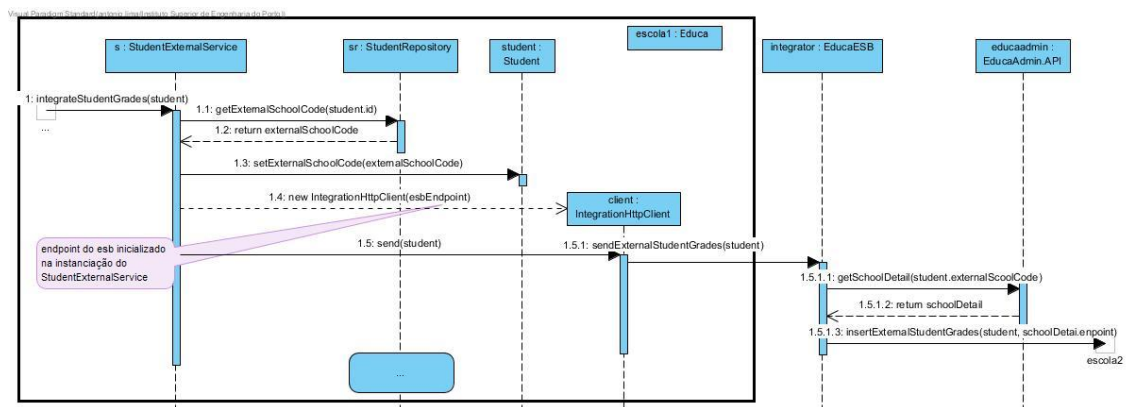


Figura 22 - Diagrama de seqüência da alternativa 3 de integração

3.4.3.4 Conclusão da análise de alternativas de integração

Analisando as várias alternativas conclui-se que todas são possíveis, mas apenas a alternativa 3 respeita os requisitos não funcionais RNF11 e RNF13, requisitos esses que são bastante significativos arquiteturalmente.

A Figura 23 é uma atualização da Figura 16, onde agora pode-se verificar onde “encaixa” o ESB na arquitetura existente. Analisando a figura, o ESB passa a ser mais uma dependência externa, sendo o componente “Educa.Infrastructure” responsável pela comunicação (produtor) ou o componente “Educa.API” (consumidor).

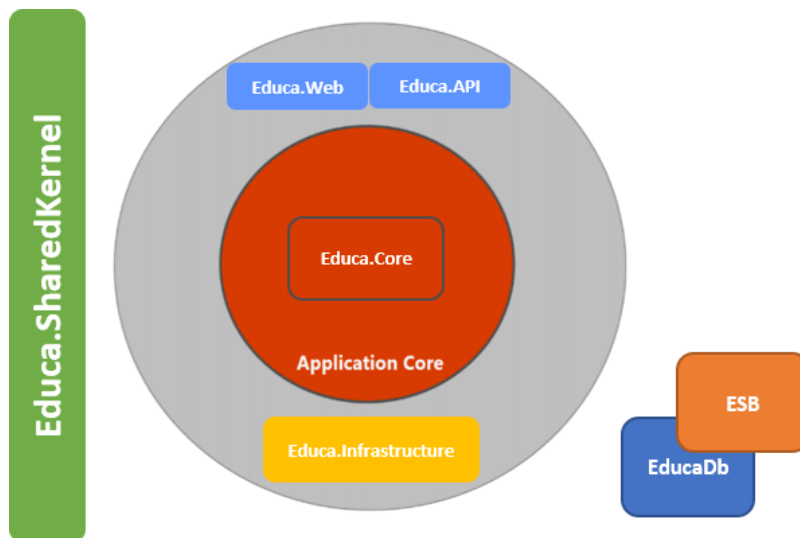


Figura 23 – Introdução do ESB na arquitetura

3.4.4 Geração de relatório e certificados

O requisito não funcional RNF08 indica que “os relatórios e os certificados têm de ser gerados, no máximo, em 3 segundos”. Para se perceber onde poderão estar os problemas de performance nestes dois casos de uso, desenvolveu-se o diagrama de seqüência da Figura 24 para analisar o fluxo de mensagens do caso de uso “UC20. Gerar certificado de aquisição de competências”.

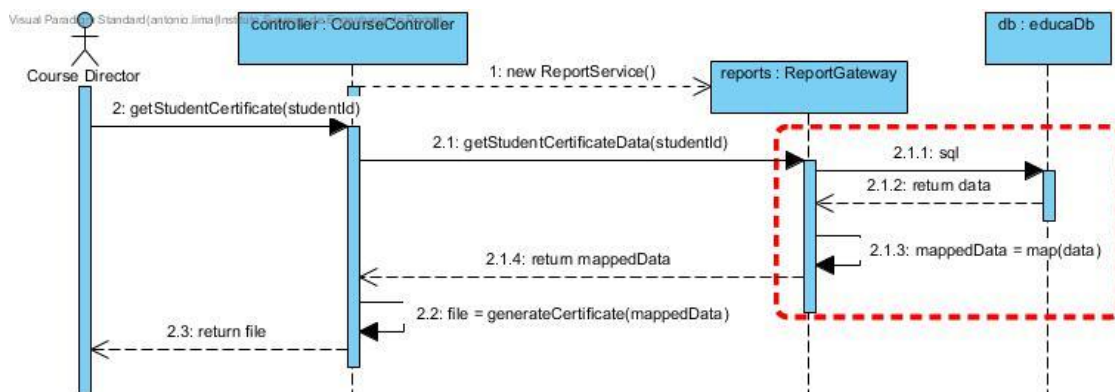


Figura 24 – Possível ponto crítico na geração de certificados

Analisando o diagrama da figura verifica-se que este caso de uso apenas consulta dados, e nunca altera o estado da aplicação. A caixa vermelha indica o ponto onde poderão existir problemas de performance. Esse ponto encontra-se na camada de acesso a dados e na própria base de dados. Assim, pode-se estar perante dois tipos de problemas:

1. Ineficiência das *queries* realizadas à base de dados;
2. Problemas nos mapeamentos entre a base de dados e a camada aplicacional.

Os dois problemas mencionados revelam que a causa de um possível problema de performance poderá ser o ORM. Na subsecção 0, referiu-se que o ORM selecionado seria o Entity Framework Core. No entanto, neste cenário, talvez este tipo de ORM não seja mais o adequado.

Uma alternativa ao Entity Framework Core poderá ser um micro-ORM, como o Dapper. Um micro-ORM é nada mais do que uma versão *light* de um ORM tradicional. Este tipo de ORMs fornecem as mesmas capacidades de mapeamento que um ORM tradicional, mas encontram-se fortemente otimizados para *queries* de consulta. Geralmente, quando se trabalha com um micro-ORM como o Dapper, ou se constrói a *query* de SQL diretamente no código, ou invoca-se um procedimento ou uma função de SQL.

A Tabela 13 mostra os resultados de execução das mesmas *queries*, perante os mesmos cenários, usando Entity Framework Core (na tabela abreviado para EF Core) e Dapper. Na tabela pode-se observar o conceito “EF Core Raw SQL”, que significa a capacidade que o Entity Framework Core tem de enviar SQL direto para a base de dados (semelhante ao Dapper).

O modelo usado nos testes trata-se de uma classe designada “Designers”, que possui relações com três entidades: “Produtos”, “Clientes” e “Contacto”. Todos os resultados estão em milissegundos.

Tabela 13 – Comparação entre Entity Framework Core e Dapper [36]

Query	Relação	EF Core	EF Core Raw SQL	Dapper Raw SQL
Todos os Designers (30 mil registos)	-	96	98	77
Todos os Designers com Produtos (30 mil registos)	1: *	251	107	91
Todos os Designers com Clientes (30 mil registos)	* : *	255	106	63
Todos os Designers com Contacto (30 mil registos)	1: 1	322	122	116

Analisando os resultados, facilmente se verifica a vantagem do Dapper sobre o Entity Framework Core. No entanto, é importante referir que, apesar da vantagem ser evidente, nem sempre significa que este deve ser usado [36], sendo sempre necessário analisar o contexto em que a necessidade se insere. No caso particular deste projeto, justifica-se o uso do Dapper já que este só será usado para o caso específico da geração de relatórios e de certificados.

Ao fazer uso de dois ORMs distintos, esta aplicação torna-se numa aplicação híbrida. Para se perceber como é que os dois ORMs irão funcionar em conjunto, desenvolveu-se o diagrama da Figura 25, que detalhe o componente “Educa.Infrastructure”, anteriormente apresentado.

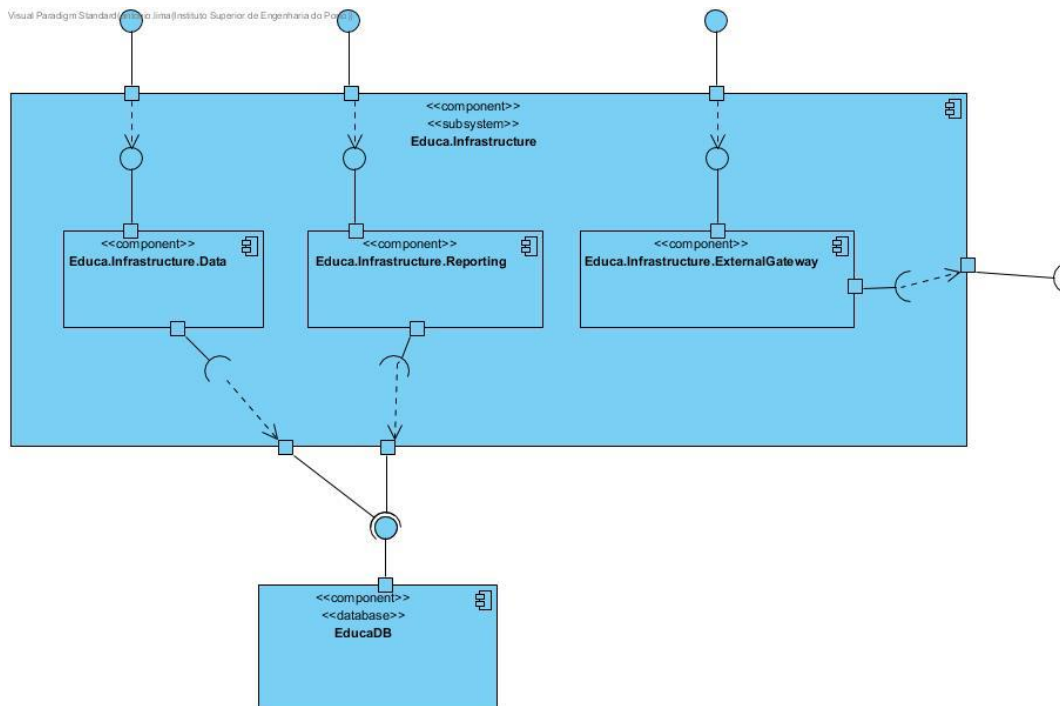


Figura 25 – Vista lógica do componente “Educa.Infrastructure”

Analisando a Figura 25, verifica-se a existência de três componentes, que são:

- **Educa.Infrastructure.Data** – componente responsável pelo acesso a dados para as funcionalidades gerais da aplicação. Este componente utiliza o Entity Framework Core como ORM.
- **Educa.Infrastructure.Reporting** – componente responsável pelo acesso a dados para a geração de relatórios e certificados. Este componente utiliza o Dapper como ORM (micro-ORM).
- **Educa.Infrastructure.ExternalGateway** – componente responsável pela comunicação com serviços externos (ex. ESB).

3.4.5 Vista de dados

A vista de dados é um diagrama que pretende representar o modelo da base de dados. Visto que este diagrama tem alguma dimensão, decidiu-se colocá-lo em anexo (Anexo 1).

3.4.6 Vista de Implantação

A vista de implantação é um diagrama que apresenta os nós em execução e os componentes neles inseridos. O diagrama da Figura 26 apresenta a vista de implantação da aplicação “Educa”.

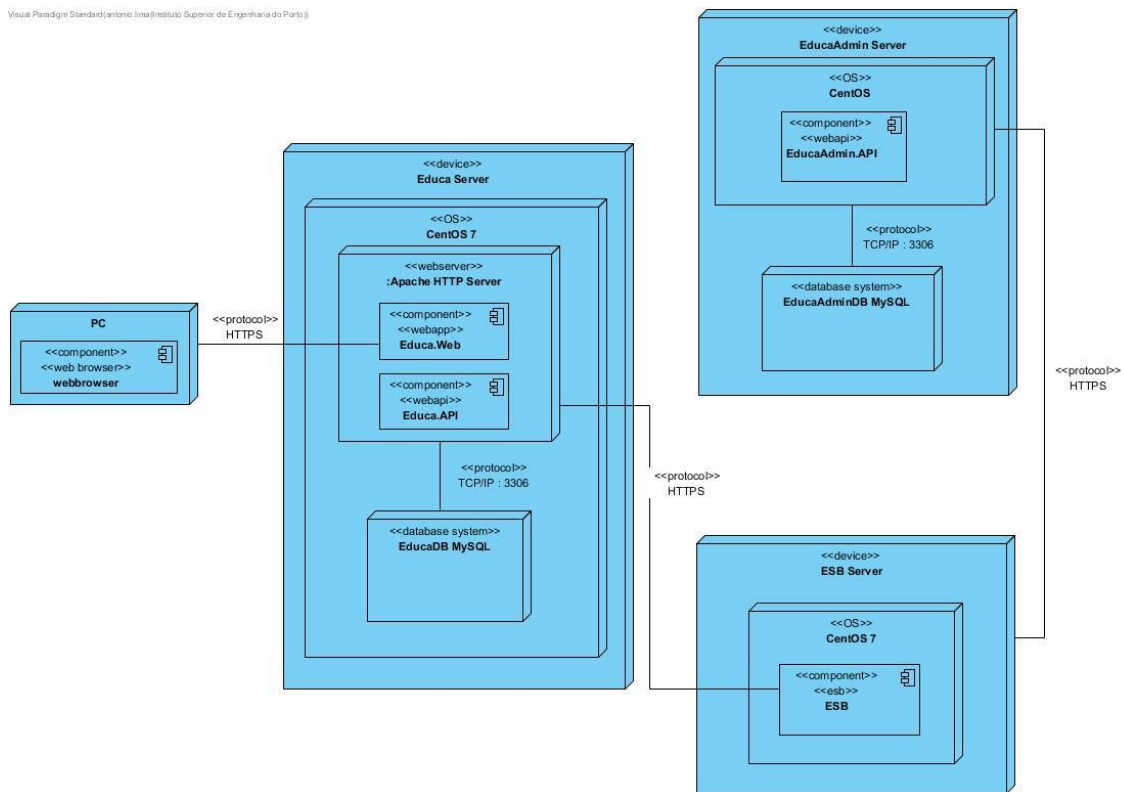


Figura 26 – Vista de implantação

Analisando o diagrama, verificamos a existência dos seguintes nós:

- **Educa Server** – representa a instalação de uma instância da aplicação “Educa” numa escola. A aplicação web “Educa.Web” e a API “Educa.API” executam numa máquina Linux, com o sistema operativo CentOS 7, num servidor Apache. Estas comunicam com uma base de dados MySql, através da porta 3306. Os utilizadores acedem à aplicação web utilizando um *browser* (webbrowser).
- **ESB Server** – representa o nó onde executa o ESB. Este está instalado numa máquina Linux.
- **EducaAdmin Server** – representa o nó onde executa a API que suporta o ESB. Tal como o Educa Server, executa numa máquina Linux, com o sistema operativo CentOS 7, num servidor Apache. A API comunica com uma base de dados MySQL, através da porta 3306.

3.4.7 Continuous Integration (CI)

Continuous Integration é uma boa prática de DevOps, onde os *developers* de uma aplicação integram frequentemente o código num sistema de controlo de versões (por exemplo, uma vez ao dia) [37]. Para além da constante integração do código, também as *builds* são automatizadas, assim como a execução dos testes, e os *deploys* para ambientes de pré-produção podem ser feitos nesta fase. Esta prática expõe um conjunto de vantagens, tais como:

- A última versão do código está sempre disponível;

- Ciclos de compilação são mais rápidos, já que os problemas são detetados mais cedo e com mais frequência;
- Facilidade em reverter versões do *software*, já que os *deploys* são automáticos.

Embora este projeto tenha sido desenvolvido apenas por uma pessoa, esta boa prática foi seguida. A Figura 27 mostra o pipeline de CI desenvolvido.

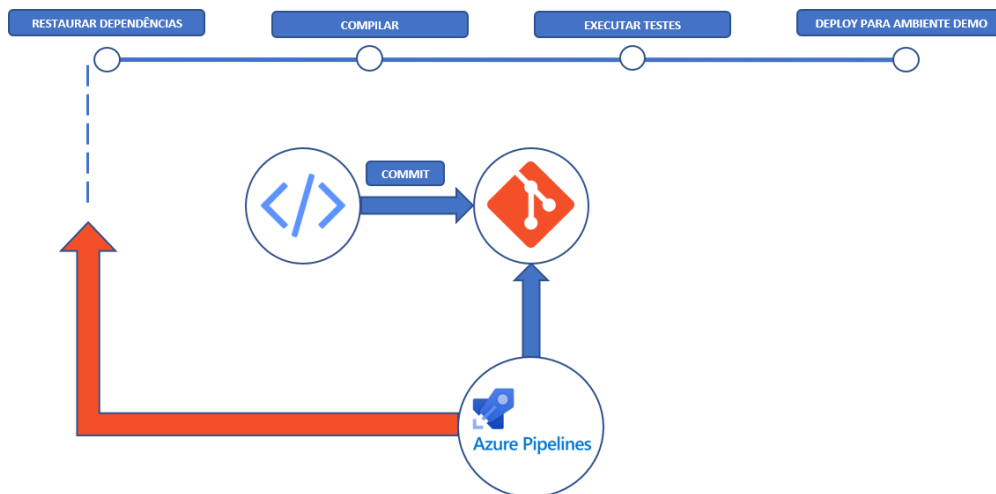


Figura 27 – Pipeline de Continuous Integration (CI)

O pipeline implementado está a executar na *cloud*, num ambiente Azure. O sistema de controlo de versões utilizado é GIT.

Analisando o pipeline verifica-se que existem quatro *steps* sequenciais, que são:

1. Restaurar dependências – neste passo são restauradas todas as dependências da aplicação;
2. Compilar – a aplicação é compilada;
3. Executar testes – são executados os testes unitários e de integração;
4. Deploy para ambiente demo – foi criado no Azure um ambiente de demonstração da aplicação. O último passo do pipeline é a *release* automática para esse ambiente.

4 Implementação

“Programming can be fun, so can cryptography; however they should not be combined.”

