



Plataforma de Gestão da Formação

CÁTIA PATRÍCIA SOUSA PINTO

Julho de 2020

Plataforma de Gestão da Formação

Cátia Patrícia Sousa Pinto

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

Orientador: Alberto Sampaio

Supervisor: Paulo Matos

Porto, julho 2020

Dedicatória

«Dedico este documento aos meus pais que com muito carinho e apoio, não mediram esforços para que chegasse a esta etapa da minha vida. Um obrigado à minha mãe nunca será suficiente. Contigo aprendi que o ontem não é igual ao hoje e que a perseverança tem o efeito mágico de fazer as dificuldades e os obstáculos desaparecerem. Contigo aprendi a nunca desistir. Obrigada ao meu pai que dignamente me mostra a importância da família e o caminho da honestidade e humildade. Contigo aprendi de modo nobre a dignificar e aceitar as minhas escolhas. Por último, obrigada ao meu avô materno, uma estrela que olha por mim independentemente de onde me encontrar...»

Resumo

Há medida que os profissionais de saúde crescem em escala internacional, paralelamente, as formações em hospitais e sistemas de saúde são cada vez mais importantes para que o atendimento aos pacientes seja rápido e irrepreensível.

Essa evolução, para organizações cujos produtos se inserem na área da saúde, passa pelas plataformas de gestão de formação, e é nessa ótica de vanguarda que surge o tema para este projeto, uma vez que, a grande preocupação da ALERT é assegurar uma formação de qualidade, ágil e com processos de formação estruturados para os seus colaboradores e clientes em relação aos seus produtos.

As plataformas de gestão da formação são desenhadas para centralizar os processos de formação de uma organização, permitindo de forma rápida supervisionar os mesmos sem que exista necessidade de interpolar entre ferramentas e plataformas diferentes.

Essas plataformas permitem ainda o acompanhamento, planeamento e a monitorização de grandes quantidades de cursos de uma organização, impedindo a perda de informação. Para além disso, diminui o erro e poupa tempo às equipas de uma empresa na programação de formações para várias instituições, uma vez que, podem consultar a disponibilidade de salas de aulas, monitorizar inscrições e acompanhar esses mesmos planos de formação de forma rápida, segura e atualizada.

Palavras-chave: ALERT, Plataforma, Formação, Profissionais de Saúde, e-learning, Java, Spring, REST

Abstract

As healthcare professionals grow in importance internationally, so does the importance of training of this professionals in terms of IT systems, so everyone can benefit from it, from patients to every involved in their treatment.

That evolution, for organizations whose products fit in the health area, goes through training management platforms, and it is in that forefront optic that the subject for this project comes up, as the biggest concern for ALERT is to ensure an agile high quality training, with structured processes for its employees and clients regarding company's products.

This type of platforms of training management are designed to centralize the training processes of any organization, allowing to control this in a much faster way without the need to "jump" in between different platforms.

This platforms also allow the planning, follow up, and monitoring of great volumes of different types of training without losing any data. Besides it lowers the risk of errors and saves time to the training teams of any company when planning their content, given that they can consult the availability of class rooms, monitoring sign-ups and follow-up all this planning in a faster and safer way.

Keywords: ALERT, Platform, Training, Healthcare Professionals, e-learning, Java, Spring, REST

Agradecimentos

Gostava de começar por agradecer à empresa ALERT a quem estarei eternamente grata pela oportunidade e voto de confiança para desenvolver uma dissertação de mestrado sobre um tema que tanto aprecio. Um agradecimento especial à minha equipa, nomeadamente, ao meu supervisor Paulo Matos por toda a paciência, rigor, profissionalismo e total disponibilidade na colaboração do desenvolvimento deste documento, bem como ao meu colega, David Trigueira, pela disponibilidade e por todos os ensinamentos. Acima de tudo, obrigada à ALERT pelo ambiente incrível e por me ajudar a tornar numa melhor profissional e numa melhor pessoa.

Um agradecimento especial, ao Professor Alberto Sampaio, pela sua orientação, apoio, disponibilidade, paciência, pelas opiniões e críticas construtivas e colaboração nas dúvidas que foram surgindo na realização deste documento.

Seguidamente, agradecer ao Instituto Superior de Engenharia do Porto (ISEP) por ter contribuído para uma segunda etapa da minha vida académica.

Aproveito também para agradecer a todos os meus amigos que de direta ou indiretamente contribuíram para que este percurso fosse disfrutado da melhor forma possível. Guardo em mim, memórias de um caminho feliz a vosso lado.

Um agradecimento muito, muito especial, à minha melhor amiga, Mariana Lei, pelo apoio incondicional durante o meu percurso académico. Palavras serão sempre pequenas para grandes atitudes.

Por fim, não podia terminar sem um sentido agradecimento aos meus pais e à minha irmã que nunca mediram esforços para que chegasse aqui. Pela paciência, pelo tempo, pelas conversas, pelas memórias, pelos ensinamentos... esta dissertação é também a realização de um sonho vosso.

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema	3
1.3	Objetivos	4
1.4	Partes Interessadas	5
1.5	Motivação Pessoal	6
1.6	Planeamento do Projeto	7
1.7	Contribuições e Valor	9
1.8	Estrutura do Documento	10
2	Estado da Arte	13
2.1	Educação e Tecnologia	13
2.2	O E-learning Empresarial	14
2.3	Sistemas de Gestão de Formação	16
2.3.1	Qual a sua Importância no Setor Empresarial?	16
2.3.2	Que Funcionalidades Suporta um TMS?	16
2.3.3	Quais são os Grupos de Utilizadores de um TMS?	17
2.3.4	FORINSIA	17
2.4	Conceitos	18
2.4.1	Web Services	18
2.4.2	API	19
2.4.3	Arquitetura REST	19
2.4.4	REST API	19
2.4.5	Training Plan/Learning Plan	20
2.4.6	Learning Program	20
2.5	Arquiteturas de Sistemas de Gestão de Aprendizagem baseados na Cloud	20
2.5.1	TalentLMS	21
2.5.2	Intuo LMS	23
2.5.3	PowerSchool Student Management System (SIS)	26
2.5.4	Adobe Captivate Prime	29
2.5.5	Comparações e Conclusões sobre os SGA Analisados	33
2.6	User Interface e Funcionalidades de SGA e TMS	33
2.6.1	Barra de Navegação	33
2.6.2	Página Inicial com Perfil de Administrator	34
2.6.3	Gestão de Inscrições	36
2.7	Comparações e Conclusões sobre Funcionalidades	36
3	Análise de Valor	39
3.1.1	Orientação	40