Kreitzberg and Shneiderman

Neste capítulo são apresentados os detalhes de desenvolvimento da solução anteriormente apresentada, focando-se essencialmente no detalhe da aplicação *web* e do ESB. Os diagramas apresentados neste capítulo têm uma granularidade mais fina, visto estarem próximos da implementação efetuada.

4.1 Implementação da aplicação *web*

Na subsecção 3.4.1 apresentou o estilo arquitetural “Clean Architecture” como base a seguir para a implementação deste projeto. Revendo a Figura 13, verifica-se que este estilo arquitetural é composto por três grandes camadas: *web*, *core* e *infraestrutura*. A Figura 28 apresenta a estrutura de pastas vista na perspectiva do IDE utilizado para o desenvolvimento, o Visual Studio 2019.

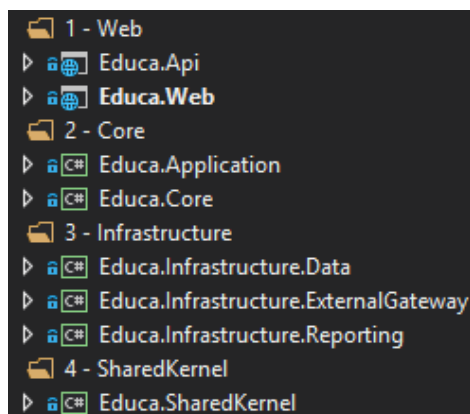


Figura 28 – Organização dos projetos

As subsecções seguintes apresentam o detalhe de cada uma das camadas da arquitetura da aplicação.

4.1.1 Camada “SharedKernel”

A camada “SharedKernel” não é apresentada no estilo arquitetural “Clean Architecture”. Esta foi implementada no projeto para se seguir um conjunto de boas práticas de reutilização. Decidiu-se iniciar a apresentação das camadas por este ponto já que todos os projetos da solução desenvolvida dependem desta camada (ver Figura 15 e Figura 16). O diagrama da Figura 29 apresenta a vista de módulos da camada.

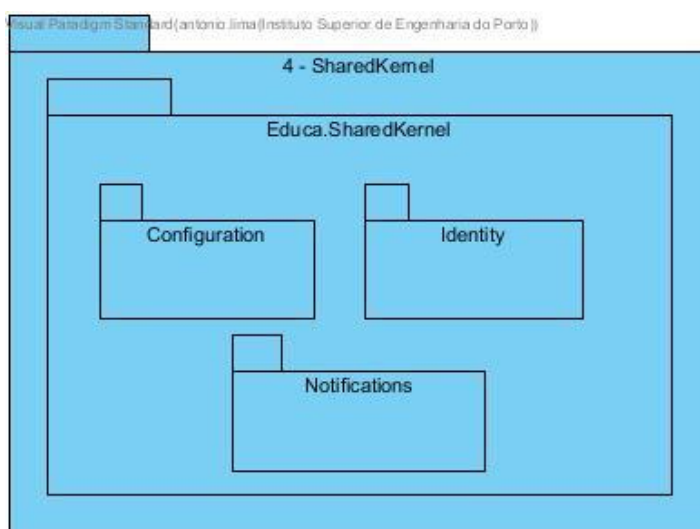


Figura 29 – Vista de módulos da camada “SharedKernel”

O diagrama apresenta três módulos, que são:

- Configuration: A *framework* ASP.NET Core fornece um ficheiro¹⁷ onde podem ser definidas variáveis de configuração, que podem ser alteradas em *runtime*. As classes definidas neste *package* contêm o valor dessas variáveis, que podem depois ser usadas nas restantes camadas da aplicação.
- Identity: Contém os nomes dos *roles* que a aplicação suporta, as classes que representam os *roles* da aplicação e a classe que representa o utilizador com a sessão ativa.
- Notifications: Contém os códigos dos erros que a aplicação pode lançar.

4.1.2 Camada Core

A camada Core é a camada onde se encontram implementados os modelos aplicacionais e a lógica de negócio. O diagrama da Figura 30 apresenta a vista de módulos da camada.

¹⁷ Na *framework* ASP.NET Core este ficheiro tem é o *appsettings.json*

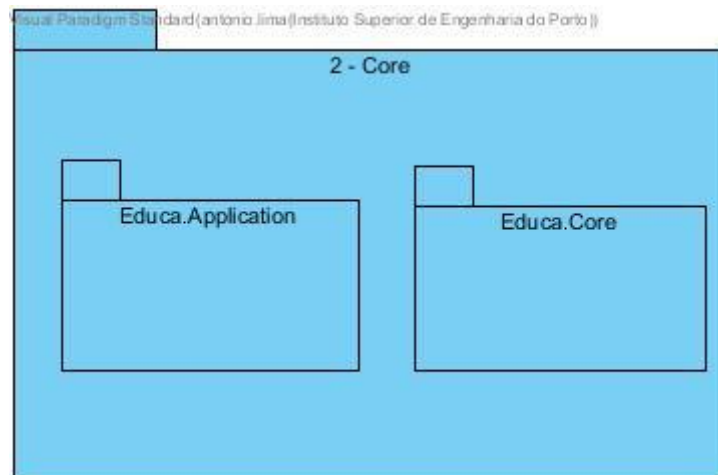


Figura 30 – Vista de módulos da camada Core

Analisando a figura apresentada verifica-se a existência de dois módulos, que são:

- **Educa.Application** – Contém os serviços aplicativos consumidos pela camada de apresentação;
- **Educa.Core** – Contém as modelos aplicativos, os serviços de domínio, os DTOs (*Data transfer objects*) e interfaces.

Dada a relevância arquitetural do módulo “Educa.Core”, apresenta-se em seguida o seu detalhe.

4.1.2.1 Módulo “Educa.Core”

Este módulo trata-se do centro da aplicação e não deve ter nenhuma dependência para além da própria *framework* em que está inserido. O diagrama da Figura 31 apresenta a vista de módulos da camada.

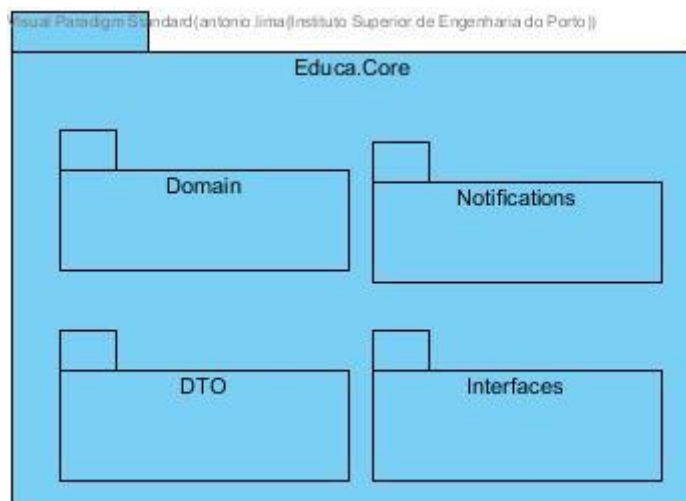


Figura 31 – Vista de módulos “Educa.Core”

O diagrama da Figura 31 apresenta os seguintes módulos:

- Domain: Contém a definição de todos os modelos aplicativos, dos serviços de domínio e de sistemas de resolução de dependências (internos à aplicação) úteis para o domínio. Foi seguida uma abordagem de desenvolvimento orientado ao domínio, pelo que a lógica de negócio está encapsulada nas entidades de domínio;
- DTO: Contém a definição dos DTOs utilizados na aplicação;
- Interfaces: Contém todas a definição de todas as interfaces para serviços externos ou de acesso a dados. Estas interfaces incluem abstrações para operações realizadas pela camada de infraestrutura, tais como o acesso a dados ou a serviços externos;
- Notifications: Contém um sistema de geração de notificações implementado para enviar mensagens de erros para a camada de apresentação. Será detalhado posteriormente.

4.1.3 Camada de Infraestrutura

Na camada de infraestrutura está implementada a lógica de acesso a dados ou a serviços externos. O diagrama da Figura 32 apresenta a vista de módulos da camada.

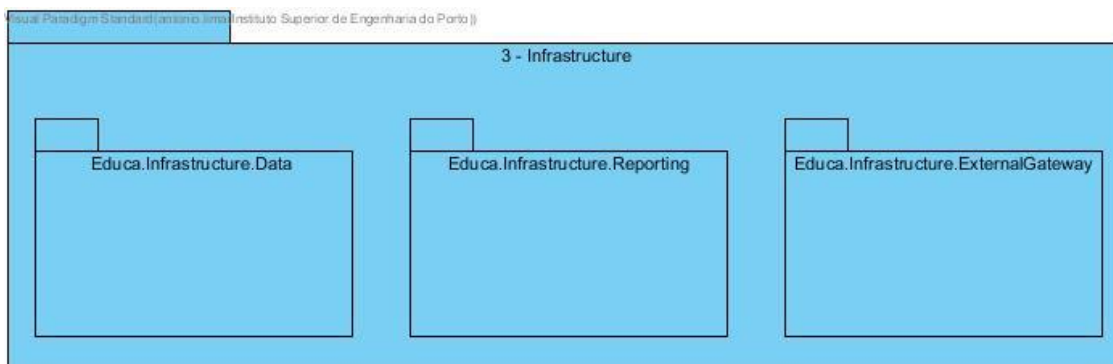


Figura 32 – Vista de módulos da camada de Infraestrutura

O diagrama da Figura 32 apresenta os seguintes módulos:

- Educa.Infrascuture.Data – Contém a lógica de acesso a dados. Relembrando o capítulo da análise, é este módulo que utiliza o Entity Framework Core como ORM.
- Educa.Infrastructure.Reporting – Contém a lógica de acesso a dados para geração de relatórios. Mais uma vez, lembrando o capítulo da análise, é este o módulo que utiliza o micro-ORM Dapper.
- Educa.Infrastructure.ExternalGateway – Contém a lógica de acesso a serviços externos, tais como o ESB ou serviços de envio de emails.

4.1.4 Camada de apresentação

A camada de apresentação é o ponto de entrada da aplicação. O diagrama da Figura 33 apresenta a vista de módulos da camada.

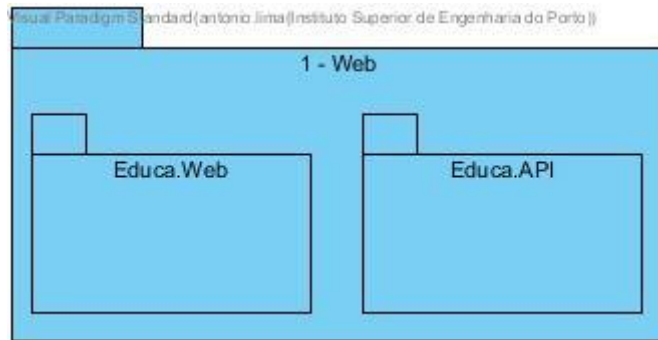


Figura 33 – Vista de módulos da camada de apresentação

O diagrama da Figura 33 apresenta dois módulos, que são:

- Educa.Web – Representa a aplicação web utilizada pelos utilizadores. Trata-se de um projeto ASP.NET Core MVC (*Model-View-Controller*), que, tal como o nome indica, segue o padrão arquitetural MCV.
- Educa.API – Representa a API implementada que recebe pedidos do ESB. Trata-se de um projeto ASP.NET Core API.

Dada a relevância arquitetural do módulo “Educa.Web”, apresenta-se em seguida o seu detalhe.

4.1.4.1 Módulo “Educa.Web”

O diagrama da Figura 34 apresenta a vista de módulos do módulo “Educa.Web”.

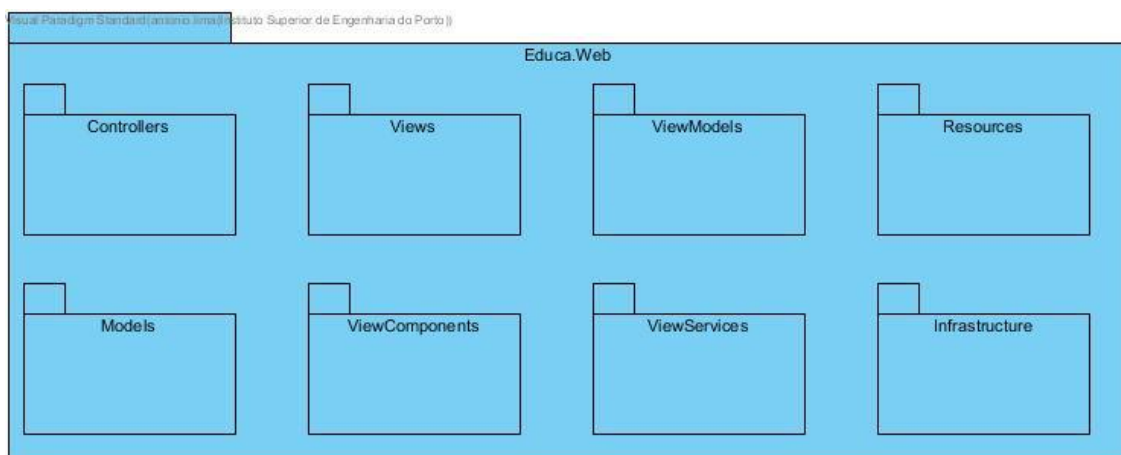


Figura 34 – Vista de módulos “Educa.Web”

Sendo o módulo da Figura 34 um projeto que segue o padrão arquitetural MVC, é de esperar encontrar referências a “Controllers”, “Views” e “ViewModels”. São ainda apresentados os seguintes módulos:

- ViewComponents – Contém *views* que são reutilizáveis em várias partes da aplicação.
- Resources – Contém os ficheiros com as traduções para as várias línguas a suportar. A aplicação foi desenvolvida com a linguagem inglesa como base. Para suportar outras línguas basta carregar os ficheiros com as diferentes traduções.
- ViewServices – Contém classes cuja sua responsabilidade é o mapeamento entre DTOs e View Models, e vice-versa. Assim, toda a lógica de mapeamentos é abstraída dos *controllers*, o que permite tornar o código mais limpo.
- Infrastructure – Contém a lógica para tratar os ficheiros carregados para aplicação (p.e, o ficheiro de alunos carregado pelo Administrador no momento de criação de um curso). É este módulo que tem a responsabilidade de mapear o conteúdo do ficheiro para um modelo (não um modelo de negócio!).

4.1.5 Evidências de boas práticas e do uso de padrões de *software*

O desenvolvimento da aplicação teve sempre em mente o uso de boas práticas e de padrões de *design* de *software*. Em seguida apresentam-se algumas dessas evidências.

4.1.5.1 Padrão Repositório + Unidade de trabalho

Os padrões repositório e unidade de trabalho são dois padrões que costumam ser usados em conjunto. O uso destes padrões permite criar uma camada de abstração entre a camada de acesso a dados e a camada da lógica de negócio, o que aumenta a testabilidade do código e a favorece o baixo acoplamento. Qualquer alteração na camada de acesso a dados não se reflete na camada da lógica de negócio.

Este padrão foi aplicado no módulo “Educa.Infrastructure.Data”. O diagrama de classes da Figura 35 apresenta um exemplo da sua implementação.

No diagrama de classes é possível verificar a existência da classe “EducaDbContext”. Esta classe é específica do Entity Framework Core e representa uma sessão com a base de dados que pode ser utilizada para realizar consultas ou alterações.

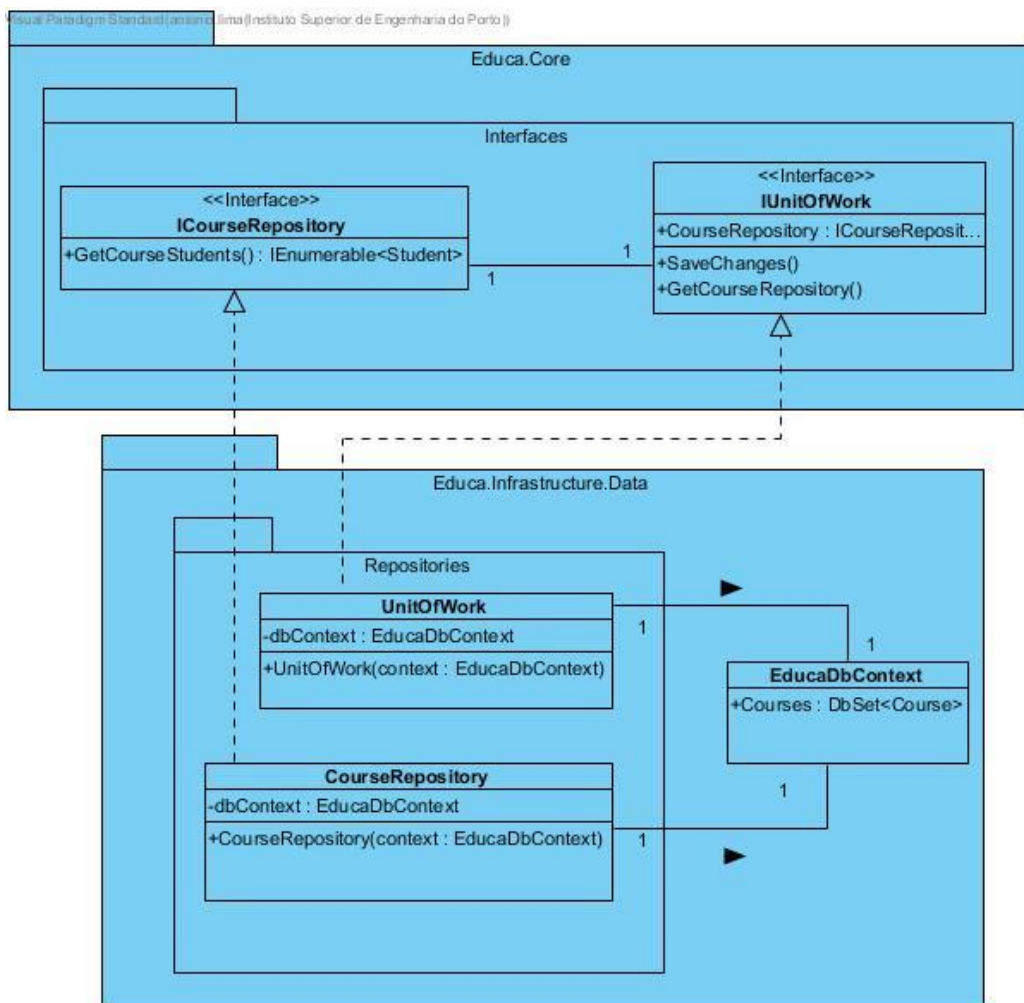


Figura 35 – Implementação do padrão repositório + unidade de trabalho

Na literatura referente ao assunto, existem vários autores que são contra a implementação destes padrões em conjunto com o uso do ORM Entity Framework. Segundo John Smith [31, p. 289], a própria classe “DbContext” (“EducaDbContext” no diagrama) do Entity Framework já aplica o padrão unidade de trabalho, e os DbSets que esta define têm a mesma funcionalidade que o padrão repositório. A análise realizada pelo autor é verdadeira, já que a própria Microsoft também o afirma [38].

No entanto, perante a arquitetura implementada, caso estes padrões não sejam utilizados, a camada Core irá depender diretamente do ORM Entity Framework, ficando esta camada exposta aos detalhes de implementação da camada de acesso a dados.

Ao se optar pela implementação destes dois padrões, está-se a respeitar os requisitos não funcionais RNF11 e RNF13 já que, caso se decida alterar a persistência, ou o próprio ORM, o impacto nas restantes camadas da aplicação é minimizado.

4.1.5.2 Padrão Estratégia

A padrão estratégia é um padrão comportamental e o seu objetivo é definir um conjunto de algoritmos, encapsulá-los e torná-los intercambiáveis. Os algoritmos definidos podem ser trocados em tempo de execução, dependendo do cliente que os está a usar [39, p. 315].

Este padrão foi utilizado para respeitar o requisito não funcional RNF13 e permitir que a introdução de novas escalas de avaliação na aplicação seja simples. A versão atual da aplicação suporta as escalas definidas na Tabela 5.

O diagrama de classes da Figura 36 apresenta a implementação deste padrão.

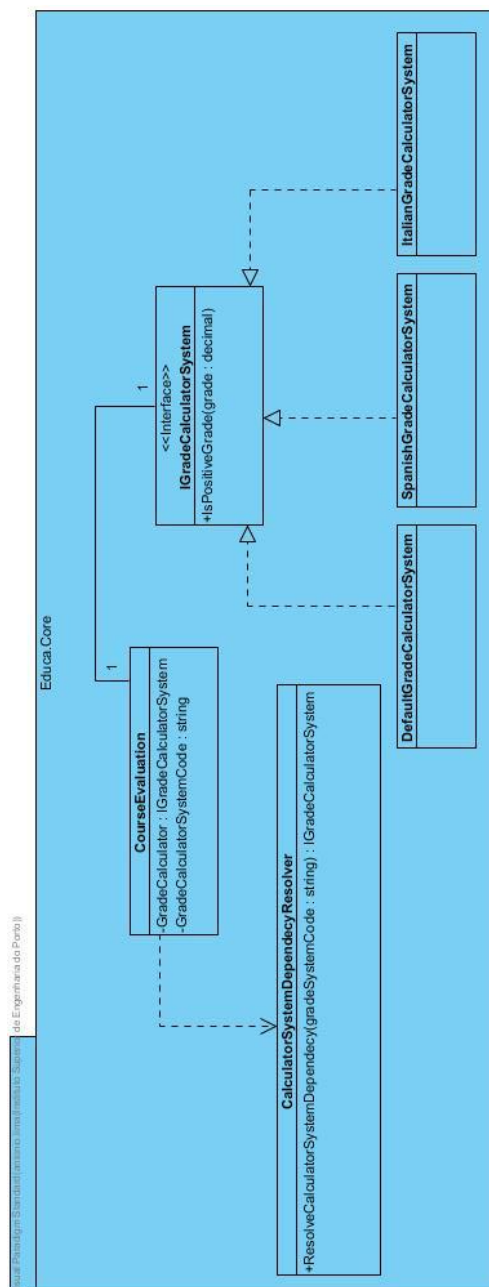


Figura 36 – Implementação do padrão “estratégia”

Caso seja necessário implementar um novo sistema de escalas, basta desenvolver um novo algoritmo que implemente a interface “IGradeCalculatorSystem”.

4.1.5.3 Padrão Notificação

Um cenário comum de uma aplicação é a interface do utilizador capturar algum tipo de informação/comando e enviar essa informação para a camada da lógica de negócio, onde irão ocorrer validações. O domínio realiza as validações e, caso alguma falhe, a camada de apresentação tem de ser notificada [40]. No entanto, respeitando a arquitetura definida para a implementação da solução, a camada *core* (onde se encontra a lógica de negócio, encapsulada nas entidades de domínio) não pode comunicar diretamente com a camada de apresentação. Então, surge a necessidade de implementar o padrão Notificação.

O diagrama de classes da Figura 37 representa a estrutura desta implementação.

Analisando o diagrama de classes apresentado verifica-se a existência de um conjunto de classes responsáveis pelo mecanismo de notificações. No *package* “Notifications” existe a classe “Notification”, que representa a estrutura de uma notificação. Esta tem um tipo, “NotificationType”. A classe “Course”, uma entidade do domínio, tem a responsabilidade de se validar a si própria. Caso exista algum erro, as notificações são criadas e registadas numa lista de notificações herdada da classe base “Entity”.

A classe “CourseService” é um serviço aplicacional responsável por receber e tratar os pedidos recebidos pela camada de apresentação. É o serviço que tem a responsabilidade de verificar se a entidade com que está a lidar criou alguma notificação. Caso tenha criado, essas notificações são adicionadas à classe “NotificationContext”. A classe “NotificationContext” é um repositório de notificações, cujo seu tempo de “vida” é igual à duração do pedido realizado ao servidor. Ou seja, assim que o pedido termine, todas as notificações perdem-se.

Posteriormente, e já na camada de apresentação, as notificações são tratadas por um filtro. Este filtro executa imediatamente antes da resposta do servidor ao *browser* ser efetuada. A responsabilidade deste filtro é analisar o repositório de notificações “NotificationContext” e caso existam notificações, adicioná-las à página a exibir ao utilizador.

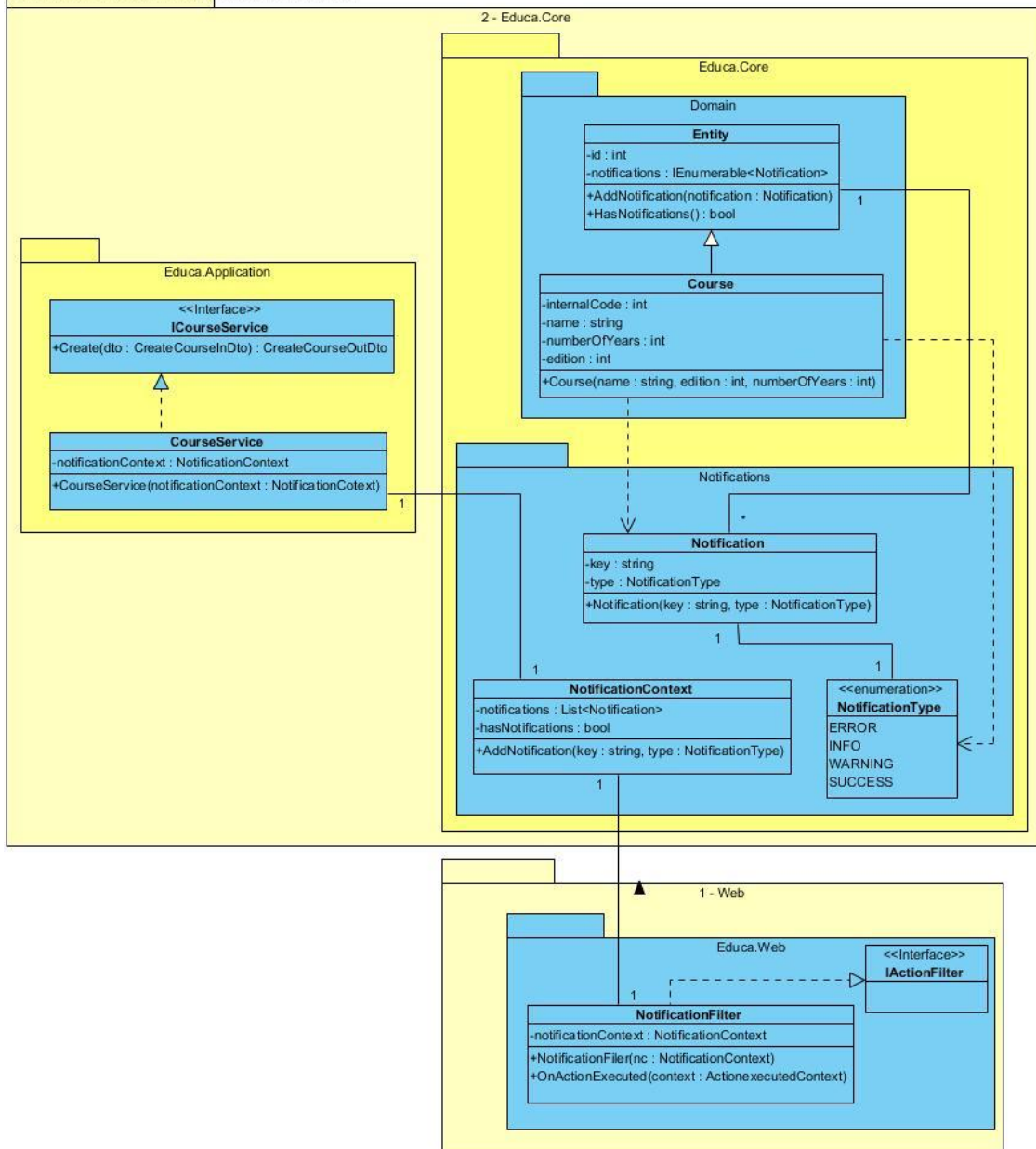


Figura 37 – Diagrama de classes do padrão Notificação

4.2 Integração de sistemas

Nesta secção apresentam-se as principais decisões tomadas para a implementação do ESB. Inicialmente apresenta-se o ESB selecionado para a implementação e, posteriormente, a estrutura do *pipeline*.

4.2.1 Capacidades de um ESB

Um ESB é uma ferramenta poderosa, essencial na integração de sistemas heterogéneos (sistemas que usam protocolos de comunicação distintos, formatos de mensagens distintos, etc.), o que permite a criação de novas soluções de negócio.

As principais características de um ESB são [35]:

- Mediação de mensagens: Manipulação do conteúdo das mensagens, da sua direção e destino, e protocolos.
- Virtualização de serviços: Capacitar serviços ou sistemas existentes com novas interfaces.
- Conversão de protocolos: Por exemplo, JMS para HTTP.
- Suporte *out-of-the-box* de padrões de integração.
- Conexão a sistemas legados e proprietários: SAP, HL7, entre outros.
- Extensibilidade.

4.2.2 Seleção do ESB

Na secção 3.3 foram referidos os passos utilizados para proceder à seleção das tecnologias a utilizar. Um desses passos passava por uma reflexão sobre os conhecimentos tecnológicos do autor, tendo em conta o tempo disponível para o desenvolvimento. Sobre o tema de integração, concluiu-se que o autor possui conhecimentos em tecnologias da WSO2¹⁸, pelo que o ESB utilizado foi o WSO2 Enterprise Service Bus.

4.2.3 Solução a implementar

Tal como apresentado na secção 3.4.3.3, pretende-se que o ESB seja a peça central na interligação de todas as instâncias da aplicação “Educa”, para permitir a troca de dados de alunos. A Figura 38 apresenta um diagrama conceptual da solução a implementar.

¹⁸ Empresa fundada em 2006 que fornece tecnologia de código aberto, focando-se essencialmente em tecnologia de integração

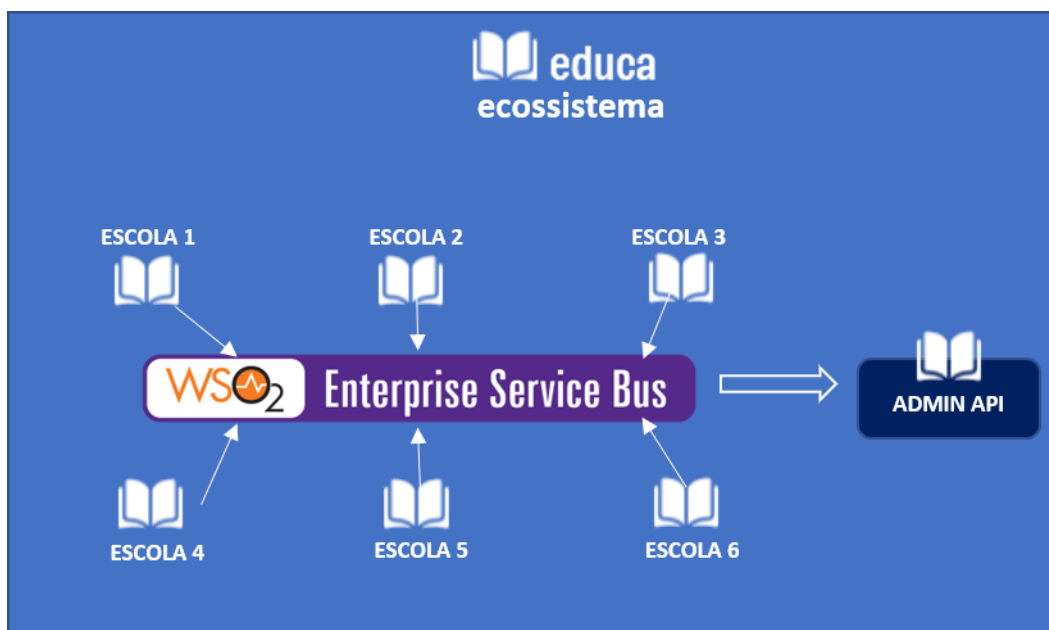


Figura 38 – Diagrama conceitual da solução a implementar

Analisando a figura, verifica-se que todas as instâncias da plataforma “Educa” comunicam com o ESB para realizar a integração de dados dos alunos. Por sua vez, o ESB comunica com uma API de administração da plataforma que possui um catálogo dos *endpoints* e escalas de avaliação de todas as instâncias em execução. Assim, o ESB consegue obter de forma dinâmica os *endpoints* para onde tem de redirecionar as mensagens.

4.2.4 Implementação do pipeline do ESB

A implementação do pipeline do ESB foi realizada em ambiente Linux, recorrendo ao IDE WSO2 Integration Studio¹⁹. Em seguida serão apresentados os desenvolvimentos realizados no ESB, desde a criação do serviço API, do *endpoint* para comunica com a “Admin API” e todos os passos seguidos na *insequence* e *outsequence*.

4.2.4.1 Criação da REST API

O ponto de entrada no ESB é uma REST API. A Figura 39 apresenta a configuração da API criada.

APIResource		
Core	Property	Value
Appearance	Basic	
	Url Style	URI_TEMPLATE
	URI-Template	/integrateStudentData
	Protocol	http,https

Figura 39 – REST API criada no ESB

¹⁹Sítio na web: <https://wso2.com/integration/integration-studio/>

Esta API recebe pedidos POST no endpoint “/integrateStudentData” e o JSON enviado segue o seguinte formato.

```
{
  "studentExternalCode": 1,
  "studentName": "Lima",
  "schoolExternalCode": 2,
  "schoolGradeScale": "es"
  "mobilityPlanHistory": {
    "lectiveYear": "2018/2019",
    "courseName": "Manutenção Industrial",
    "subjectsGrades": [
      {
        "subjectName": "Desenho",
        "grade": 19
      },
      {
        "subjectName": "Cálculo",
        "grade": 20
      }
    ]
  }
}
```

Código 1 – Formato do JSON enviado para a REST API do ESB

Os significados das propriedades enviadas no JSON estão definidos na tabela.

Tabela 14 – Especificação do JSON enviado para a REST API do ESB

Propriedade	Definição
studentExternalCode	Código do aluno na escola de origem
studentName	Nome do aluno
schoolExternalCode	Código da escola de origem do aluno
schoolGradeScale	Escala de avaliação utilizada na escola onde o aluno frequentou o programa de mobilidade
mobilityPlanHistory	Array com todas as notas do aluno, a todas as disciplinas, agrupadas por ano letivo

4.2.4.2 Criação do endpoint para o serviço externo “Educa Admin”

Para o ESB comunicar com o serviço “Educa Admin” é necessário definir um *endpoint* para esse serviço externo. O excerto do Código 2 mostra a definição do endpoint em XML. Este *endpoint* tem o nome “GetSchoolInfoEP”, trata-se de um pedido GET e envia no *url* a propriedade “schoolExternalCode”. A resposta deste serviço será a informação da escola pedida (*endpoint* da API da escola e a escala de avaliação utilizada).

```
<endpoint name="MOCK_ES_GetSchoolInfoEP" xmlns="http://ws.apache.org/ns/synapse">
  <http method="get" uri-
  template="http://www.educadm.azurewebsite.net/api/admin/getSchoolDetail?code={uri.v
  ar.schoolextcode}"/>
</endpoint>
```

Código 2 – Definição do *endpoint* do serviço externo “Educa Admin”

4.2.4.3 Vista geral do Pipeline do ESB

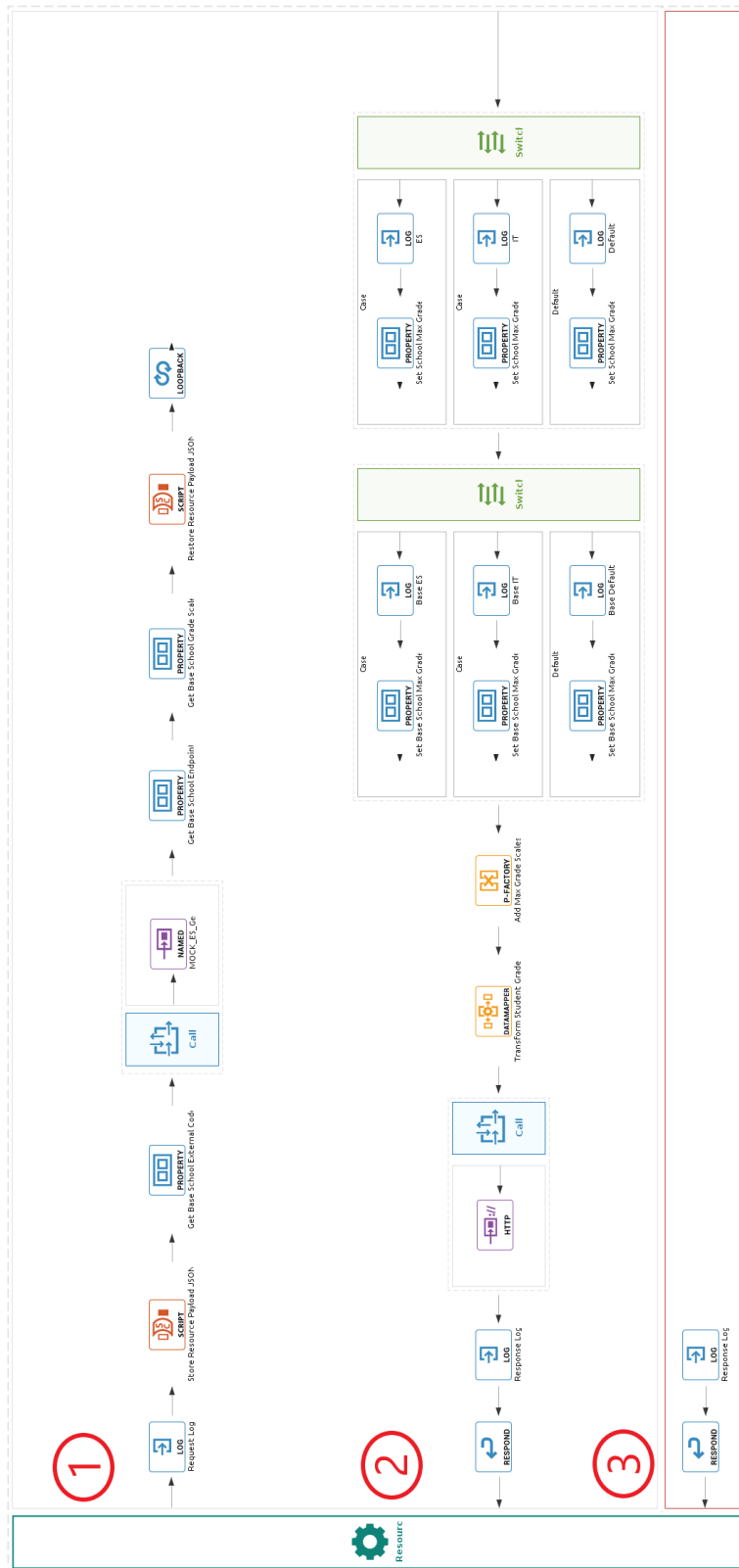


Figura 40 – Representação do pipeline implementado no ESB

A Figura 40 apresenta o pipeline implementado no ESB. Este pipeline encontra-se dividido em três partes (representadas pelos números na imagem), que são:

1. *Insequence*: é a sequência que recebe o pedido e o redireciona para outro serviço;
2. *Outsequence*: é a sequência que responde ao serviço que originou o pedido;
3. *Faultsequence*: é a sequência de falha, que contém a lógica de tratamento de erros.