3.1.2	Identificação e Análise Funcional	48
3.1.3	Criação de Alternativas	54
3.1.4	Análise e Avaliação	55
3.1.5	Implementação	70
4	Desenvolvimento da Solução	71
4.1	Grupos de Utilizadores e suas Características	71
4.2	Vista de Casos de Uso	72
4.3	Requisitos Não Funcionais	77
4.4	Modelo de Domínio.....	79
4.5	Vista de Dados.....	81
4.6	Interface Gráfica.....	85
4.6.1	Cabeçalho	85
4.6.2	Rodapé.....	86
4.6.3	Barra de Navegação.....	86
4.6.4	Vista de Detalhes de uma Instituição	87
4.6.5	Vista Principal de Cursos	88
4.6.6	Vista de Detalhes de um Curso	89
4.6.7	Criação de um <i>Training Plan</i>	90
4.6.8	Vista de Detalhes de um <i>Training Plan</i>	91
4.6.9	Criação de uma Sessão	92
5	Arquitetura	94
5.1	Diagrama de Componentes	94
5.2	Objetivos e Restrições Arquiteturais	95
5.3	Tecnologias e Processos de Desenvolvimento.....	96
5.3.1	Tecnologias Seleccionadas	96
5.3.2	Justificação	96
5.4	Vista Lógica	96
5.5	Vista de Implantação	98
6	Realização de Casos de Uso de maior Relevância	101
6.1	Autenticação e Autorização	101
6.2	Visualizar Lista de Utilizadores da Plataforma.....	108
6.3	Criar Instituição	112
6.4	Criar Curso	115
6.5	Visualizar Cursos	118
6.6	Visualizar Detalhes de <i>Training Projects</i>	121
6.7	Criar <i>Training Plan</i>	124
7	Testes.....	129

7.1	Comparação entre Testes Unitários e de Integração.....	129
7.2	Contextualização.....	129
7.3	Testes Unitários	130
7.3.1	Controladores.....	131
7.3.2	Serviços.....	132
7.3.3	Repositórios.....	134
7.4	Testes de Integração.....	135
8	Experimentação e Avaliação	137
8.1	Objetivo	138
8.2	Metodologia	138
8.3	Execução	140
8.4	Resultados.....	140
8.5	Conclusão.....	143
9	Conclusão	145
9.1	Objetivos Alcançados	145
9.2	Limitações e Trabalho Futuro	146
9.3	Apreciação Final.....	146
10	Referências.....	147
11	Anexo A - Questionário de Usabilidade	153

Lista de Figuras

Figura 1 – Visão do problema/desafio proposto	2
Figura 2 – Tamanho do Mercado do Sistema de Gestão de Aprendizagem, 2018-2026 (US\$ Milhões) (Arterburn, 2019)	15
Figura 3 - Chamada API. Adaptado de Filipe e Ricardo (2018).	20
Figura 4 - Pedido REST para <i>user</i> com id = 1. Adaptado de Filipe e Ricardo (2018) e TalentLMS API (TalentLMS, 2020)	22
Figura 5 – Resposta HTTP em JSON dos dados da <i>skill</i> com id = 63081 (INTUO).....	24
Figura 6 - Resposta HTTP em JSON dos dados do report do user com id = 1817 (INTUO)	25
Figura 7 – Resposta HTTP em JSON com dados dos grupos de utilizadores da plataforma (INTUO)	26
Figura 8 – Visão geral da arquitetura do PowerSchool SIS (PowerSchool).....	27
Figura 9 – Arquitetura da aplicação Adobe Captivate Prime (Adobe, 2017)	30
Figura 10 – Fluxo de dados da Adobe Captivate Prime (Adobe, 2017)	32
Figura 11 – <i>Navbar</i> TalentLMS (Pappas, TalentLMS Overview).....	34
Figura 12 - <i>Navbar</i> FORINSIA de um administrador.....	34
Figura 13 – Homepage de um administrador TalentLMS (Pappas, TalentLMS Overview)	35
Figura 14 – Homepage de um administrador Adobe Captivate Prime (Pappas, Adobe Captivate Prime).....	35
Figura 15 - Gestão de inscrições na plataforma FORINSIA	36
Figura 16 – Processo Análise de Valor. Adaptado de Nick Rich e Matthias Holweg (Rich & Holweg, 2000)	40
Figura 17 – <i>New Concept Development</i> (Koen, 2001).....	41
Figura 18 – <i>Customer Chain</i>	49
Figura 19 – Casa de Qualidade	51
Figura 20 – <i>Pairwise Comparison</i> : Gestão da Formação	54
Figura 21 – Árvore de divisão hierárquica	56
Figura 22 – Árvore de divisão hierárquica com valores associados.....	69
Figura 23 - Diagrama de casos de uso - <i>Administrator</i>	73
Figura 24 - Modelo de Domínio do sistema.....	80
Figura 25 – Diagrama de Entidade-Relação	82
Figura 26 - Cabeçalho para a nova plataforma	85
Figura 27 - Rodapé para a nova plataforma.....	86
Figura 28 - Barra de navegação para a nova plataforma	86
Figura 29 - Comportamento suportado pelo ícone na barra de navegação, à direita para a nova plataforma.....	87
Figura 30 - Comportamento suportado pela <i>dropdown</i> com o texto <i>Add</i> para a nova plataforma.....	87
Figura 31 – Vista de detalhes de uma instituição para a nova plataforma.....	88
Figura 32 - Vista principal de Cursos	89
Figura 33 – Vista de detalhes de um curso para a nova plataforma.....	90

Figura 34 - Criação de um Training Plan.....	91
Figura 35 – Vista de detalhes de um training plan para a nova plataforma	92
Figura 36 – Criação de uma sessão para a nova plataforma.....	93
Figura 37 – Diagrama de componentes	94
Figura 38 – Vista Lógica da plataforma	97
Figura 39 – Diagrama de implantação.....	99
Figura 40 – Spring Security - Arquitetura no processo de Autenticação e Autorização JWT. Adaptado de (grokonez, 2020).....	102
Figura 41 - Diagrama de sequência com o fluxo de execução do UC7.....	109
Figura 42 – Página com código HTML para o UC7.....	111
Figura 43 – Diagrama de sequência com o fluxo de execução do UC16.....	112
Figura 44 – Página com código HTML para o UC16.....	115
Figura 45 - Ciclo de vida típico de um curso. Adaptado de (Courses, 2020).....	115
Figura 46 - Diagrama de sequência com o fluxo de execução para o UC3	116
Figura 47 – Página com código HTML para o UC3.....	118
Figura 48 – Diagrama de sequência com o fluxo de execução para visualizar cursos	119
Figura 49 – Página com código HTML que lista cursos para <i>Trainees</i> e <i>Team Managers</i>	120
Figura 50 – Página com código HTML que lista cursos para <i>Administrators</i> , <i>Local Coordinators</i> , <i>Training Managers</i> e <i>Trainers</i>	121
Figura 51 - Diagrama de sequência com fluxo de execução para o UC20	122
Figura 52 – Página com código HTML que exhibe a informação relativa a um <i>Training Project</i>	124
Figura 53 - Diagrama de sequência e fluxo de interação para o UC12	125
Figura 54 – Página com código HTML que permite ao utilizador criar um novo <i>training plan</i>	127
Figura 55 - Resultados da execução dos testes unitários e integração.....	130
Figura 56 - Declaração de <i>mocks</i> para os testes unitários da classe "TrainingProjectController"	131
Figura 57 - Teste unitário que permite obter um <i>training project</i> através do seu id	132
Figura 58 – Declaração de <i>mocks</i> para os testes unitários da classe "InstitutionService"	132
Figura 59 – Teste unitário que permite criar uma instituição.....	133
Figura 60 – Configuração da classe "TrainingPlanRepository"	134
Figura 61 – Teste unitário que permite obter os metadados de <i>training plans</i> através do id do <i>training project</i> pedido.....	134
Figura 62 - Declaração de <i>mocks</i> para os testes de integração da classe "SessionMeetingController"	135
Figura 63 – Teste de integração que permite obter todas as sessões existentes na base de dados	136
Figura 64 - Escala de Likert. Adaptado de Sam, Andrew e Camille (2012).....	139
Figura 65 – Inserção das respostas dos inquiridos no Excel.....	142
Figura 66 – Resultados obtidos da pontuação SUS.....	142