4.2.4.4 Análise da *Insequence*

A Figura 41 apresenta o passo 1 da análise da *Insequence*.



Figura 41 – Análise da Insequence do ESB (passo 1)

A primeira tarefa realizada na *Insequence* é guardar o *payload* recebido (mensagem original) numa propriedade, para reutilizar posteriormente. Para esta tarefa foi utilizada uma Script Task. O Código 3 apresenta o código utilizado nesta tarefa.

```
var payload = mc.getPayloadJSON();
mc.setProperty("jsonPayload",JSON.stringify(payload));
```

Código 3 – Código utilizado na primeira tarefa

Na segunda tarefa, a propriedade “schoolExternalCode” é extraída do *payload* e é registada numa nova propriedade (“uri.var.schoolextcode”), que será utilizada na terceira tarefa, a invocação do serviço da “Admin API”. Este serviço irá retornar o endpoint da API e a escala de avaliação usada na escola de origem. Como está a ser usado um mediador “Call”, o fluxo continua na *Insequence*.

A Figura 42 apresenta o passo 2 da análise da *Insequence*.

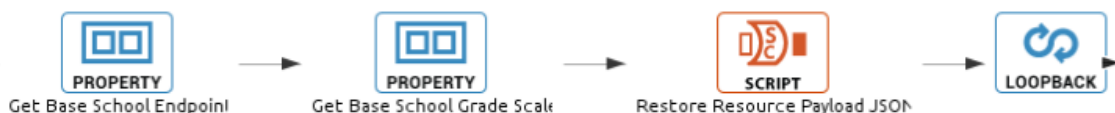


Figura 42 – Análise da Insequence do ESB (passo 2)

Depois de realizado o pedido ao serviço “Admin API”, é analisado o *payload* e o *endpoint* e a escala de avaliação da escola de origem são registadas em duas propriedades distintas (“BaseSchoolEndpoint” e “BaseSchoolGradeScale”). Depois disso é realizado o *restore* do *payload* inicial, isto porque após um pedido a um serviço, o *payload* que se encontrava ativo até então é perdido. Realizado um Loopback, o fluxo de mensagens transita para a Outsequence.

4.2.4.5 Análise da Outsequence

A Figura 43 apresenta o passo 1 da análise da Outsequence.

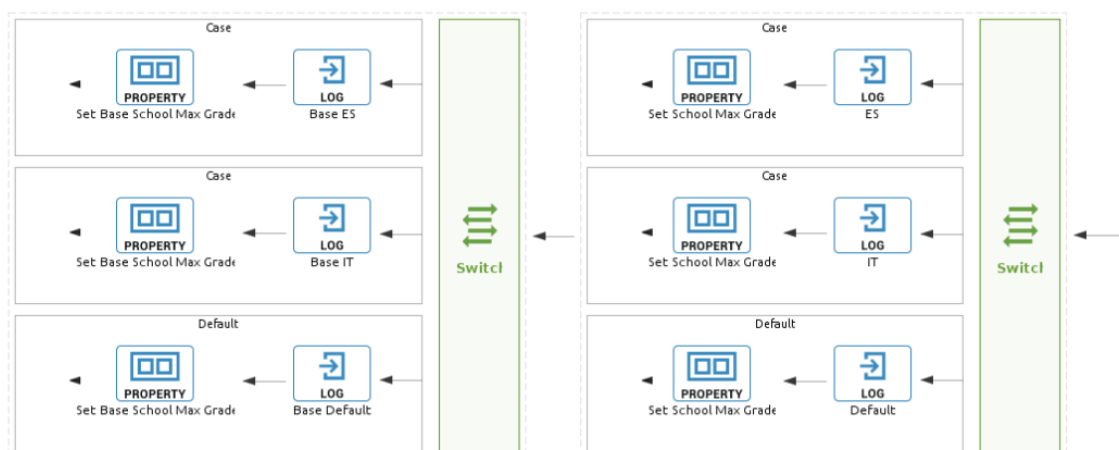


Figura 43 – Análise da Outsequence do ESB (passo 1)

O primeiro passo da Outsequence passa por utilizador dois mediadores Switch para verificar qual a escala de avaliação da escola de origem (segundo *switch*, no sentido da direita para a esquerda) e a escala de avaliação da escola onde o aluno realizou o programa de mobilidade (primeiro *switch*). Depois de identificada a escala, para cada um dos *switchs*, é criada uma propriedade com a nota máxima de cada uma das escalas. Estas propriedades irão ser utilizadas posteriormente no cálculo da conversão da nota. O Código 4 apresenta um excerto de um dos *switch*.

```
<switch source="json-eval($.schoolGradeScale)">
  <case regex="((^|, )(es))+$">
    <log description="ES" level="custom">
      <property name="school-grade-scale" value="es"/>
    </log>
    <property description="Set School Max Grade" name="SchoolMaxGrade"
scope="default" type="DOUBLE" value="10.0"/>
  </case>
  <case regex="((^|, )(it))+$">
    <log description="IT" level="custom">
      <property name="school-grade-scale" value="it"/>
    </log>
    <property description="Set School Max Grade" name="SchoolMaxGrade"
scope="default" type="DOUBLE" value="10.0"/>
  </case>
  <default>
    <log description="Default" level="custom">
      <property name="school-grade-scale" value="default"/>
    </log>
    <property description="Set School Max Grade" name="SchoolMaxGrade"
scope="default" type="DOUBLE" value="20.0"/>
  </default>
</switch>
```

Código 4 – Excerto de um dos *switch* representado na Figura 43

A Figura 44 apresenta o passo 2 da análise da Outsequence.



Figura 44 - Análise da Outsequence do ESB (passo 2)

Posteriormente à identificação das escalas de avaliação e das respetivas notas máximas, é criado um novo *payload* utilizando o mediador Payload Factory. Esse *payload* transita para um mediador DataMapper, onde será realizado o cálculo da nota (Figura 45).

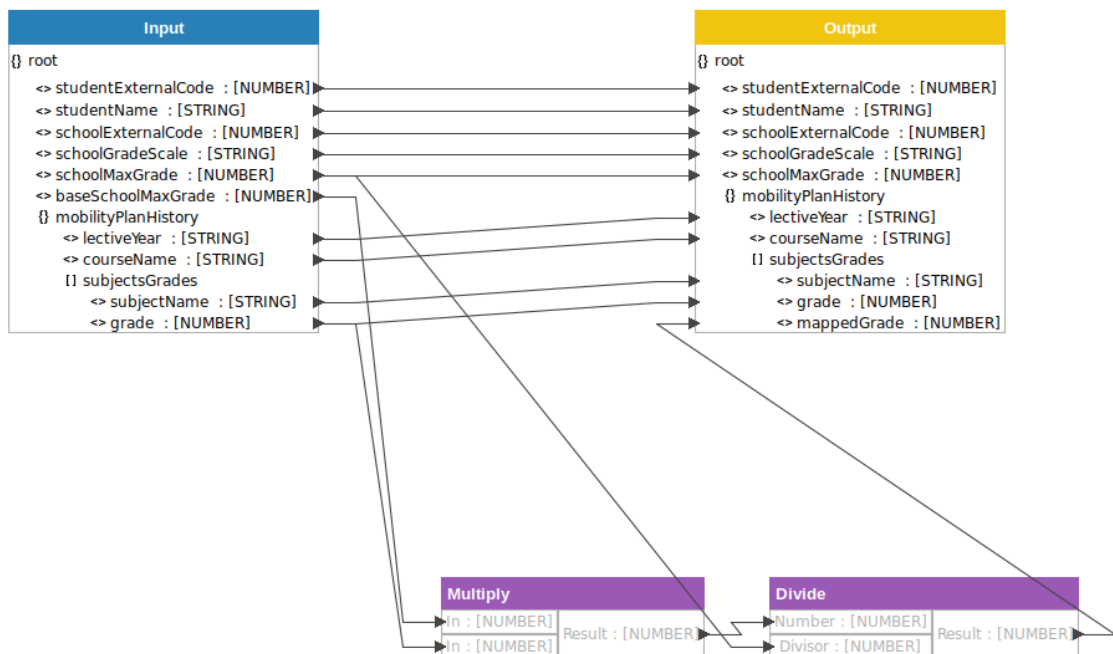


Figura 45 – Mapeamento e cálculo da nota

Como se pode verificar pela Figura 45, para cada registo é realizada uma operação aritmética durante o processo de mapeamento, que irá dar origem à nota convertida.

Depois do mapeamento, o serviço da API da escola de origem do aluno é invocado com a informação já mapeada.

5 Avaliação

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live”

John F. Woods

Como referido inicialmente, este projeto sofreu uma série de teste de *software* (unitários, integração). A metodologia utilizada no desenvolvimento dos testes, assim como os seus resultados, serão especificados neste capítulo.

Também será explicado o guião de testes criado para os testes de usabilidade, os cenários de testes criados e os resultados obtidos.

5.1 Uma breve introdução aos testes

Num mundo ideal, todos querem sempre testar todas as partes do *software* e atingir as fantásticas métricas de 100% de *code coverage*²⁰, no entanto, isto é praticamente impossível. Até o programa mais simples pode ter uma quantidade ínfima de combinações de *inputs* e *outputs* e criar casos de testes para todas as possibilidades é impraticável [41]. Testar uma aplicação na sua totalidade é uma tarefa demorosa e que requer bastantes recursos humanos, o que significa um acréscimo monetário considerável ao valor do projeto.

Com esta introdução, não se está a defender e a apoiar a não implementação de testes. Os testes são das tarefas mais importantes no desenvolvimento do *software*, pois são estes que vão garantir que o *software* “faz bem o que tem que fazer”. No entanto, as equipas não devem viver iludidas nos valores de 100% de *code coverage*, e por várias razões, tais como:

1. Taxas de 100% de *code coverage* não significam que todas as possibilidades de teste estão cobertas;

²⁰ Métrica de *software* que indica a taxa de cobertura de testes

2. *Code coverage* ≠ testes com qualidade.

O que as equipas devem procurar é saber olhar para o produto que estão a desenvolver e perceber:

1. Que testes vão realizar;
2. Que funcionalidades vão testar, com principal destaque para as mais críticas;
3. Qual a taxa de *code coverage* que cria uma confiança aceitável para disponibilizar o produto em produção.

Com isto pretende-se realçar que tudo deve ser analisado e adaptado ao contexto e realidade que se está a viver, e nunca testar apenas por testar, ou na busca de um “100%” de cor verde num relatório final de execução de testes, que, na realidade, poderá significar um “50%”.

5.2 Implementação dos testes

Nos objetivos iniciais deste projeto definiu-se que se iriam implementar testes unitários, de integração e de sistema. A Figura 46 apresenta a estrutura de testes no projeto “Educa”.

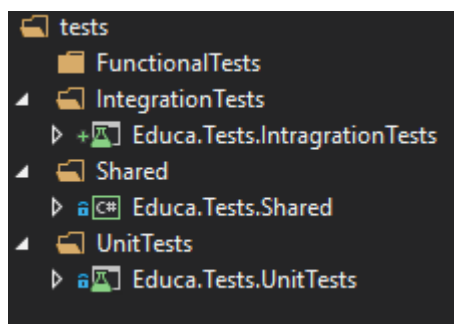


Figura 46 – Estrutura de testes do projeto “Educa”

As próximas subsecções detalham a implementação de cada um desses tipos de teste.

5.2.1 Testes unitários

O objetivo dos testes unitários é testar uma parte única da lógica aplicacional, e não testar como o código funciona com outras dependências externas. Estes tipos de testes correm em memória, num processo, e não dependem da base de dados, nem de acessos a ficheiros [34].

Dadas estas características, deve-se sempre desenvolver um grande número de testes unitários, já que os seus tempos de execução são bastante positivos. Aliás, estes testes devem ser executados com bastante frequência, idealmente sempre que existe um envio de código para o sistema de controlo de versões.

Neste projeto os testes unitários incidiram sobre as entidades do domínio e os serviços aplicacionais. Visto que os serviços aplicacionais dependem dos repositórios de acesso a dados

(através de interfaces) foi utilizada a *framework* Moq, que permite resolver estas dependências com dependências *mock*²¹. Como ferramenta de testes utilizou-se o xUnit²².

Para o desenvolvimento dos testes foi seguido o padrão AAA (Arrange, Act, Assert). Tem conceito tem uma funcionalidade específica, que é [42]:

- **Arrange:** secção do código de teste onde os objetos são inicializados e os parâmetros enviados para o método a testar são preenchidos;
- **Act:** secção do código de teste que invoca o método a testar;
- **Assert:** secção do código de teste que verifica que o método sujeito ao teste tem o comportamento esperado.

Também a nomenclatura dos casos de teste segue um padrão comum, para que se consiga perceber apenas pelo seu nome qual é a tua finalidade e comportamento esperado.

```
[Fact]
public async Task
Should_RegisterUserOutDtoSuccessPropertyBeFalse_When_UserIsNotCreated()
{
    // Arrange
    var inputDto = _userTestUtil.GetGenericRegisterUserInDto();

    var identityUser = new User(inputDto.FirstName, inputDto.LastName,
inputDto.AvatarUrl, inputDto.NickName, inputDto.Email);
    var educaUser = new EducaUser(1);

    _mockIdentityService.Setup(
        x => x.CreateUser(inputDto.FirstName, inputDto.LastName,
inputDto.AvatarUrl, inputDto.Email, inputDto.Password, inputDto.NickName,
inputDto.Role)
    ).ReturnsAsync((User)null);

    _mockUnitOfWork.Setup(x => x.EducaUserRepository.Add(educaUser));

    var useCase = new UserService(_mockUnitOfWork.Object,
_mockIdentityService.Object);

    // Act
    var result = await useCase.RegisterUser(inputDto);

    // Assert
    Assert.False(result.IsSuccess);
}
```

Código 3 – Excerto de código de um teste de criação de utilizadores

O excerto Código 3 apresenta um teste unitário à lógica de registo de utilizadores. É possível verificar o uso do padrão AAA, já que o bloco se encontra claramente dividido em três partes. Este teste pretende testar o fluxo de erro quando o registo de um novo utilizador falha. O caso de uso do registo de utilizador depende do serviço gestão de identidade para a criação de

²¹ Neste caso são utilizados *mock objects*, que são objetos que imitam objetos reais para a realização dos testes

²² Sítio na web: <https://xunit.net/>

utilizadores. Para este teste, esse serviço está a ser utilizado através de *mock objects*, o que permite controlar o retorno do mesmo. Neste caso, está-se a indicar explicitamente que se pretende que esse serviço falhe no registo, para depois validar que o retorno do serviço em teste indica a ocorrência de um erro.

5.2.2 Testes de integração

Os testes de integração têm o objetivo de testar o sistema como um todo, testando todas as interfaces e interações entre os diferentes componentes de um *software* (incluindo a base de dados, *file system*, etc.) [43].

Até à data de finalização desta dissertação não foram realizados qualquer tipo de testes de integração, estando os mesmo planeados para as iterações futuras. No entanto, e tal como se pode verificar na Figura 46, a estrutura de testes já está preparada para suportar este tipo de testes.

5.2.3 Testes de sistema

Com os testes de sistema pretende-se perceber como o sistema reage a uma carga anormal de utilizadores. Voltando a analisar a Tabela 11 – Requisitos não funcionais identificados, verificamos que o requisito não funcional RNF08 especifica que “os relatórios e certificados têm de ser gerados, no máximo, em 3 segundos”. Assim, decidiu-se selecionar o relatório de detalhe de um curso para a execução dos testes. Este relatório retorna o número total de disciplinas, alunos, professores, diretores de cursos e *outcomes* que estão associados ao curso em questão.

Os testes de carga executaram no ambiente Azure, sobre a aplicação “Educa” que foi instalada no ambiente de demonstração. O teste simulou 250 utilizadores em simultâneo a tentar gerar o relatório. O tempo de execução do teste foi de cinco minutos. A Figura 47 apresenta os resultados do teste.

Analisando os resultados da Figura 47 verifica-se que foram executados, no total, 68339 pedidos, o que dá uma média de 227.8 pedidos por segundo. O tempo de resposta médio de cada pedido foi de 0.08 segundos e a taxa de sucesso foi de 100%. Perante o requisito RNF08, e para a geração de relatórios, o sistema cumpre com o especificado. De realçar que a implementação deste caso de uso utiliza um micro-ORM.

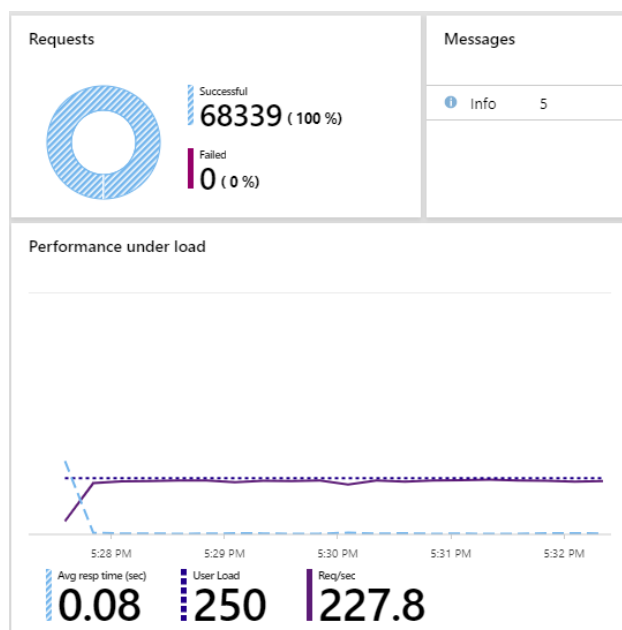


Figura 47 – Teste de carga na geração de relatórios

O teste seguinte passou pela realização de um teste de carga no processo de geração de certificados de alunos. Este teste foi realizado nas mesmas condições do teste anterior, ou seja, com 250 utilizadores em simultâneo a gerar o certificado para um aluno, durante cinco minutos. A Figura 48 apresenta os resultados do teste.

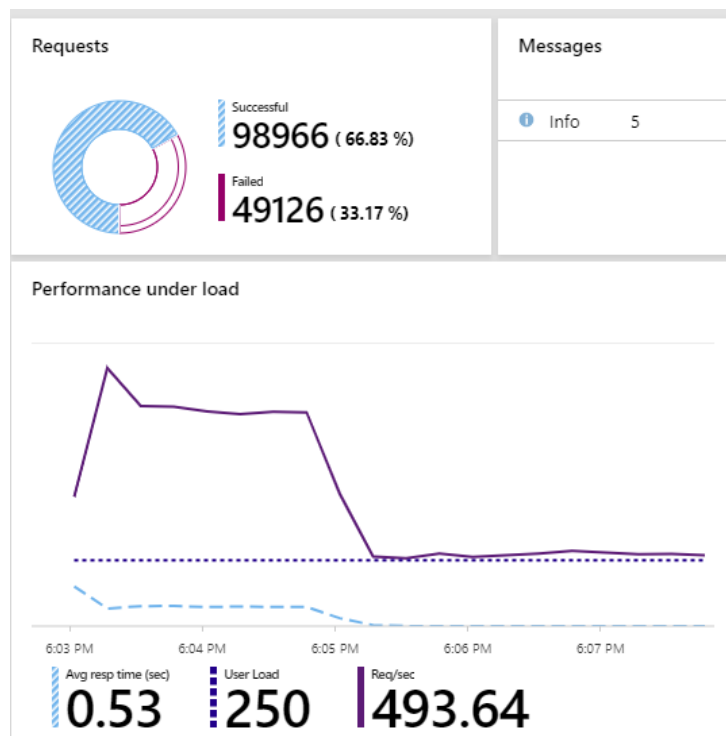


Figura 48 – Teste de carga na geração de certificados

Analisando a Figura 48 verifica-se que os resultados já são diferentes, comparando com o último teste. Neste caso foram realizados, em média, 493.64 pedidos de geração de certificados por segundo, com um tempo médio de resposta de 0.53 segundos. Estes valores cumprem com o requisito não funcional RNF08, no entanto, é importante realçar que este teste obteve uma taxa de sucesso de 66.83%, o que significa que 33.17% dos pedidos realizados falharam. A taxa de pedidos com erros pode ser reduzida utilizando um micro-ORM no serviço de geração de certificados, tal como apresentado na subsecção 3.4.4. Na análise deste requisito identificou-se essa necessidade, no entanto, a implementação atual da geração de relatórios encontra-se a usar um ORM tradicional.

A Tabela 15 resume os resultados dos testes de carga.

Tabela 15 – Resumo dos testes de carga efetuados

Teste	Nº de utilizadores	Nº de pedidos total	Nº de pedidos (por segundo)	Tempo médio de resposta (segundos)	Pedidos com sucesso (%)	Pedidos com erro (%)
Geração de relatório	250	68339	227.8	0.08	100	0
Geração de certificado	250	148092	493.64	0.53	66.83	33.17

5.3 Avaliação de usabilidade

Esta secção é dedicada à avaliação de usabilidade realizada à aplicação desenvolvida. Para realizar esta avaliação, foi realizada uma sessão de demonstração do funcionamento da aplicação a professores da Escola Profissional de Aveiro, e também a alguns membros do GECAD. Posteriormente disponibilizou-se um guião de testes, acompanhado de um questionário online, para os participantes responderem. O principal objetivo do questionário era recolher a opinião do utilizador sobre a utilização e aspeto visual da aplicação “Educa”.

Depois de recolhidos os resultados dos questionários, os mesmos foram analisados e as conclusões foram formuladas.

5.3.1 Metodologia para a sessão de avaliação

Antes de se iniciar os planeamentos das sessões de avaliação da aplicação desenvolvida, realizou-se uma sessão de demonstração da aplicação a professores da Escola Profissional de Aveiro, e a alguns elementos do GECAD. Inicialmente estava planeado que esta sessão fosse realizada presencialmente, o que acabou por não se verificar, devido a indisponibilidades de ambas as partes. Assim, a sessão realizou-se via Skype e teve uma duração aproximada de 1h30 (uma hora e trinta minutos). Após a sessão foi disponibilizado via email um guião de testes, acompanhado de um questionário *online*.

O guião para a avaliação de usabilidade apresenta na sua constituição os seguintes elementos: uma introdução, onde é apresentado o objetivo do guião e as pré-configurações dos ambientes de testes, os utilizadores e respetivas credenciais de acesso, e os cenários de teste em avaliação. Os cenários de teste pretendem analisar os seguintes casos:

1. Primeiro cenário: cenário exclusivo do Diretor de Curso, onde se pretende analisar a experiência de navegação nas tarefas de gestão de cursos, disciplinas, turmas, *outcomes* e certificados.
2. Segundo cenário: simulação do momento de avaliação de uma turma do 3ºano do curso de Manutenção Industrial.

O questionário final foi desenvolvido no Google Forms²³, uma ferramenta *online* disponibilizada pela Google que permite a criação de questionários. O questionário desenvolvido encontra-se dividido nas seguinte oito secções:

1. Cargo na instituição: pretende-se saber qual o cargo da pessoa que está a responder ao questionário (Administrador, Diretor de Curso, Professor ou Orientador de Empresa);
2. Experiência de utilização: pretende-se analisar a experiência global de usabilidade da plataforma;
3. Funcionalidades de administração: pretende-se avaliar as principais funcionalidades de administração, realizadas pelo administrador;
4. Funcionalidades de gestão de curso: pretende-se avaliar as principais funcionalidades de gestão de curso, realizadas pelo diretor de curso;
5. Funcionalidades de gestão de disciplinas: pretende-se avaliar as principais funcionalidades de gestão de disciplinas, realizadas pelo diretor de curso;
6. Avaliação de alunos: pretende-se avaliar o processo de avaliação de alunos;
7. Avaliação final: pretende-se recolher a opinião global da aplicação;
8. Recomendações: campo de resposta aberta onde podem ser colocadas recomendações para o melhoramento da aplicação.

Todas as questões do guião seguem a escala de Likert. Esta escala é um intervalo de valores que descreve um determinado atributo, sendo que cada valor da escala representa um nível diferente desse atributo. O objetivo principal desta escala é medir o nível de concordância ou não concordância referente a uma afirmação [44]. A tabela apresenta os valores da escala usados neste questionário.

²³ Sítio na web: <https://www.google.com/forms/about/>

Tabela 16 – Escala usada no questionário de usabilidade

Escala
Discordo totalmente
Discordo
Sem opinião
Concordo
Concordo totalmente

Na questão do grupo 8 do questionário, a avaliação final, não existe a opção central “Sem opinião”, já que aqui pretende-se que o sujeito faça uma escolha positiva ou negativa.

A definição do número de participantes que deveriam realizar a sessão baseou-se nos estudos de Virzi. Segundo Virzi [45], uma sessão de avaliação com apenas cinco participantes permite identificar aproximadamente 80% dos problemas de usabilidade. Nesta sessão participaram 10 pessoas, pelo que, teoricamente, será possível identificar 90% dos problemas de usabilidade [46].

5.3.2 Resultados e interpretação de resultados

Nesta subsecção são apresentados os resultados obtidos no questionário. Neste questionário participaram 10 pessoas: quatro professores, dois orientadores de empresa, dois administradores e dois diretores de curso. Esta distribuição está representada no gráfico da Figura 49.

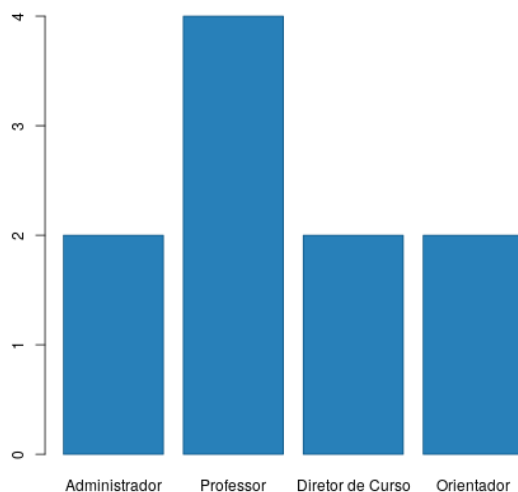


Figura 49 – Cargos dos participantes

O gráfico da Figura 50 ilustra a opinião dos participantes quanto à experiência de utilização da aplicação. Analisando o gráfico verifica-se que a opinião dos participantes quanto à experiência de utilização é bastante positiva. De todos os pontos em análise, destaca-se uma classificação “Discordo” no que diz respeito ao bom comportamento da aplicação em dispositivos móveis.

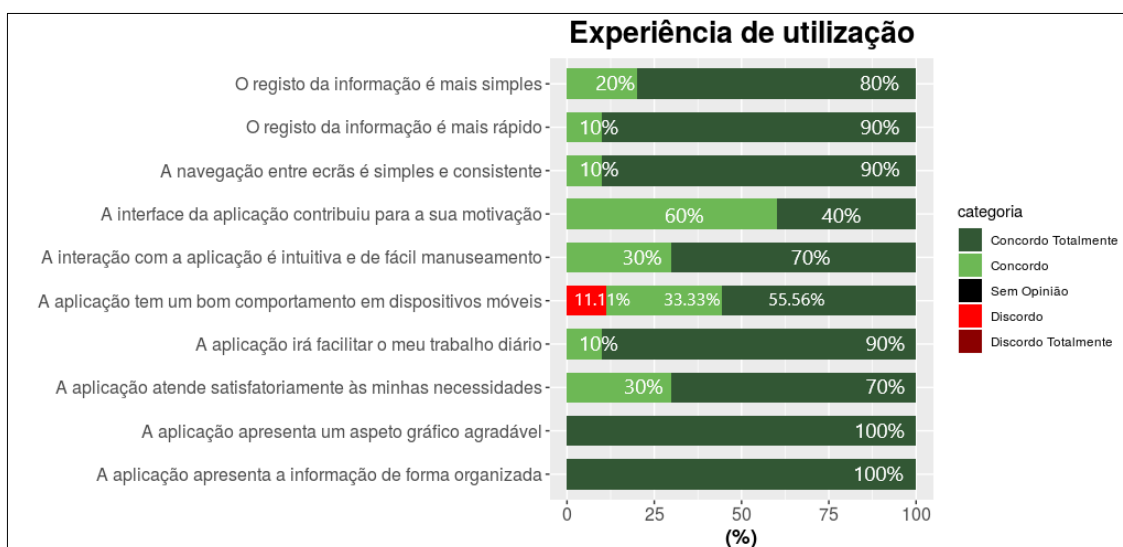


Figura 50 – Classificação da experiência de utilização

Quando questionados sobre a experiência de utilização das funcionalidades de administração, os participantes foram unânimes e concordaram totalmente em todas as afirmações (Figura 51).

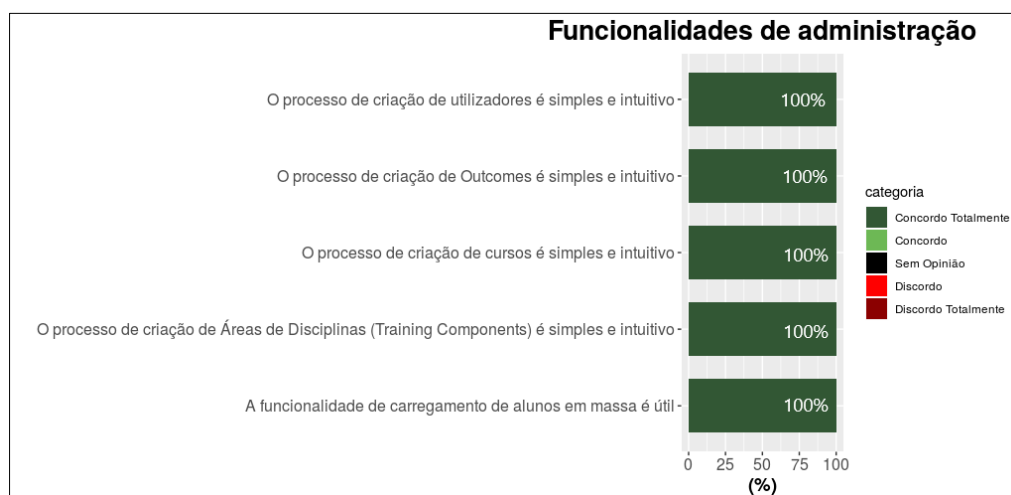


Figura 51 – Classificação das funcionalidades de administração

O gráfico da Figura 52 ilustra a opinião dos participantes relativamente à experiência de utilização das funcionalidades de gestão de disciplinas. Todos os participantes concordam totalmente sobre a simplicidade do processo de criação de outcomes de disciplinas. Quanto à visualização dos outcomes criados, os participantes já dividem as opiniões entre o “Concordo totalmente” e “Concordo”.

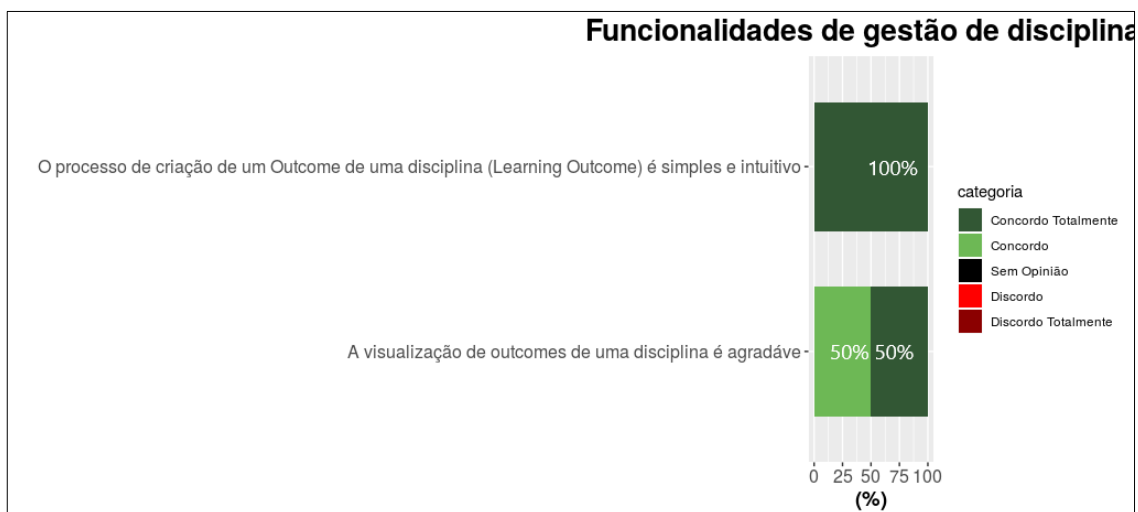


Figura 52 – Classificação das funcionalidades de gestão de disciplinas

Na Figura 53 encontra-se ilustrada a opinião dos participantes relativamente à experiência de utilização das funcionalidades de gestão de cursos. Neste gráfico destaca-se a divisão de opiniões nos processos de criação de turmas e de disciplinas. Apesar desta divisão, ambas as opiniões são positivas.

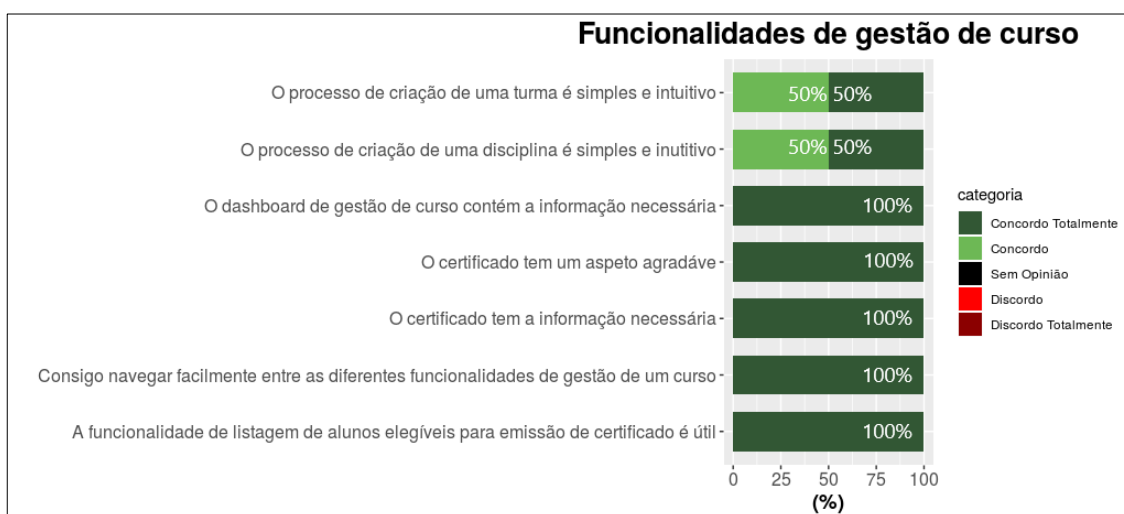


Figura 53 – Classificação das funcionalidades de gestão de curso

No gráfico da Figura 54 encontra-se ilustrada a opinião dos participantes relativamente ao processo de avaliação dos alunos. Todos os participantes concordam totalmente sobre a facilidade do processo de avaliação, no entanto, 10% dos participantes não concordam totalmente sobre a presença da informação necessária no formulário de avaliação.