Lista de Tabelas

Tabela 1 - Principais funcionalidades da plataforma FORINSIA.....	18
Tabela 2 – Dados Pedido REST para <i>user</i> com id = 1	21
Tabela 3 – Propriedades relevantes.....	23
Tabela 4 – Web <i>Services</i> estudados através da Intuo API.....	24
Tabela 5 – Propriedades relevantes do objeto da Figura 6	25
Tabela 6 – Funcionalidades oferecidas pelos sistemas de gestão de aprendizagem e sistema de gestão de formação	36
Tabela 7 – Benefícios e Sacríficos. Adaptado de Woodall (2003).....	44
Tabela 8 – <i>Perceived Value</i>	46
Tabela 9 – Escala fundamental – Níveis de importância de comparações (Nicola, 2019).....	57
Tabela 10 – Comparação de prioridades dos critérios (matriz de prioridades).....	57
Tabela 11 – Somatório das colunas de cada critério	58
Tabela 12 – Comparação de prioridades dos critérios normalizados.....	58
Tabela 13 – Prioridade dos critérios	59
Tabela 14 - Valores IR definidos pelo Laboratório Nacional de Oak Ridge (Nicola, 2019)	59
Tabela 15 – Matriz de prioridades das alternativas para o critério suporte.....	61
Tabela 16 – Somatório de cada coluna das alternativas do critério suporte	61
Tabela 17 – Valores normalizados das alternativas do critério suporte.....	62
Tabela 18 – Prioridades das alternativas do critério suporte	62
Tabela 19 – Matriz prioridades das alternativas para o critério migração de dados	63
Tabela 20 – Somatório de cada coluna das alternativas do critério migração de dados	63
Tabela 21 – Valores normalizados das alternativas do critério migração de dados.....	64
Tabela 22 – Prioridades das alternativas do critério migração de dados	64
Tabela 23 – Matriz de prioridades das alternativas para o critério customização e personalização	65
Tabela 24 – Somatório de cada coluna das alternativas do critério customização e personalização	65
Tabela 25 – Valores normalizados das alternativas do critério customização e personalização	66
Tabela 26 – Prioridades das alternativas do critério customização e personalização.....	66
Tabela 27 – Matriz de prioridades das alternativas para o critério integração e utilização.....	67
Tabela 28 – Somatório de cada coluna das alternativas do critério integração e utilização.....	67
Tabela 29 – Valores normalizados das alternativas do critério integração e utilização	68
Tabela 30 – Prioridades das alternativas do critério integração e utilização	68
Tabela 31 - Tabela que apresenta as permissões dos restantes grupos de utilizadores para os casos de uso expostos.....	73
Tabela 32 - Descrição das entidades do modelo de domínio	80
Tabela 33 - Catálogo dos elementos do diagrama de Entidade-Relação.....	82
Tabela 34 - Item original vs item correspondente em Português Europeu (Martins, Rosa, Queirós, Silva, & Rocha, 2015).....	139

Tabela 35 - Dados das respostas do questionário.....	141
Tabela 36 - Valores de referência para interpretação do resultado do SUS (McLellan, Muddimer, & Peres, 2012)	142

Lista de Códigos

Código 1 – Método de validações e recolha de dados do <i>token</i> de acesso.....	103
Código 2 - Criação do CustomUserDetails com dados obtidos do LDAP e da base de dados .	104
Código 3 - Criação e adição do <i>token</i> e da <i>thumbnail</i> ao cabeçalho da resposta.....	105
Código 4 – Método de validação e recolha de dados do <i>token</i> de acesso	106
Código 5 - Processo de autorização	107
Código 6 - Configurações de segurança de acesso a <i>endpoints</i> da API	108
Código 7 – Implementação do método que identifica a <i>role</i> do utilizador e encaminha para um dos dois cenários possíveis	110
Código 8 - Implementação do método para obter as instituições que pertencem a um utilizador com permissão <i>Local Coordinator</i>	110
Código 9 - Implementação do método que permite obter uma instituição pelo seu id	111
Código 10 – Implementação do método que permite obter os utilizadores da plataforma...	113
Código 11 – Implementação do método que permite validar o formulário e encaminhar o pedido para o respetivo serviço na camada de apresentação	114
Código 12 - Implementação do método responsável por verificar se já existe um curso com o mesmo nome	117
Código 13 - Método do controlador de cursos que recebe o pedido de adição de curso.....	118
Código 14 – Implementação do método de obtenção da lista de cursos com o estado passado por parâmetro.....	120
Código 15 – Implementação do método que chama o serviço de <i>training projects</i> na camada de apresentação de modo a espoletar o pedido HTTP à API.....	123
Código 16 – Implementação do método responsável por retornar a lista de training plans através do id do training project.....	123
Código 17 - Método do controlador de <i>training plans</i> que recebe o pedido de adição	126

Acrónimos e Símbolos

Lista de Acrónimos

RH	Recursos Humanos
SGA	Sistema de Gestão de Aprendizagem
LMS	<i>Learning Management System</i>
API	<i>Application Programming Interface</i>
TMS	<i>Training Management System</i>
CMS	<i>Course Management System</i>
REST	<i>Representational State Transfer</i>
JSON	<i>JavaScript Object Notation</i>
URI	<i>Uniform Resource Identifier</i>
CRUD	<i>Create, Read, Update, Delete</i>
HTTP	<i>HyperText Transfer Protocol</i>
SIS	<i>Student Information System</i>
UI	<i>User Interface</i>
MVC	<i>Model-View-Controller</i>
OLTP	<i>Online Transaction Processing</i>
OLAP	<i>Online Analytical Processing</i>
ODBC	<i>Open Database Connectivity</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
SSL	<i>Secure Sockets Layer</i>
TLS	<i>Transport Layer Security</i>
AWS	<i>Amazon Web Services</i>
VE	<i>Value Engineering</i>
VA	<i>Value Analysis</i>

FFE	<i>Fuzzy Front End</i>
NPD	<i>New Product Development</i>
NCD	<i>New Concept Development</i>
SI	Sistemas de Informação
QFD	<i>Quality Function Deployment</i>
AHP	<i>Analytic Hierarchy Process</i>
RC	Razão de Consistência
IC	Índice de Consistência
RNF	Requisito Não Funcional
JPA	<i>Java Persistence API</i>
MVVM	<i>Model, View, ViewModel</i>
DTO	<i>Data Transfer Object</i>
AD	<i>Active Directory</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
JWT	<i>JSON Web Token</i>
CORS	<i>Cross-Origin Resource Sharing</i>

Lista de Símbolos

λ	<i>Lambda</i>
®	Marca registrada

1 Introdução

O propósito deste capítulo é caracterizar o projeto, descrevendo com brevidade o contexto, o problema, os objetivos para a resolução do mesmo, as partes interessadas e a motivação pessoal. Para além disso, é descrito o planeamento do projeto, a contribuição e valor do trabalho e, por último, a estrutura do documento.