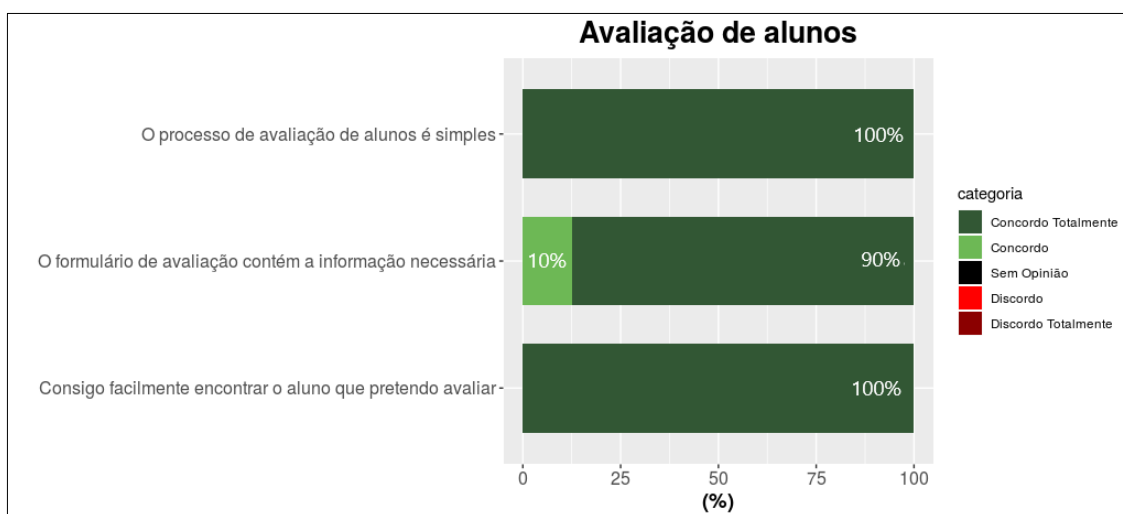


Figura 54 – Classificação do processo de avaliação de alunos

No gráfico da Figura 55 encontra-se ilustrada a avaliação final dos participantes. Analisando a figura verifica-se que 90% dos participantes avaliam a aplicação como “Muito Boa” e os restantes 10% avaliam como “Boa”.

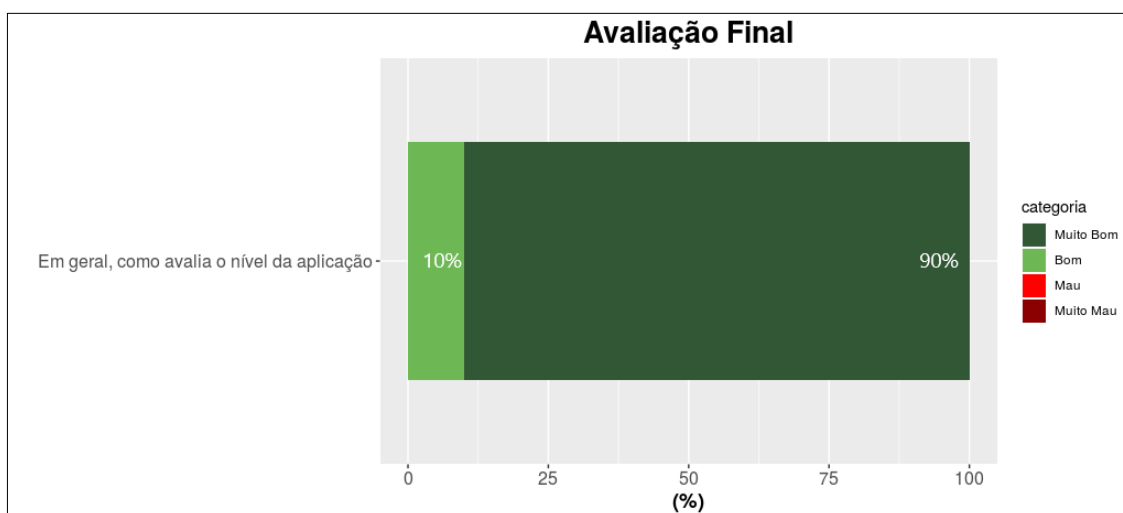


Figura 55 – Avaliação final da plataforma

Na última secção do questionário era solicitado aos participantes a introdução de alguns comentários ou sugestões. Na Tabela 17 são apresentadas as sugestões e comentários recebidos.

Tabela 17 – Sugestões e comentários dos participantes

Disponibilização de funcionalidade para gerar certificados em massa.
O botão para abrir a página de gestão de uma disciplina está como <i>dropdown</i> . Poderia ser um botão normal.
Na página principal apresentar atalhos para os cursos/disciplinas ativos
Disponibilização de ferramenta para criar cursos/turmas em massa

5.3.3 Apreciações finais

A avaliação da usabilidade da aplicação teve como objetivo analisar a interação dos participantes com a aplicação. A elaboração do guião foi essencial para o sucesso do questionário, já que na sessão de demonstração os participantes sentiram dificuldades em reter toda a informação passada.

Após a análise dos dados recolhidos verifica-se que os participantes vêem esta aplicação como uma ferramenta que irá simplificar o trabalho diário. No geral, todos concordam totalmente com todas as afirmações, havendo alguma divisão nos processos de criação de disciplinas e turmas. No entanto, apesar das divisões, ambas as classificações são positivas. No que diz respeito ao aspeto visual da aplicação, 60% dos participantes concordam que o aspeto é agradável, e os restantes 40% concordam totalmente. Quanto à responsividade em dispositivos móveis, existe alguma divisão das classificações, com 11.1% dos participantes a discordarem da boa responsividade em dispositivos móveis. A avaliação final dos participantes veio confirmar que a opinião geral sobre a aplicação é bastante positiva.

6 Conclusões

Este capítulo apresenta um resumo do trabalho desenvolvido, destacando os pontos mais relevantes. São também descritos os objetivos realizados, bem como as limitações e trabalho futuro.

6.1 Resumo

O Abandono Escolar precoce é um problema conhecido na União Europeia. De acordo com a estratégia “Europa 2020”, até o ano 2020, a percentagem de AEP deve ser inferior a 10% e pelo menos 40% da geração jovem deve possuir formação superior. No sentido de cumprir com os objetivos estipulados pela Comissão Europeia, surge o projeto piloto “**From birth to adult age – a WBL successful Practice!**”. Um dos grandes objetivos deste projeto é o desenvolvimento de uma plataforma informática de validação de competências de alunos.

Esta dissertação teve como objetivo desenvolver a plataforma supramencionada, seguindo as melhores práticas de desenvolvimento de *software*.

No capítulo 1 foi realizada uma breve contextualização, a definição do problema, os objetivos estipulados e resultados esperados. Já no capítulo 2 foi apresentado o estado da arte, onde se definiram vários conceitos essenciais para a dissertação. Foi neste capítulo que se verificou que a plataforma desenvolvida se insere nas áreas do LMS, onde existem já alguns produtos *open-source* líderes do mercado, como o Moodle e o OLAT. Ainda no capítulo 2 foi apresentada a análise de valor.

No capítulo 3 foi realizada a análise o *design* da solução a desenvolver. Foram apresentados o modelo de domínio e os requisitos funcionais e não-funcionais identificados, requisitos esses que vieram a definir a arquitetura final da solução. Com base nos requisitos identificados foram apresentadas as tecnologias utilizadas, onde se verificou que o ASP.NET Core seria a melhor solução a adotar. Na análise da melhor arquitetura a implementar, verificou-se a necessidade da existência de um ESB, por forma a facilitar os requisitos de integração de dados entre as diferentes instâncias da aplicação. Foi aqui também que se detetou que a utilização de um micro-ORM poderia ser útil para funcionalidades de geração de relatórios e certificados. Por

fim, foi apresentada uma possível solução de implantação do sistema e também o *pipeline* de CI utilizado durante todo o processo de desenvolvimento.

No capítulo 4 foram apresentados os detalhes de implementação da solução desenvolvida, onde se procurou detalhar os aspetos mais significativos, tais como alguns detalhes de implementação da aplicação *web* e alguns detalhes de implementação do ESB. No que diz respeito ao ESB, concluiu-se que o WSO2 Enterprise Service Bus seria a melhor solução para o desenvolvimento deste projeto.

Posteriormente, no capítulo 5, foi discutida a avaliação da solução. Verificou-se que foram realizados testes unitários, de sistema e de usabilidade, não estando, para já, desenvolvidos quaisquer testes de integração. Nos testes de sistemas verificou-se que o sistema respondeu de forma positiva aos testes de carga efetuados. Já na análise dos resultados dos testes de usabilidade, concluiu-se que, de forma geral, os participantes do questionário concordam que a plataforma é útil e que lhes pode ser útil nas tarefas diárias. Verificou-se também que poderão existir algumas inconsistências na responsividade em dispositivos móveis, já que foi o ponto onde houve mais discrepância de classificações.

6.2 Objetivos realizados

Na secção foram definidos os objetivos propostos para o desenvolvimento desta dissertação. A tabela apresenta esses objetivos e as suas percentagens de conclusão.

Tabela 18 – Objetivos propostos e percentagem de conclusão

Objetivo	Percentagem de conclusão
Levantamento e especificação de requisitos	100%
Desenvolvimento da arquitetura	100%
Desenvolvimento da aplicação	100%
Suporte de várias línguas	100%
UI responsiva	100%
Integração de dados entre instâncias	100%
Geração de relatórios	20%
Geração de certificados	100%
Apresentação de indicadores	30%
Realização de testes unitários	90%
Realização de testes de integração	0%
Realização de testes de sistema	100%
Realização de testes de usabilidade	100%

Analisando a percentagem de conclusão dos objetivos estipulados, verifica-se que uma grande maioria dos objetivos foram cumpridos com sucesso. De facto, a não geração de relatórios e a não realização de testes de integração foram um aspeto negativo, no entanto, a plataforma encontra-se preparada para suportar estes desenvolvimentos.

É importante realçar que o requisito da funcionalidade de integração de dados entre instâncias não estava estipulado nos desenvolvimentos, tendo surgido no início do mês de maio.

6.3 Limitações e trabalho futuro

Durante a fase de desenvolvimento identificaram-se algumas limitações que poderão ser alvo de reparo em trabalho futuro.

A nível arquitetural a aplicação suporta a geração de relatórios, no entanto, nesta fase ainda não estão a ser gerados quaisquer relatórios. Quanto à apresentação de indicadores, nesta fase a aplicação apresenta um conjunto de indicadores por curso, mas estes também podem evoluir.

Relativamente aos desenvolvimentos futuros, devem ser acrescentados mais indicadores à aplicação e devem ser desenvolvidos relatórios. Quanto à implementação do ESB, é necessário garantir a autenticação entre os vários intervenientes no processo de integração (Educa, ESB e API de Administração). Nesta fase, as integrações são realizadas, mas não existe qualquer tipo de autenticação entre os diferentes sistemas. Quanto à realização de testes, a aplicação deve ser complementada com o desenvolvimento de testes de integração.

Visto este projeto tratar-se de um piloto, não foram desenvolvidos mecanismos de segurança que vão de encontro à lei da proteção de dados (RGPD). No entanto, a tecnologia escolhida para o desenvolvimento da plataforma já fornece algum suporte. Será interessante como trabalho futuro analisar este ponto.

Referências

- [1] European Commission, “Reducing early school leaving,” 2011.
- [2] European Commission, “Europe 2020 - A European strategy for smart, sustainable and inclusive growth,” Bruxelas, 2010.
- [3] European Commission, “Early leavers from education and training,” 2019. [Online]. Available: <https://ec.europa.eu/eurostat/web/education-and-training/visualisations>. [Accessed: 19-Feb-2019].
- [4] R. Dale, “Early school leaving: Lessons from research for policy makers (An independent expert report submitted to the European Commission),” p. 60, 2010.
- [5] McKinsey Center for Government, “Education to Employment: Designing a System that Works,” 2012.
- [6] European Commission, “Cooperação europeia (EF 2020),” 2019. [Online]. Available: https://ec.europa.eu/education/policies/european-policy-cooperation/development-skills_pt. [Accessed: 18-Feb-2019].
- [7] European Commission, “Grupos de trabalho EF 2020,” 2019. [Online]. Available: https://ec.europa.eu/education/policies/european-policy-cooperation/et2020-working-groups_pt.
- [8] European Commission, “Erasmus+,” 2019. [Online]. Available: http://ec.europa.eu/programmes/erasmus-plus/about_pt.
- [9] European Commission, “Erasmus + - Guia do Programa,” 2019.
- [10] Comissão Europeia, “Repensar a educação - Investir nas competências para melhores resultados socioeconómicos,” pp. 0–21, 2012.
- [11] European Commission, “EFP - Ensino e Formação Profissional,” 2019. [Online]. Available: https://ec.europa.eu/education/policies/eu-policy-in-the-field-of-vocational-education-and-training-vet_pt. [Accessed: 19-Feb-2019].
- [12] European Commission, “Vocational education and training for better skills, growth and jobs,” *Comm. Staff Work. Doc.*, pp. 1–57, 2012.
- [13] “O que é LMS?” [Online]. Available: <https://www.edools.com/faq/o-que-e-lms/>. [Accessed: 10-Apr-2019].
- [14] E. Prata and V. Marques, “Adaptive E-Learning Systems Foundations and Implementation,” *E-learning 2012, IADIS Int. Conf.*, 2012.
- [15] N. Dabbagh and B. Bannan-Ritland, *Online Learning- Concepts, Strategies, and Applications*. Upper Saddle River, NJ: Pearson, 2005.
- [16] A. Piña, “An Overview of Learning Management Systems,” 2010.

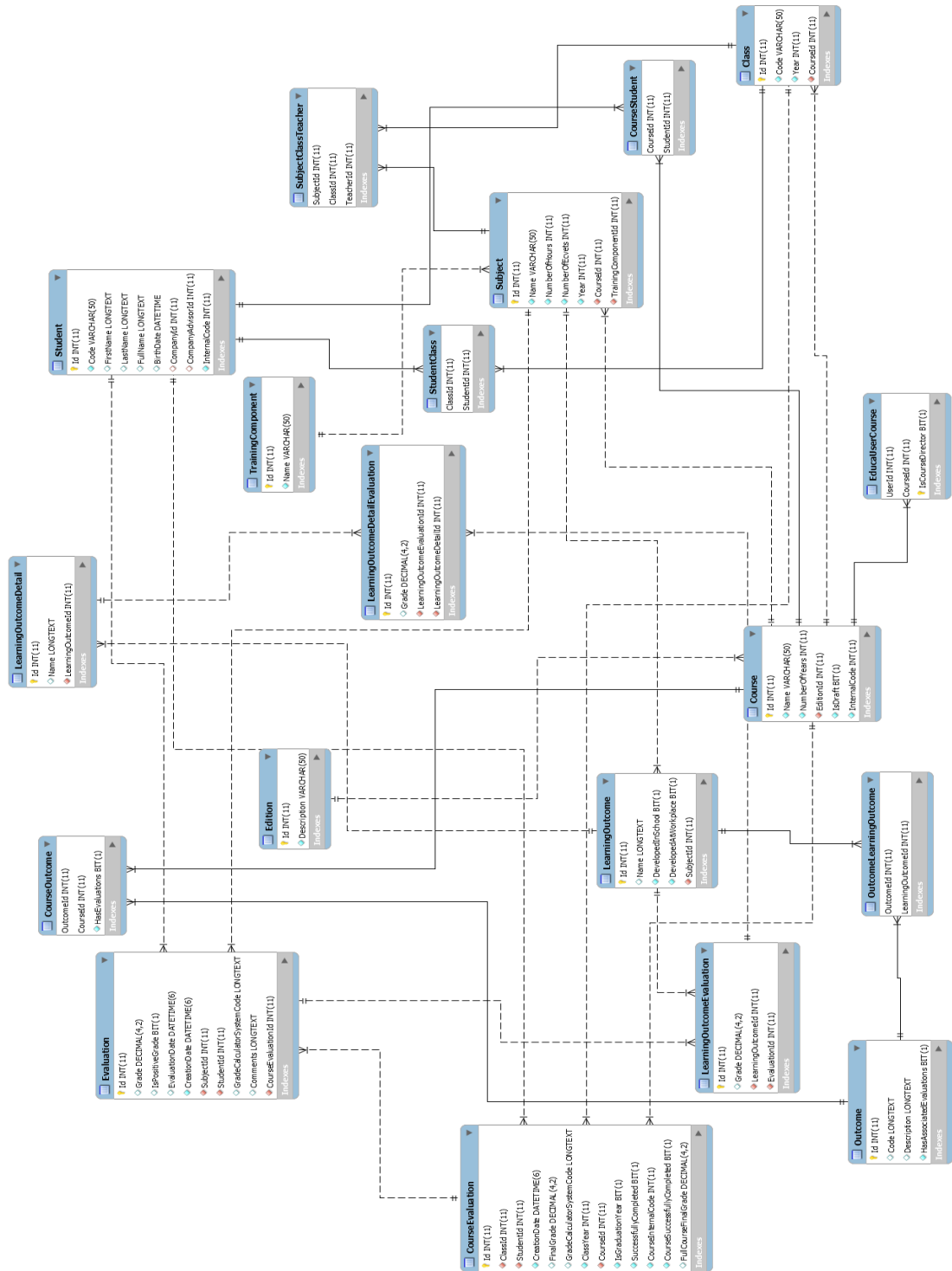
- [17] Z. Balogh and M. Turcáni, "Possibilities of modelling web-based education using IF-THEN rules and fuzzy petri nets in LMS," Faculty of Natural Sciences, Constantine the Philosopher University in Nitra, 2011.
- [18] Moodle, "Moodle," 2018. [Online]. Available: https://moodle.org/pluginfile.php/2840042/mod_page/content/19/Moodle-Logo-RGB%281%29.png. [Accessed: 20-Jun-2019].
- [19] Ed-room, "Plataforma Moodle - Características gerais." [Online]. Available: <https://ed-room.com/moodle/plataformamoodle>. [Accessed: 19-Jun-2019].
- [20] M. Piotrowski, "What is an E-Learning Platform?," ZHAW Zurich University of Applied Sciences, 2010.
- [21] "OLAT," 2018. [Online]. Available: <https://elearningindustry.com/directory/elearning-software/olat>. [Accessed: 16-Jun-2019].
- [22] Y. Kats, "Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications," Ellis University, USA & Rivier College, USA.
- [23] "Blackboard." [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/7/77/Blackboard_Inc._logo.png. [Accessed: 18-Jun-2019].
- [24] Blackboard, "Competency-Based Education," 2018. [Online]. Available: <https://www.blackboardopenlms.com/resource/cbe/>. [Accessed: 12-Jun-2019].
- [25] P. Teza, V. Miguez, R. Fernandes, and G. Dandolini, "Direcionadores do processo de inovação: o papel da estratégia, liderança e cultura," Federal Insitute of Santa Catarina, 2010.
- [26] P. Clements *et al.*, *Documenting Software Architectures - Views and Beyond*, Second. Addison Wesley, 2011.
- [27] A. Van Lamsweerde, *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley, 2009.
- [28] E. Evans, "Domain - Driven Design Reference," p. 59, 2016.
- [29] R. Pressman and B. Maxim, *Software Engineering - A Practitioners Approach*, 8th Editio. McGraw-Hill Education, 2015.
- [30] J. De Oliveira and M. Bruchet, *Learning ASP.NET Core 2.0*. Birmingham - Mumbai: Packt, 2017.
- [31] J. P. Smith, *Entity Framework Core In Action*. Manning, 2018.
- [32] P. Clements, L. Bass, and R. Kazman, *Software Architecture in Practice*, 3rd Editio. Addison Wesley, 2013.
- [33] R. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, 1st Editio. Prentice Hall, 2017.
- [34] S. Smith, *Architecting Mordern Web Applications with ASP.NET Core and Microsoft*

Azure. Microsoft Corporation, 2019.

- [35] K. Indrasiri, *Beginning WSO2 ESB - A comprehensive beginner to expert guide for learning WSO2 ESB 5.0*, 1st Editio. Apress, 2016.
- [36] J. Lerman, "Dapper, Entity Framework and Hybrid Apps," 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/magazine/mt703432.aspx>. [Accessed: 24-May-2019].
- [37] S. Vadapalli, *DevOps: Continuous Delivery, Integration, and Deployment with DevOps*, 1st Editio. Birmingham: Packt Publishing, 2018.
- [38] Microsoft, "Design the infrastructure persistence layer." [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design>. [Accessed: 20-Jun-2019].
- [39] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.
- [40] M. Fowler, "Notification," 2004. [Online]. Available: <https://martinfowler.com/eaaDev/Notification.html>. [Accessed: 20-Jun-2019].
- [41] G. Myers, C. Sandler, and T. Badgett, *The Art Of Software Testing*, 3rd Editio. John Wiley & Sons, 2012.
- [42] Microsoft, "Unit test basics," 2019. [Online]. Available: <https://docs.microsoft.com/en-us/visualstudio/test/unit-test-basics?view=vs-2019>. [Accessed: 24-Jun-2019].
- [43] D. Graham, E. Veenendall, I. Evans, and R. Black, *Foundations of Software Testing - ISTQB Certification*, 1st Editio. London, United Kingdom: Cengage Learning EMEA, 2006.
- [44] R. Hartson and P. Pyla, *The UX Book*, 1st Editio. Waltham, MA: Elsevier, 2012.
- [45] R. Virzi, "Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?," *Hum. Factors*, vol. 34, 1992.
- [46] J. Piairo, "Desenvolvimento de Interface Inteligente para Suporte à Gestão e Controlo de Produção," Faculdade de Engenharia da Universidade do Porto, 2012.

Anexos

Anexo 1 – Modelo de dados ER



Anexo 2 – Guião de testes



Guião de Testes

Plataforma Educa

OBJETIVO

Este guião tem o objetivo de orientar o processo de testes da aplicação EDUCA. Com a realização destes testes pretende-se analisar a experiência do contato do utilizador com a aplicação. Após a conclusão dos testes, responda ao questionário, clicando [aqui](#).

PRÉ-CONFIGURAÇÕES REALIZADAS

Para a realização destes testes foram realizadas as seguintes pré-configurações na aplicação:

1. Configuração de acessos à plataforma (ver **Erro! A origem da referência não foi encontrada.**);
2. Criação do curriculum²⁴ do 3º ano do curso de Manutenção Industrial, incluindo a configuração das respetivas disciplinas, *outcomes* e turmas. Este curso é usado no **cenário de teste 2**.

Na **Erro! A origem da referência não foi encontrada.** encontram-se listados os utilizadores que foram criados para a realização dos testes. O acesso à aplicação é realizado com os e-mails e respetivas passwords.

²⁴ O curriculum do curso de Manutenção Industrial foi enviado em conjunto com este guião

Cargo	Email	Password
Administrador	admin@educa.com	Admin12#
Diretor de Curso, Professor	pedro.carlos@educa.com	Educa12#
Professor	susana.vasconcelos@educa.com	Educa12#
Professor	tiago.costa@educa.com	Educa12#
Professor	paulo.andrade@educa.com	Educa12#

APRESENTAÇÃO DOS CENÁRIOS DE TESTE

Cenário 1 (exclusivo para o diretor de curso)

O cenário 1 é exclusivo para o Diretor de Curso. Os testes incluídos neste cenário focam-se essencialmente nas funcionalidades exclusivas deste tipo de utilizador, tais como, a gestão de cursos, disciplinas, turmas, *outcomes* e certificados.

Cenário 2 (diretor de curso e professor)

O cenário 2 passa pela simulação do momento de avaliação de uma turma do 3º ano do curso de Manutenção Industrial. A turma em causa trata-se de uma turma de alunos finalistas, o que significa que caso um estudante seja aprovado com sucesso, o certificado de aquisição de competências fica disponível para emissão por parte do Diretor de Curso.

Os testes encontram-se divididos por cargo. Faça os testes correspondentes ao cargo do utilizador que vai usar.

CENÁRIO 1 – DIRETOR DE CURSO

Grupo de testes destinado ao Diretor de Curso

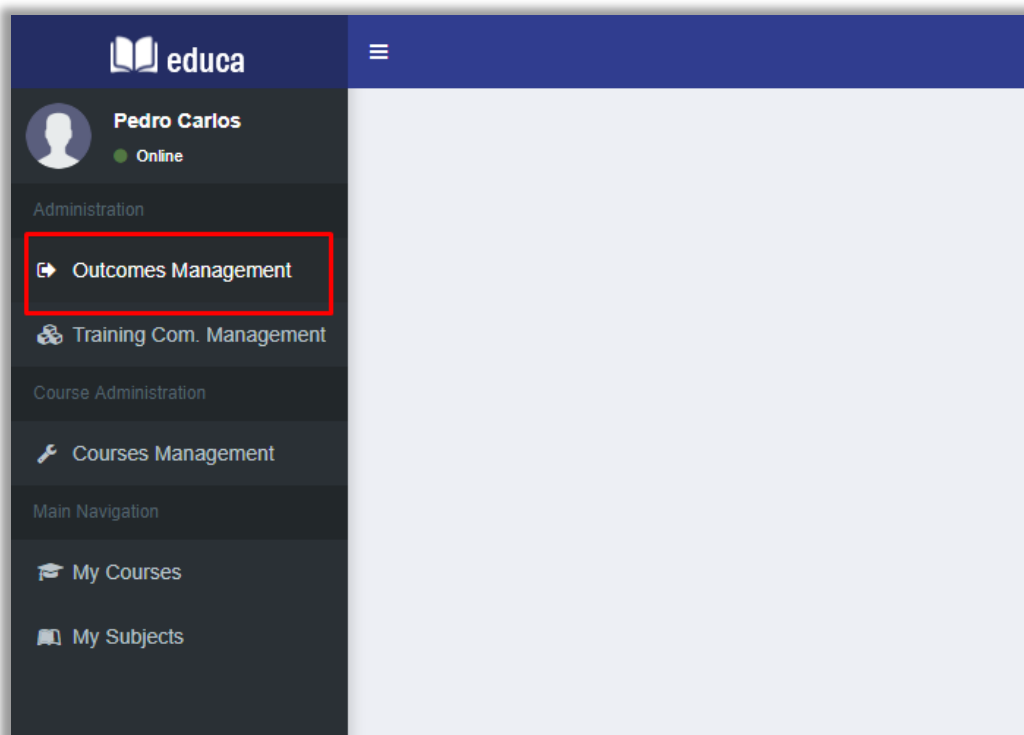
1. Acesso à aplicação

Aceda à aplicação clicando no seguinte endereço:
<https://educagecad.azurewebsites.net>

Assim que a aplicação abrir, faça o *login* com o Diretor de Curso listado na **Erro! A origem da referência não foi encontrada..**

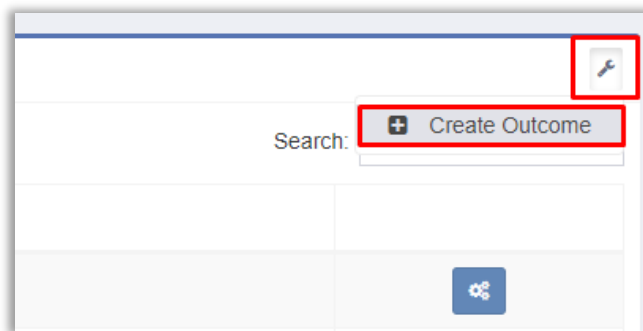
2. Gestão de *Outcomes*

Após o *login*, e através do menu lateral, navegue para a página de gestão de *outcomes*.



Ao entrar na gestão de *Outcomes* são listados todos os *outcomes* já configurados no sistema.

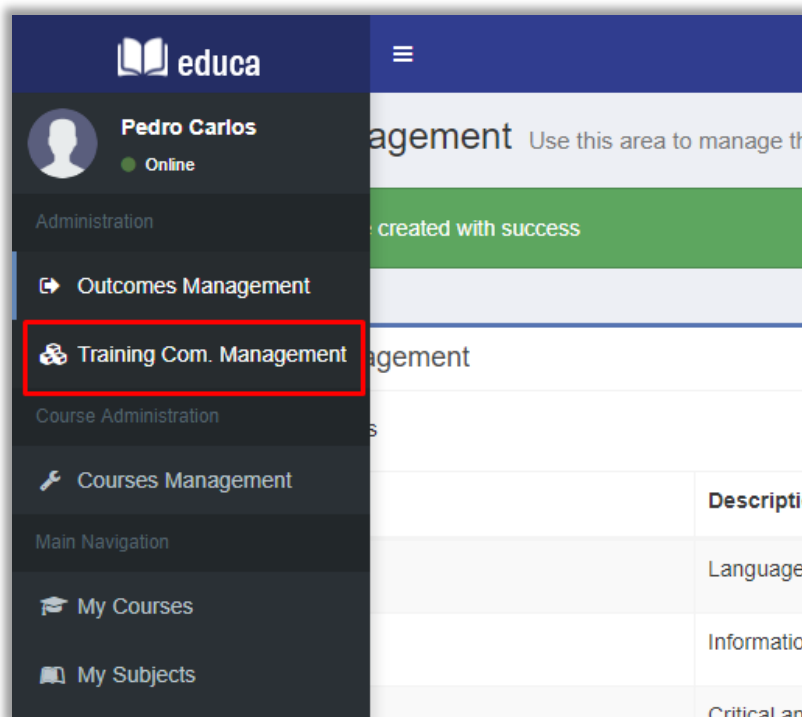
No canto superior direito da listagem, clique no botão para adicionar um novo *outcome*.



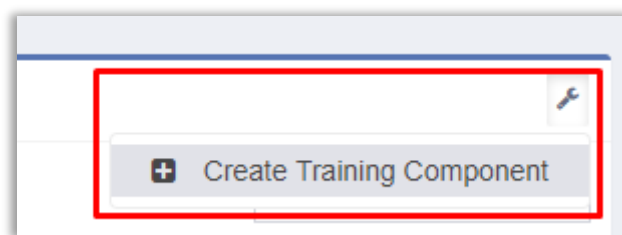
Após o clique irá abrir um formulário de criação de um novo *outcome*. Preencha o formulário e clique no botão “**Save**”.

3. Gestão de Componentes de Ensino

Através do menu lateral, navegue para a página de gestão de componentes de ensino (Training Components).



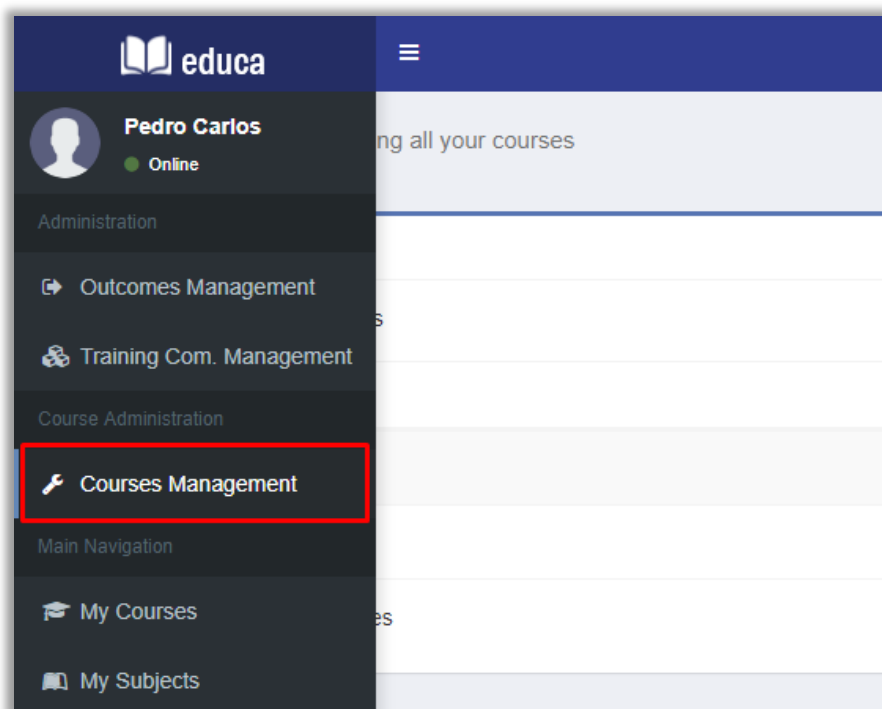
Ao entrar na página são listadas todos as componentes de ensino já configuradas no sistema. No canto superior direito da listagem, clique no botão para adicionar uma nova componente de ensino.



Após o clique irá abrir um formulário de criação de uma nova componente de ensino. Preencha o formulário e clique no botão **“Save”**.

4. Gestão de Cursos

Através do menu lateral, acesse à página de gestão de cursos.

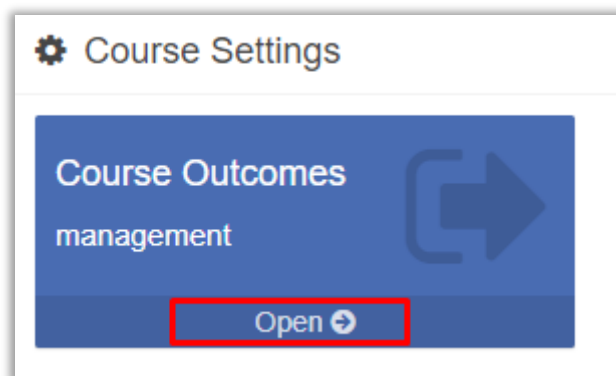


Ao entrar na página são listados todos os cursos em que o utilizador atual é Diretor de Curso. Para o curso **“Test Course”**, clique no botão que se encontra na última coluna da listagem para entrar na área de gestão do curso.

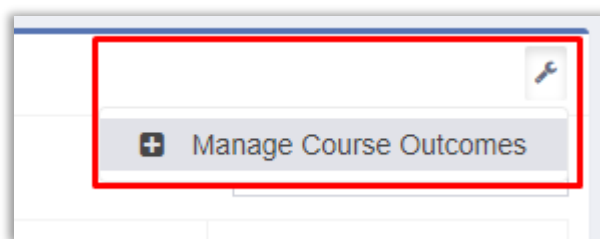


GESTÃO DE *OUTCOMES* DO CURSO

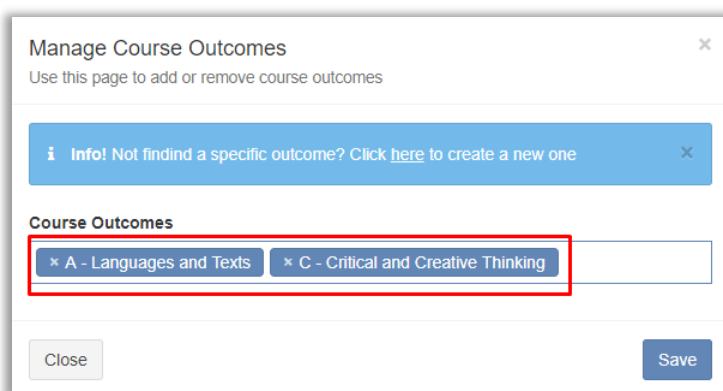
Na área de gestão de curso, nas definições de curso, clique no atalho para abrir a página de gestão de *outcomes* do curso.



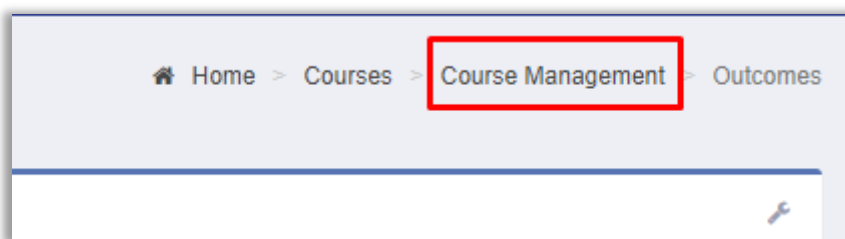
Ao entrar na área de gestão são listados todos os *outcomes* (caso exista algum) que estão associados ao curso. Para associar *outcomes* ao curso, clique no botão presente no canto superior direito da listagem.



Após o clique irá abrir um formulário para associar *outcomes* ao curso. Selecione os *outcomes* desejados e clique no botão “**Save**”.

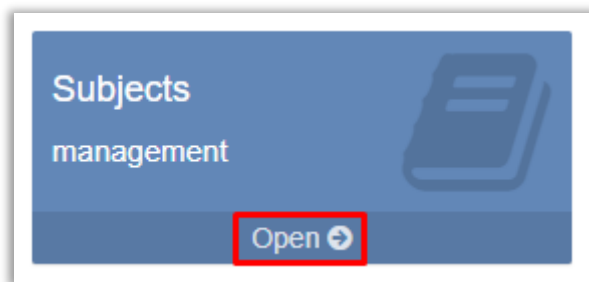


Depois de associados os *outcomes*, no canto superior direito da página, clique em “**Course Management**” para regressar à página de gestão do curso.

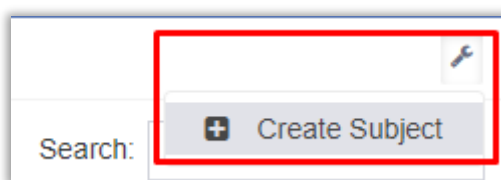


GESTÃO DE DISCIPLINAS

Na área de gestão de curso, nas definições, clique no atalho para abrir a página de gestão de *outcomes* do curso.