1.1 Contexto

“Fundada em 1999, a ALERT Life Sciences Computing (ALERT) dedica-se totalmente à criação de uma solução integrada para a total informatização dos cuidados de saúde em países inteiros, HMO, hospitais e centros de saúde, passando pelos processos individuais de cada paciente.” (Computing, 2019).

“Apto para a Web e *Cloud*, as soluções ALERT® constituem variantes de configurações de um único produto, com apenas uma linha de desenvolvimento caracterizada pela sua capacidade de configuração local em cada mercado. Estas características permitem à ALERT concentrar os seus esforços num produto universal que se encontra em constante mudança, sendo enriquecido por contribuições de todo o mundo.” (Computing, 2019).

Este projeto nasce da constante preocupação da ALERT em assegurar uma adequada e eficiente formação aos utilizadores dos seus produtos, otimizando os processos de gestão inerentes. Assim, existiu necessidade de desenhar uma solução tudo-em-um de forma a criar um sistema que atenda às necessidades do seu público em movimento, com ramificação ou integração dos produtos já existentes da ALERT® eLEARNING. A Figura 1 apresenta essa ramificação.

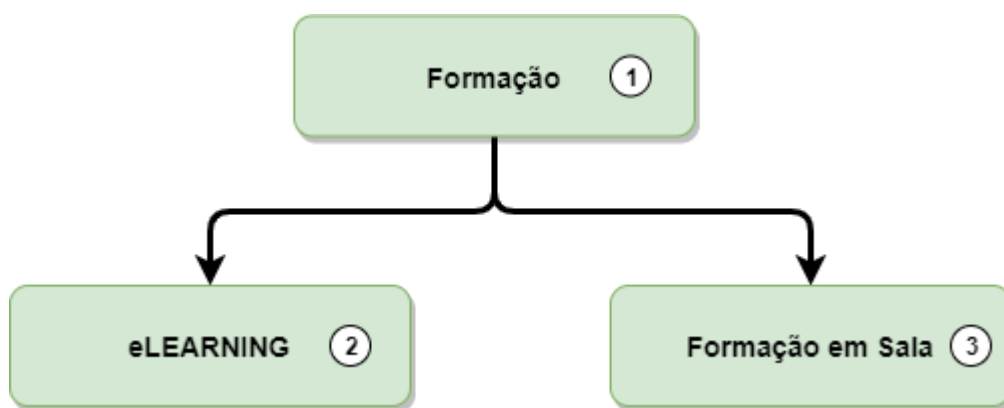


Figura 1 – Visão do problema/desafio proposto

No que respeita à formação, existe a formação em sala e o e-learning. O cenário “eLEARNING” (representado pelo retângulo com o número 2) refere-se ao produto ALERT® eLEARNING, mais comumente designado por Sistema de Gestão de Aprendizagem (SGA) – do inglês Learning Management System (LMS), que oferece aos utilizadores uma formação completa em diversos produtos ALERT®, por intermédio de uma plataforma *web* e de uma interface intuitiva.

Uma plataforma *web* é um portal que expõe funcionalidades com o propósito de servir a empresa. O termo interface intuitiva é aplicado quando os utilizadores compreendem o comportamento e efeito da mesma sem experiência ou treino prévio.

Assim, e similarmente ao produto principal ALERT®, este projeto (cenário “Formação em sala” representado pelo retângulo com o número 3) emerge da necessidade de acompanhamento na instrução dos seus colaboradores e clientes, uma vez que, a formação sobre o produto clínico é importante para o modelo de negócio da empresa. Adicionalmente, a formação não necessita de interagir, forçosamente, com produtos ALERT, como por exemplo, formação em ferramentas ou linguagens de programação, como complemento ao cenário “eLEARNING”.

O cenário “eLEARNING” já se encontra informatizado, isto é, a plataforma em si já responde àquilo que é a gestão de aprendizagem desenvolvida num contexto de e-learning. O sistema de gestão de aprendizagem que a ALERT apresenta aos seus clientes e colaboradores responde às necessidades daquilo que por definição é um SGA, no entanto, relativamente à formação em sala, a ALERT não possui esse produto, ou seja, não existe informatização e, portanto, o desafio desta tese surge para colmatar essa carência, informatizando não só a formação em sala, mas também uma modalidade aprofundada do que acontece para o cenário representado pelo retângulo número 3.

Face ao exposto nos parágrafos anteriores, a plataforma a desenvolver não é um sistema concorrente com o e-learning, isto é, um Sistema de Gestão de Aprendizagem, mas sim uma Plataforma de Gestão da Formação. Este projeto colocado pela ALERT, foca-se no estudo e desenvolvimento de uma solução *enterprise* baseada nos princípios da Web 2.0. A formação em sala, a gestão manual de informação que exige mais tempo comparativamente com o despendido através de uma gestão informatizada, a necessidade de acompanhamento dos

colaboradores não por uma questão de pressão, mas efetivamente por qualidade futura da empresa e, conseqüentemente, do produto, são motivos que influenciam o desenvolvimento de uma plataforma como esta.

1.2 Problema

Paralelamente à formação realizada através do ALERT® eLEARNING, as metodologias de formação em sala continuam a ter uma presença bastante significativa nos projetos da ALERT, onde a documentação em papel continua a ser um dos recursos usados como meio de registro dessas atividades. Neste sentido, torna-se fundamental que os processos de formação em sala sejam informatizados e otimizados para que se apresentem mais ágeis, eficazes e produtivos, visando funcionalidades de planejamento, gestão de inscrições, confirmação das presenças dos formandos, avaliação de aprendizagem e monitorização da formação.

O *software* ALERT® é um produto que oferece funcionalidades que permitem a gestão, segurança, utilização de perfis diversos e o rigor aquando da adoção dos sistemas de informação de atividade clínica em serviços de saúde, nomeadamente, hospitais e postos médicos. As partes interessadas (técnicos de saúde, enfermeiros e médicos) beneficiam de sistemas de informação que apoiam as suas atividades clínicas nos diferentes serviços (consulta externa, internamento, cirurgia, urgência, entre outros), de modo a otimizar todo o processo.

Como referido anteriormente, este projeto nasce da constante preocupação da ALERT em prover meios para assegurar formação aos utilizadores dos seus produtos ao otimizar os processos de gestão inerentes. O ALERT® é um sistema clínico global que utiliza tecnologia tátil e oferece uma multiplicidade de produtos e serviços. De entre esses, importa destacar o produto ALERT® eLEARNING, um SGA que fornece aos utilizadores uma formação completa em diversos produtos ALERT®, por intermédio de uma plataforma *web* e de uma interface de utilizador intuitiva.

O desenvolvimento de uma Plataforma de Gestão da Formação apoiará a administração dos processos da formação em sala de aula, facilitando o acesso a recursos de aprendizagem, avaliações e de notificações.

Com base na Figura 1, o problema desta tese, por um lado, passa pela disponibilização de funcionalidades e ferramentas que permitam informatizar aquilo que são as atividades da formação em sala, não deixando de lado o que já existe da parte do e-learning (cenário “eLEARNING”), tendo como produto final a informatização do cenário “Formação em Sala” e uma apreciação do todo que é apresentado no cenário pelo número 2.