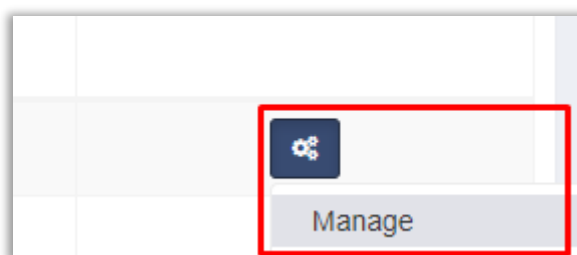


Ao entrar na área de gestão são listadas todas as disciplinas (caso exista alguma) que estão associadas ao curso. Para criar uma disciplina, clique no botão presente no canto superior direito da listagem.

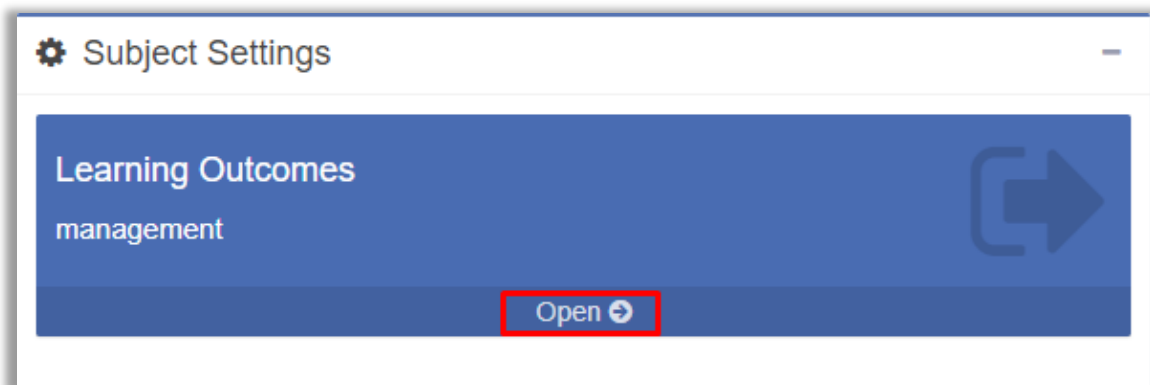


Após o clique irá abrir um formulário para criar a disciplina. Preencha o formulário e clique no botão **“Save”**.

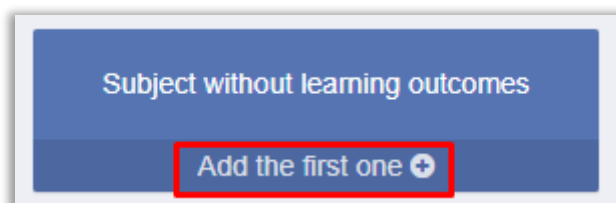
Após a criação da disciplina, esta irá ser listada na listagem de disciplinas do curso. Clique no botão que se encontra na última coluna da listagem para abrir a área de gestão da disciplina.



Na área de gestão da disciplina, nas definições, clique no atalho para abrir a página de gestão de *outcomes* da disciplina.



Visto que se trata de uma nova disciplina, esta ainda não contém nenhum *outcome* criado e, como tal, será apresentada uma mensagem a sugerir a criação do primeiro *outcome*. Clique no botão para avançar com a criação.

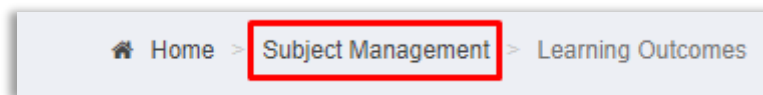


Após o clique irá abrir o formulário de criação de um novo *outcome* para a disciplina. Preencha²⁵ o formulário e clique no botão **“Save”**. Após a criação, o *outcome* é apresentado.

A screenshot of a form for creating an outcome. The form has a blue header bar with the text "Identifies the various types of maintenance as well as their direct and indirect costs" and a minus sign on the right. Below the header, there is a white section with the text "Name Identifies the various types of maintenance as well as their direct and indirect costs". Below this, there are two checkboxes: "Developed in School" (checked) and "Developed at Workplace" (unchecked). Below the checkboxes, there is a section titled "Essential Learnings" with a light blue background. Below this section, there is a text input field containing the text "Knows the work orders and prepares work reports".

Depois de criado o *outcome* da disciplina, no canto superior direito da página, clique em **“Subject Management”** para regressar à página de gestão da disciplina.

²⁵ Pode consultar o curriculum do curso de Manutenção Industrial (fornecido em conjunto com este guia) para preencher o formulário

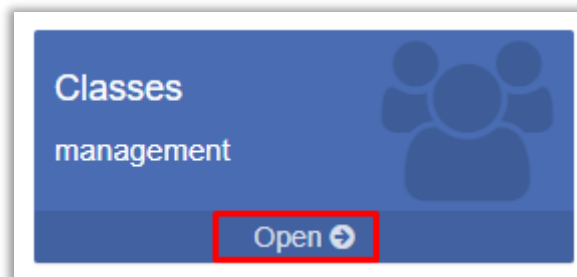


Na área de gestão da disciplina, na secção de informação, clique no nome do curso para regressar à área de gestão do curso.

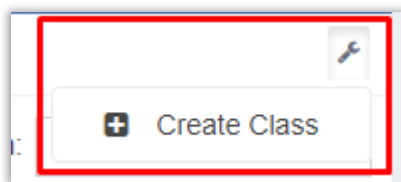
Course	Test Course
Hours	1
Training Component	Scientific

GESTÃO DE TURMAS

Na área de gestão do curso, nas definições, clique no atalho para abrir a página de gestão de turmas.



Ao entrar na área de gestão são listadas todas as turmas (caso exista alguma) que estão associadas ao curso. Para criar uma turma, clique no botão presente no canto superior direito da listagem.



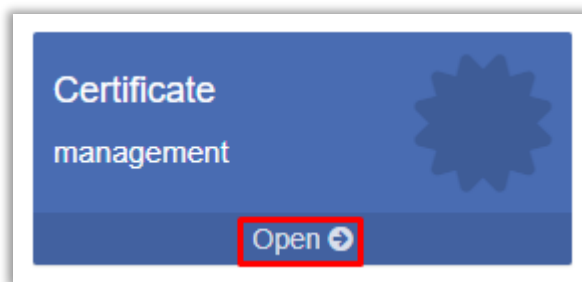
Após o clique irá abrir um formulário para criar a disciplina. Preencha o formulário, indicando o código da turma, o ano do curso em que esta turma se insere, os alunos que a compõem e quais os professores que lecionam as disciplinas para esta turma. Para finalizar, clique no botão “Save”.

Para regressar à página de gestão do curso, clique no botão “Course Management” tal como representado na **Erro! A origem da referência não foi encontrada.**

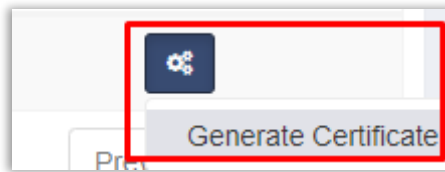
GESTÃO DE CERTIFICADOS

Nota: Para que o certificado seja gerado com as cores corretas é necessária uma configuração adicional no *browser*. Para facilitar este teste sugerimos que o realize no Google Chrome.

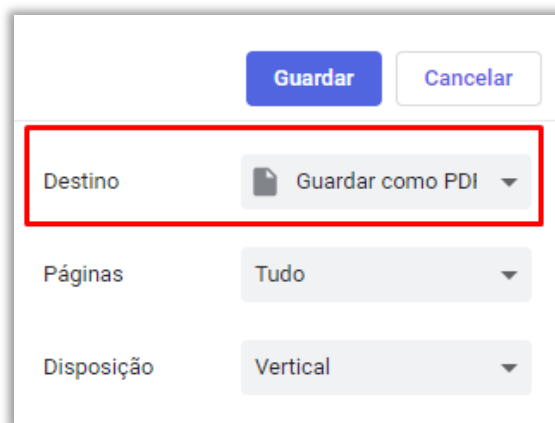
Na área de gestão de curso, nas definições, clique no atalho para abrir a página de gestão de certificados.



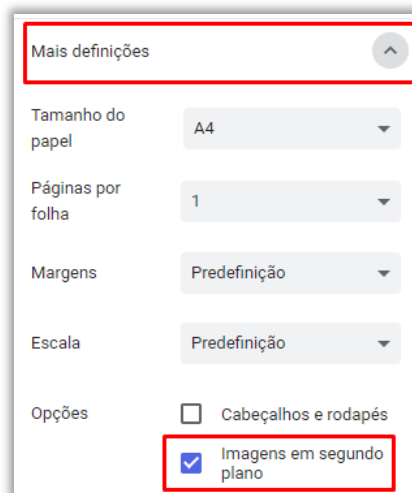
Ao entrar na área de gestão são listados todos os alunos elegíveis para emissão de certificado de aquisição de competências. Para o aluno **Bruno Coutinho Henriques**, clique no botão que se encontra na última coluna da listagem para gerar o certificado.



Após o clique, o certificado é gerado e a janela de impressão abre automaticamente. Na opção “Destino” selecione “Guardar como PDF”.



Caso algumas cores estejam em falta, clique em “Mais definições” e selecione a opção “Imagens em segundo plano”.



Clique em “Guardar”. O PDF é gerado e guardado no local indicado.



Skills Certificate | Industrial Maintenance

Summary

Student Number	Student Name	Final Grade	
3	Bruno Coutinho Henriques	16,02	Approved

Student Grades by year

2018/2019	Grade 16,02
-----------	---------------

Technology & Processes

Grade 16,87

Recognizes the operating principle of electric machines

Learning Outcome Grade 18,33

Developed in school

Essential Learning Name	Grade
Knows and interprets the connection schematics of electric machines	20,00
Chooses the electric machine according to the intended purpose	18,00
Makes right associations	17,00

CENÁRIO 2 – DIRETOR DE CURSO E PROFESSOR

Grupo de testes destinado ao Diretor de Curso e Professor

1. Acesso à aplicação

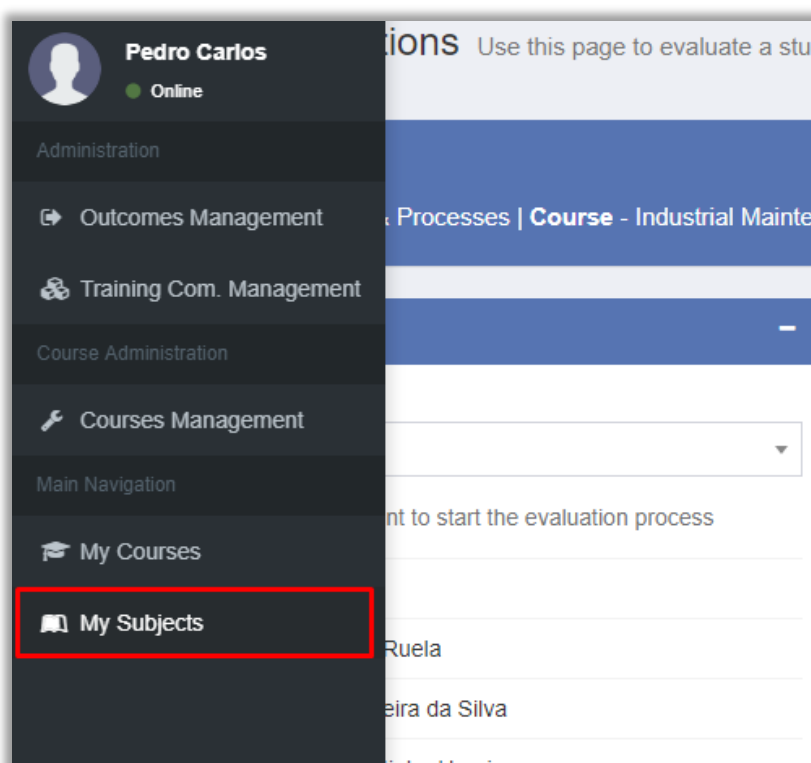
Aceda à aplicação clicando no seguinte endereço:

<https://educagecad.azurewebsites.net>

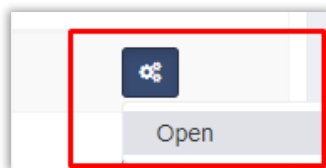
Assim que a aplicação abrir, faça o *login* com um dos utilizadores listados na **Erro! A origem da referência não foi encontrada..**

2. Avaliar aluno

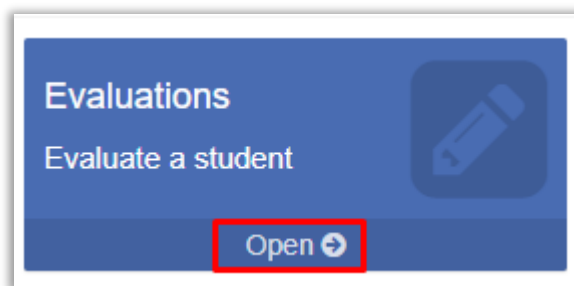
Após o *login*, e através do menu lateral, navegue para a página “**My Subjects**”.



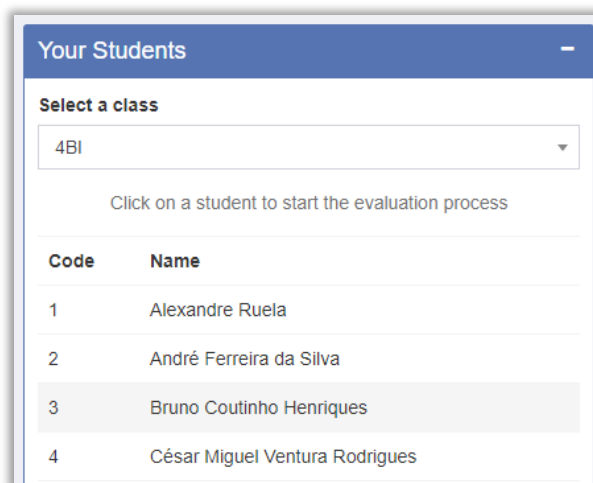
Ao entrar na página são listadas todas as disciplinas que o utilizador atual leciona. Numa disciplina à sua escolha, clique no botão que se encontra na última coluna da listagem para abrir a disciplina.



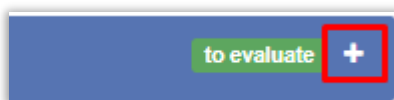
Depois de abrir a área da disciplina, na secção de ações, clique no atalho para abrir a página de avaliações de alunos.



Depois de abrir a página de avaliações, na secção “**My Students**”, selecione uma turma. Depois de selecionada a turma são apresentados todos os alunos dessa turma. Selecione o aluno que pretende avaliar.



Ao selecionar um aluno é carregado o formulário de avaliação na secção “**Evaluation Form**”. Clique no botão (+) para abrir o *outcome*.



Depois de aberto, preencha a coluna “**Grade**” com as notas do aluno.

Grade
20
18
17

Para concluir a avaliação, **avalie o aluno face a todos os *outcomes* que são listados.**

Assim que preencher todas as notas, clique no botão “**Save**”. Após a gravação da avaliação, verifique que a nota final do aluno foi calculada.

Evaluation created at	Final Grade
02/07/2019 23:29:49	16,87

Os testes terminaram. Por favor, responda ao questionário final, clicando [aqui](#).

Obrigado pela sua colaboração.

Anexo 3 – Questionário

Inquérito de satisfação

Este questionário é realizado de forma anónima.

Com este inquérito pretende-se recolher a opinião sobre a experiência do contato do utilizador com a aplicação Educa.

*Obrigatório

Secção sem título

1. Cargo na instituição *

Marcar tudo o que for aplicável.

- Administrator de Sistema
- Diretor de Curso
- Professor
- Orientador de empresa
- Outra: _____

2. Experiência de utilização *

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Concordo	Concordo Totalmente	Sem Opinião
A aplicação apresenta a informação de forma organizada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A aplicação apresenta um aspeto gráfico agradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A interação com a aplicação é intuitiva e de fácil manuseamento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A navegação entre ecrãs é simples e consistente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A interface da aplicação contribuiu para a sua motivação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O registo da informação é mais simples	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O registo da informação é mais rápido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A aplicação irá facilitar o meu trabalho diário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A aplicação atende satisfatoriamente às minhas necessidades	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A aplicação tem um bom comportamento em dispositivos móveis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. Funcionalidades de administração

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Concordo	Concordo Totalmente	Sem Opinião
O processo de criação de utilizadores é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de criação de cursos é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de criação de Outcomes é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de criação de Áreas de Disciplinas (Training Components) é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A funcionalidade de carregamento de alunos em massa é útil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Funcionalidades de gestão de curso

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Concordo	Concordo Totalmente	Sem Opinião
O dashboard de gestão de curso contém a informação necessária	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consigo navegar facilmente entre as diferentes funcionalidades de gestão de um curso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de criação de uma disciplina é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de criação de uma turma é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A funcionalidade de listagem de alunos elegíveis para emissão de certificado é útil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O processo de geração de um certificado é simples	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O certificado tem um aspeto agradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O certificado tem a informação necessária	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Funcionalidades de gestão de disciplinas

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Concordo	Concordo Totalmente	Sem Opinião
O processo de criação de um Outcome de uma disciplina (Learning Outcome) é simples e intuitivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A visualização de outcomes de uma disciplina é agradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Avaliação de alunos

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Concordo	Concordo Totalmente	Sem Opinião
O processo de avaliação de alunos é simples	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consigo facilmente encontrar o aluno que pretendo avaliar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O formulário de avaliação contém a informação necessária	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Avaliação Final


Marcar apenas uma oval por linha.

	Muito Mau	Mau	Razoável	Bom	Muito Bom
Em geral, como avalia o nível da aplicação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Deixe-nos uma recomendação

Obrigado pela sua participação!

Anexo 4 – Excerto de um certificado de aquisição de competências

 Skills Certificate Industrial Maintenance			
Summary			
Student Number	Student Name	Final Grade	
3	Bruno Coutinho Henriques	16,02	Approved
Student Grades by year			
2018/2019			Grade 16,02
Technology & Processes			
Grade 16,87			
Recognizes the operating principle of electric machines			
Learning Outcome Grade 18,33			
Developed in school			
Essential Learning Name	Grade		
Knows and interprets the connection schematics of electric machines	20,00		
Chooses the electric machine according to the intended purpose	18,00		
Makes right associations	17,00		
Recognizes the basic principles of pneumatics and hydraulics			
Learning Outcome Grade 18,00			
Developed in school			
Essential Learning Name	Grade		
Knows and executes the assembly of pneumatic / hydraulic circuits	18,00		
Completes tasks on time	17,00		
Reveals ease in maintaining an adequate working pace	19,00		