Em suma, até ao momento, a ALERT utiliza papel para traduzir os processos de formação em sala. Com o desenvolvimento da nova plataforma, será possível obter uma visão geral do e-learning, isto é, consumir serviços e dados disponibilizados pela mesma (cenário “eLEARNING”), nomeadamente, saber em que formações determinados alunos participaram, inscrições, datas

das formações, quantas pessoas compareceram ou não, registo dos colaboradores e dos clientes, que formações existem, estão previstas, entre outros. Assim, o problema atual da ALERT é ter informatizado o cenário “eLEARNING” e não possuir a visão integrada do que também acontece no cenário “Formação em Sala”, isto é, dos principais processos do ciclo da formação, no contexto da sala de aula, e garantir uma maior eficiência na sua gestão e produtividade.

1.3 Objetivos

Com o desenvolvimento de uma plataforma de gestão da formação, através da ligação à Internet e de forma remota, é possível aceder à mesma inserindo as credencias (*username* e *password*) únicas e personalizadas para cada utilizador registado na base de dados da instituição a que pertence. A cada registo é associado um determinado perfil que permite aceder a um conjunto de funcionalidades específicas dentro da plataforma.

Para colaboradores da ALERT, será possível obter uma visão de:

- Gestão de projetos, nomeadamente, gestão dos planos anuais de formação, de clientes/entidades;
- Gestão de formandos, como por exemplo, vista histórica dos seus dados sobre ações de formação frequentadas, classificações e certificações adquiridas;
- Gestão de formadores;
- Gestão de colaboradores e clientes;
- Gestão de cursos, com funcionalidades extra para editar e emitir horários dos planos de formação;
- Gestão de ações, nomeadamente, edição de folhas e/ou documentos de inscrição e presença;
- Consultas e pesquisas;
- Configurações administrativas;
- Acompanhar planos de formação de aprendizagem;
- Emitir relatórios;

Como referido na secção 1.1, o objetivo deste projeto não é desenvolver um produto concorrente com o e-learning, mas sim uma nova plataforma *web* que informatize os pontos

enumerados anteriormente. Face ao exposto, seguidamente são evidenciados os pontos mais importantes para o desenvolvimento deste projeto e concretização dos seus objetivos:

- Analisar, recolher e definir os processos chave de formação numa empresa. Entender como a formação contribui para o desenvolvimento pessoal e profissional das pessoas, melhora o desempenho organizacional, facilita a integração dos colaboradores e contribui para a melhoria dos índices de produtividade e capacidade competitiva da organização;
- Conhecer o estado da arte de sistemas de gestão de aprendizagem e plataformas de gestão da formação. Sobre os mesmos, efetuar o devido levantamento de requisitos que permitam identificar e recolher funcionalidades a implementar, bem como planeamento da implementação das funcionalidades. Importa mencionar a necessidade de analisar o estado da arte de plataformas e-learning e plataformas de gestão da formação, nomeadamente, as suas bases de dados, arquiteturas e mecanismos de autenticação e autorização;
- Estudar e reconhecer o processo administrativo efetuado pela ALERT para que a plataforma a desenvolver seja o mais fidedigna possível no que respeita à informação dos seus colaboradores e clientes;
- Definir a arquitetura para a nova plataforma Web;
- Desenvolvimento de funcionalidades e ferramentas para apoio à formação;
- Desenvolvimento de testes de qualidade que assegurem a usabilidade, rigor e otimização da plataforma;

1.4 Partes Interessadas

A definição das partes interessadas tem como objetivo identificar as pessoas que irão utilizar o produto desenvolvido por meio de diversas funcionalidades. Como Freeman (2001) classificou, essas partes interessadas são todas as entidades, grupos de entidades ou organizações que tenham capacidade para interferir ou sofrer interferência relativamente às atividades de determinada organização.

Também segundo o guia PMBok® (Institute, 2017), todos os projetos têm partes interessadas que tanto são afetadas, como também podem afetar o projeto de forma positiva ou negativa.

Face ao evidenciado nos parágrafos anteriores, em seguida são enumeradas as principais partes interessadas (utilizadores registados no produto ALERT® eLEARNING) da plataforma a ser desenvolvida para responder ao problema apresentado na secção 1.2:

- **Colaboradores ALERT** (utilizadores registados aquando a sua integração na empresa), que afetam e são afetados pela organização, uma vez que, serão responsáveis por executar os processos da ALERT. Estes podem inscrever-se em formações e obter certificados;
- **Empresa ALERT** que investe no seu *software* ao motivar o desenvolvimento de uma plataforma de gestão da formação que responde às suas necessidades. Combater o excesso de tempo despendido aquando a elaboração de um plano de formação para colaboradores e clientes. Diminuição também de recursos físicos. Maior facilidade de acesso a dados estatísticos. Disponibilização de conteúdos ALERT® e possibilidade de impulsionar o crescimento dos colaboradores através da funcionalidade de inscrição num plano de formação. Facilitar o acompanhamento da formação dos seus colaboradores e clientes;
- **Cientes ALERT** (hospitais, centros de saúde, entre outros) que compram os produtos para os disponibilizar aos profissionais que trabalham nesses locais. Demonstram interesse na aquisição dos produtos desde que os mesmos se encontrem devidamente adaptados à sua cultura para que, assim, possam melhorar os serviços que disponibilizam, tendo um produto de qualidade, seguro, intuitivo e interativo para todos os utilizadores;
- **Profissionais de saúde** (técnicos, enfermeiros, médicos, entre outros) que utilizam os produtos e/ou consomem os serviços disponibilizados diariamente. Numa relação de comparação, é possível afirmar que os produtos ALERT® eLEARNING afetam a vida dos profissionais de saúde, tal como os profissionais de saúde afetam diretamente os resultados da empresa. Com o novo produto ALERT®, os técnicos, enfermeiros e médicos podem acompanhar o progresso relativamente à sua formação;
- **Outros *softwares* e *providers*** – CRM para a equipa de vendas, integração com outras plataformas e uma vertente financeira para a captura de pagamentos;

Como visto, a plataforma a desenvolver impacta estas entidades que demonstram ou possuem algum tipo de interesse nos produtos ALERT®. No âmbito deste projeto, é necessário conhecer os processos administrativos não apenas pelo que a empresa e os seus colaboradores necessitam de planear, preencher e executar para realizar formação em sala, mas também para compreender as necessidades primárias dos clientes e profissionais de saúde e de que forma esta plataforma afeta o desenvolvimento sobre o conhecimento nos produtos ALERT® (tanto produto clínico como produto e-learning).

1.5 Motivação Pessoal

Sobre a motivação pessoal, aceitar desenvolver uma plataforma desta dimensão não foi muito difícil, uma vez que, a curiosidade sobre a educação e plataformas e-learning despertou em

mim há alguns anos. Poder participar no desenvolvimento de uma é imensamente gratificante, pois acredito genuinamente que estamos a falar do futuro do processo de formação em sala. As funcionalidades para o qual esta plataforma se encontra desenhada transcendem soluções inacreditáveis e insubstituíveis. A criação de planos de formação genéricos para organizações, a realização de exames para obter certificação, o acompanhamento na formação de colaboradores e clientes face a determinado produto e/ou matéria que seja favorável à evolução de uma empresa é exponencial.

Ainda assim, para além do interesse pessoal, a proposta colocada insere-se na área que considero de maior oportunidade para um engenheiro informático. A divisão constante entre engenheiros que compreendam apenas de *back-end* (aquilo que o utilizador não vê, ou seja, a lógica de negócio, o lado dos servidores) e o *front-end* (aquilo que o utilizador vê e com o qual interage, ou seja, a interface gráfica de um programa ou aplicação) é cada vez mais evidente. Por causa disto, a possibilidade de desenvolver ambas as componentes era o objetivo de estágio, pois só assim posso aspirar ao perfil de programadora que consegue trabalhar não somente com um determinado tipo de programação, mas também compreender de forma razoável uma porção de tecnologias envolvidas num sistema, nomeadamente: base de dados, infraestrutura, sistema operativo, servidor web, redes, CSS(3), HTML(5), Angular 8, Java, Spring Boot, entre outros.

Em conclusão, pretendo atingir o perfil de *Full Stack Developer* e ajudar a empresa a atingir os seus objetivos. Tenciono ser um membro ativo, companheiro, dedicado e integrante do sucesso dos produtos ALERT® de forma a ajudar a equipa que integro (equipa e-learning) a concretizar todos os objetivos a que me propuser e desenvolver as minhas capacidades de comunicação.

1.6 Planeamento do Projeto

No início do desenvolvimento deste projeto, foi definida uma metodologia de trabalho minuciosa, detalhada e exata que refletisse todas as etapas e processos desenvolvidos no trabalho de pesquisa, bem como um planeamento com o intuito de fornecer metas e objetivos diários.

Assim, o projeto foi dividido em duas grandes fases, tendo em conta os prazos de entrega definidos no âmbito da unidade curricular em que este projeto se enquadra (TMDEI). Para além disso, foi necessário considerar as necessidades, os tempos e os objetivos da empresa para este projeto. Tendo em consideração ambas as partes, o planeamento foi sofrendo alterações à medida que o projeto progredia, atendendo ao *feedback* de todas as partes envolvidas no seu desenvolvimento. Todo o planeamento foi desenvolvido com o auxílio do diagrama de Gantt (Diagrama de Gantt, 2017), TortoiseSVN (Bhoir & Patil, 2018) e *software* JIRA (Atlassian, 2020).

Assim, numa primeira fase, como referido anteriormente, foi imprescindível planear e definir tarefas a serem executadas antes do término do projeto:

- **Estudo de sistemas de gestão de aprendizagem e plataformas de gestão da formação:** entender o seu conceito, para que servem e de que forma afetam ambientes e-learning e formação nas empresas;
- **Estudo sobre o mercado e-learning e b-learning** (modalidade de ensino que combina elementos da formação à distância em regime de e-learning com elementos da formação em sala), os seus dados e as tendências principais;
- **Analisar e compreender os conceitos de um SGA e plataformas de gestão da formação:** perceber as características principais, dominar e interpretar o negócio do processo da formação e de que modo a qualidade de um produto deste tipo melhora a resposta das organizações;
- **Analisar versões de sistemas de gestão de aprendizagem e plataformas de gestão da formação:** perceber onde a plataforma a desenvolver deve ser melhor que as concorrentes. Interiorizar e definir o melhor *design*;
- **Levantamento de requisitos funcionais** com base nos sistemas de gestão de aprendizagem e plataformas de gestão da formação: identificar funcionalidades importantes que compõe sistemas e plataformas deste tipo;
- **Estudo da documentação da ALERT sobre formação:** observar, interligar e analisar documentação que permite compreender o processo de formação da empresa. Com isto, será possível identificar atributos das tabelas de base de dados;
- **Analisar perfis de utilizadores para casos de uso:** identificar perfis e compreender de que modo os mesmos afetam a lógica de negócio da plataforma;
- **Analisar alternativas relativamente à fase de desenvolvimento:** o período de análise deve destacar estruturas previamente identificadas e que devem ser aproveitadas durante a implementação da solução de forma a maximizar a eficiência da fase de desenvolvimento;
- **Reconhecer a proporção do problema:** não analisar somente a solução, pois o problema exposto contém detalhes que merecem a mesma ponderação. Portanto, aquando da definição da lógica de uma funcionalidade, é necessário relembrar os detalhes anteriores, uma vez que, podem influenciar os seguintes estágios de *design* e desenvolvimento;
- **Analisar o valor da solução proposta para o cliente:** não priorizar determinados recursos em detrimento de outro caindo no erro de construir uma plataforma inconsistente;

- **Compreender a importância dos requisitos não funcionais:** usabilidade, elasticidade, entre outros. Isto permitirá identificar os recursos para o desenvolvimento da plataforma;
- **Analisar a arquitetura de *software* do produto:** desenvolver um produto do zero implica a criação de novos serviços. Contudo, é importante dedicar tempo para estudar a arquitetura que melhor beneficie o produto.

Numa segunda fase, já com as considerações retiradas durante a investigação realizada na primeira fase, foram definidas as seguintes tarefas mais relevantes:

- **Melhorar o *design*** elaborado para a plataforma;
- **Analisar e desenvolver os requisitos definidos** na primeira fase tendo em conta o desenho elaborado para a solução;
- **Experimentar e avaliar** os requisitos desenvolvidos;
- **Analisar** os resultados;
- **Aferição da satisfação** das partes interessadas com o projeto desenvolvido.

1.7 Contribuições e Valor

Com o desenvolvimento de uma plataforma de gestão da formação a empresa ALERT poderá visualizar dados estatísticos referentes a planos de formação finalizados e por iniciar, formandos que necessitam de formação no produto clínico, disponibilizar cursos e realizar atividades durante a formação em sala, entre outros. Esta plataforma vem reforçar as funcionalidades disponibilizadas pelo produto ALERT® eLEARNING, bem como disponibilizar novas que se adequem às necessidades da empresa.

Assim, pretende-se que a plataforma desenvolvida disponibilize um vasto leque de funcionalidades que espelhem o processo de formação não informatizado, ou seja, através da ligação à Internet e por meio de um dispositivo, o utilizador consiga desenvolver um plano de formação informaticamente e não manualmente. O acesso à informação será mais rápido.

Face ao exposto, este projeto contribui com os seguintes aspetos:

- Valorização do produto ALERT® eLEARNING;
- Oferecer a melhor solução para instituições ALERT relativamente ao processo de formação em sala, permitindo o acesso rápido à informação, uma vez que, esta se encontra num local único. Isto, é uma forma de reduzir os custos envolvidos nos processos de formação em sala para a ALERT;

- Aplicação de novas tecnologias que contribuem para o sucesso da plataforma, nomeadamente, uma interface gráfica responsiva e intuitiva;
- Permitir que os utilizadores da plataforma acompanhem o seu progresso na formação dos produtos clínicos ALERT® ou peçam nova formação de modo a renovarem os seus conhecimentos.

1.8 Estrutura do Documento

Excluindo o capítulo corrente, o documento apresenta mais capítulos (sem esquecer as referências e os anexos), particularmente:

- **Estado da Arte:** este capítulo expõe o conhecimento adquirido sobre sistemas de gestão de aprendizagem e plataformas de gestão da formação. Ao longo do mesmo é explicado o papel e as principais responsabilidades das empresas na gestão da formação dos seus colaboradores e clientes, seguido de uma abordagem face a algumas *Application Programming Interface* (API) dos exemplos mostrados, bem como o estudo (se possível) da arquitetura de *software* das mesmas. Para além disso, é elaborada uma tabela de comparação entre as funcionalidades encontradas aquando o estudo das várias plataformas, bem como a apresentação visual das mesmas. No fim de cada estudo, são identificadas algumas conclusões pertinentes para o produto desenvolvido;
- **Análise de Valor:** este capítulo apresenta a análise de valor do produto desenvolvido. Esta análise diz respeito às funções do produto desenvolvido, de modo a atender às demandas do cliente. Para compreender esses requisitos funcionais, o processo de revisão inclui a finalidade para a qual o produto é utilizado através da utilização de várias técnicas e/ou modelos que auxiliam na tomada de decisão. Estes permitem agregar valor para o cliente e gerar benefício para a empresa;
- **Desenvolvimento da Solução:** este capítulo apresenta os perfis de utilizadores identificados para o projeto desenvolvido, bem como os requisitos funcionais e não funcionais. Em seguida é apresentado o modelo de domínio e a interface gráfica a adotar na plataforma;
- **Arquitetura:** este capítulo destina-se à descrição e apresentação arquitetural da plataforma. A contextualização do *software*, como meio integrante de um sistema de componentes internas e externas. São apresentados os diagramas de componentes, pacotes e implantação;
- **Realização de Casos de Uso de maior Relevância:** tendo em consideração a fase de desenvolvimento da solução, neste capítulo são apresentados os desenhos e a implementação dos casos de uso com maior relevância do sistema;

- **Testes:** este capítulo mostra como as camadas principais da API (controladores, serviços e repositórios) foram testadas, usando testes unitários e de integração;
- **Experimentação e Avaliação:** este capítulo descreve o processo pelo qual o produto passa para ser experimentado e avaliado por colaboradores ALERT;
- **Conclusão:** neste capítulo são descritos os objetivos alcançados, as limitações encontradas para o desenvolvimento deste projeto, aquilo que efetivamente se pretende da plataforma no futuro e, por último, é prestada uma apreciação final;
- **Referências:** este capítulo apresenta as referências consultadas e que permitiram a composição deste documento.

2 Estado da Arte

No presente capítulo, é abordado o espaço que a tecnologia passou a ocupar no mundo da educação. Posteriormente, é abordado o e-learning empresarial e a forma como a chegada de sistemas de gestão de aprendizagem e sistemas de gestão de formação para este setor distingue a qualidade de projetos de empresa grande e pequena. Seguidamente, são apresentados alguns sistemas e é elaborada uma tabela de comparação das funcionalidades dos exemplos abordados. Por fim, é feito um estudo sobre as interfaces de utilizador de LMS e TMS.

2.1 Educação e Tecnologia

Atualmente, é possível afirmar que se olha para a tecnologia mais pelas respostas que oferece do que pelas questões que levanta, ou seja, caminhamos hoje para mais uma das transições sociais que transformam a sociedade ao longo dos tempos (Moreira Kenski, 2003). Para se compreender este processo dinâmico, além da necessidade de refletir sobre as mudanças constantes da sociedade, quer sejam na sua forma de agir, pensar, consumir, relacionar e exercer a cidadania, também é necessário compreender a evolução dos dispositivos tecnológicos que fizeram parte dessas modificações. Assim, entende-se que as alterações sociais estão diretamente ligadas às transformações tecnológicas da qual a sociedade se apropria para se desenvolver e manter (Moreira Kenski, 2003).

Para Castells (2000), a habilidade ou inabilidade de uma sociedade dominar a tecnologia, fazer uso e decidir o seu potencial tecnológico, modifica a sociedade a um ritmo acelerado e traça a história e o destino social dessas comunidades; remetendo que essas modificações não

ocorrem de forma igual e completa em todos os lugares, ao mesmo tempo e instantaneamente em todas as realidades, mas sim como um processo temporal e para alguns, demorado.

Para Patrícia (Ramos), “a Internet trouxe ao mundo da educação e da formação novos paradigmas para ensinar e aprender”, o que significa que através dela se tornou possível a disponibilização de conteúdos de forma mais rápida, segura e interativa que antigamente. Isto implica a necessidade de novos modelos de ensino e aprendizagem por meio de ferramentas digitais que disponibilizem conteúdos de aprendizagem. Estas são vulgarmente denominadas como Sistemas de Gestão de Aprendizagem (SGA ou LMS) e tem capacidade para centralizar em si o mesmo tipo de funções que o aluno encontraria num espaço de formação presencial.

Como afirmou Albert Einstein (2011), “a educação é o que resta após alguém esquecer aquilo que aprendeu na escola”. Esta afirmação enfatiza a necessidade de aprendizagem contínua na educação. Consequentemente, as capacidades individuais dos formandos devem ser atualizadas repetidamente o que levantou a necessidade de arranjar novas formas de ensino, permitindo que o formando escolha aquilo que deseja aprender, a hora a que pretende fazê-lo e na forma que melhor se adegue ao seu ritmo de aprendizagem.

Com base na informação dos parágrafos anteriores e enquadrando a mesma num contexto empresarial, o que efetivamente se pretende numa organização é a formação contínua dos seus colaboradores e clientes através de uma aprendizagem instantânea, online, autodidata, em movimento e colaborativa. A tecnologia e-learning é a resposta à formação empresarial uma vez que permite criar cursos com base nas necessidades de cada organização.

2.2 O E-learning Empresarial

A chegada de plataformas e-learning em ambientes empresariais permitiu às organizações disponibilizar conteúdo e atividades de aprendizagem (The global e-learning market is expected to reach \$238 billion by 2024, 2019). Importa ainda referir que, algumas organizações apresentam dificuldade em agrupar pessoas para formações em sala por questões logísticas ou devido ao impacto que isso teria sobre os fluxos de trabalho e continuidade dos negócios (Corporate E-learning Market, 2020).

O e-learning empresarial garante que os colaboradores se encontrem atualizados e desenvolvam requisitos para o bom funcionamento das aplicações. Isto permite aos colaboradores acompanhar os seus registos online e compreender como o desempenho de cada um afeta positiva ou negativamente os objetivos da organização e permite aos recursos humanos (RH), através de processos automatizados, gerar relatórios personalizados e controlar a retenção de talentos (Corporate E-learning Market, 2020).

A Figura 2 apresenta um gráfico que permite compreender a evolução do mercado dos sistemas de gestão de aprendizagem entre o período de 2018 e 2026.

Global Learning Management System Market (US\$ Mn), 2018 to 2026

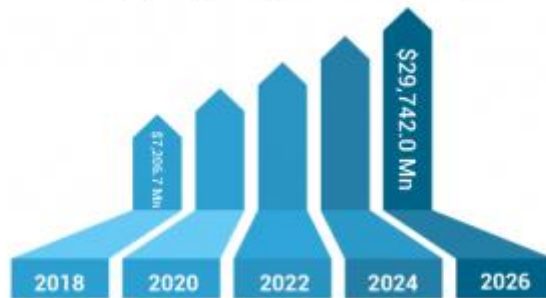


Figura 2 – Tamanho do Mercado do Sistema de Gestão de Aprendizagem, 2018-2026 (US\$ Milhões) (Arterburn, 2019)

O mercado do LMS foi avaliado em 7,206.7 milhões de dólares em 2018 e deverá atingir 29,742.0 milhões de dólares até 2026 (Arterburn, 2019). Devido a este crescimento, as empresas passaram a utilizar a *cloud*, uma vez que, “as plataformas e-learning requerem muito hardware e recursos de *software*” (Bhutanie & Bhardwaj, 2019). A *cloud* “não é uma entidade física, mas sim uma rede vasta de servidores remotos espalhados pelo mundo que se encontram interligados e que devem funcionar como um ecossistema único. Estes servidores foram concebidos para armazenar e gerir dados, executar aplicações, fornecer conteúdos ou um serviço” (Microsoft).

Tendo em conta o texto de opinião de Michelle (2019), os benefícios de sistemas de gestão de aprendizagem baseados na *cloud* apresentam: menor custo, maior segurança na troca e armazenamento de dados, melhor acessibilidade, implantação rápida, tecnologias atualizadas, espaço de armazenamento maior, possibilidade total de personalização e escalável. Em contrapartida, o crescimento do mercado empresarial relativamente a estes sistemas pode ser prejudicado devido a alguns dos desafios enfrentados pelos SGA baseado na *cloud*, nomeadamente, riscos relacionados com a proteção de dados ou falta de Internet (Michelle, 2019).

Em suma, desde que exista ligação à Internet, uma empresa consegue visualizar, criar e associar planos de formação a utilizadores e instituições da mesma. Adicionalmente, os utilizadores podem criar cursos utilizando a sua própria documentação (vídeos, apresentações) ou conteúdo disponível na Web (Pappas, TalentLMS Reviews 2020). Com um SGA é possível expor uma sala de aula virtual e acompanhar a documentação e ministração de cursos online. A sua importância aumenta quando adaptado ao setor empresarial, uma vez que, proporciona a elaboração de formações personalizadas, em tempo real e direcionadas aos colaboradores e clientes conforme as necessidades de cada um (Michelle, 2019). Contudo, de que forma um gestor de equipa pode acompanhar a evolução da aprendizagem dos membros da sua equipa?

2.3 Sistemas de Gestão de Formação

Com base no que foi referido na secção 2.2, é possível aferir que a crescente globalização dos mercados e a externalização das atividades formativas, associadas ao elevado grau de especialização e crescente importância do nível de serviço prestado ao cliente, indica que é necessário dotar as empresas de ferramentas que permitam aumentar a competitividade das suas formações. Os Sistemas de Gestão da Formação, em inglês *Training Management System* (TMS), são plataformas especificamente projetadas para transmitir conteúdo e-learning com o qual os colaboradores e clientes de uma organização iniciam o processo de desenvolvimento de novas competências, melhoram o seu desempenho e acrescentam valor à organização (Training Management Software Systems).

Um TMS, determinante ao sucesso ou insucesso de uma empresa, pode conduzir ao sucesso, oferecendo as ferramentas necessárias para criar planos de formação específicos para todos os departamentos de uma empresa. Isto permite construir uma base de formação personalizada que se responsabilizará pelos processos de formação que abordam competências específicas para alguns dos membros da organização (como por exemplo, necessidade de adquirir novos conhecimentos ou ajudar no processo de integração de novos colaboradores, entre outros (Training Management Software Systems).

2.3.1 Qual a sua Importância no Setor Empresarial?

Um TMS é desenhado para ser o núcleo dos processos de formação de uma organização, permitindo de forma rápida supervisionar os mesmos sem que exista necessidade de interpolar entre ferramentas e sistemas diferentes (Training Management Software Systems). Um sistema de gestão de formação permite centralizar toda a administração de gestão de cursos que, conseqüentemente, leva à não perda de informação, uma vez que, algumas organizações continuam a recorrer ao suporte de papel para guardar e monitorizar esses processos.

O agendamento e a execução de grandes quantidades de cursos demandam organização e coordenação entre as equipas de uma empresa. Este tipo de planeamento ajuda a garantir que as aulas estão a decorrer quando deveriam, que existem salas de aulas, formadores e recursos disponíveis (Training Management Software Systems). Para além disso, diminuiu o erro, poupa tempo e esforços das equipas que necessitam de planear formações para várias instituições.

2.3.2 Que Funcionalidades Suporta um TMS?

Todos os sistemas de gestão de formação diferem nas suas funcionalidades. Em seguida, é apresentado aquilo que um TMS deve incluir (Training Management Software Systems):

- Um sistema de gestão de cursos, em inglês *Course Management System* (CMS), para permitir a publicação de cursos online diretamente no TMS. Isto ajuda a poupar tempo e a não editar a mesma informação em dois sistemas diferentes;

- Um CRM, um sistema que permite armazenar a informação quer dos novos, quer dos antigos colaboradores e clientes de uma organização;
- Um sistema de relatórios ligado ao TMS. Isto permite gerar relatórios dos dados inseridos no TMS instantaneamente e atualizados com a versão mais recente dos mesmos, sem trabalho adicional para as equipas;
- Um LMS que ofereça e-learning aos formandos.

2.3.3 Quais são os Grupos de Utilizadores de um TMS?

Como referido anteriormente, um TMS pode ser adaptado às necessidades de formação das organizações destacando os grupos de utilizadores: *Training Managers, Administrators e Trainees* (Training Management Software Systems). Contudo, podem existir organizações que necessitem de uma extensão desse grupo para o seu TMS. Cada perfil é único e responsável, apresentando funcionalidades comuns ou distintas.

2.3.4 FORINSIA

Em seguida é apresentado o único TMS que ajudou no desenvolvimento da arquitetura e levantamento de requisitos para este projeto. Apesar de existirem outros sistemas de gestão de formação, os mesmos foram considerados irrelevantes pela falta de informação.

A FORINSIA é um *software* de gestão da formação online para formação interna das empresas. Esta plataforma garante uma gestão integrada, sólida e permanente e permite às empresas gerir e otimizar os processos pedagógicos e e-learning (INSIA). Esta ferramenta permite gerir os cursos de formação até à análise de dados de resultados da atividade formativa e é um tipo de *software* cliente-servidor. Isto significa que é necessário estar em constante comunicação com o servidor, pois existe uma grande quantidade de informação a circular pela rede (INSIA).

Durante o processo formativo, existem vários intervenientes com funções distintas e por esta razão, a FORINSIA oferece funcionalidades de acordo com os seguintes tipos de utilizadores: *Administrator, Coordenador, Técnico Administrativo, Trainer, Trainee* e Empresa (INSIA). Com estes 6 perfis é possível assegurar o cumprimento das regras associadas a empresas e implementar uma prática de gestão da formação com otimização dos processos.

A Tabela 1 apresenta as funcionalidades administrativas oferecidas pela plataforma e que acabam por ser distintas das apresentadas nas próximas secções, nomeadamente, em 2.5.1, 2.5.2, 2.5.3 e 2.5.4.

Tabela 1 - Principais funcionalidades da plataforma FORINSIA

Funcionalidades administrativas
Área privada para funções e consultas
Histórico de ações frequentadas pelo formando
Histórico de formação ministrada pelo formador
Gestão de candidaturas
Gestão de caracterização/edições, cursos e módulos
Constituição de ações (seleção de formandos e formadores)
Gestão e reserva de equipamentos e salas
Registo de pautas de avaliação por módulos
Registo histórico da formação externa por formando
Consultas e ações contextualizadas em empresas e clientes
Gestão de clientes
Gestão de inquéritos de avaliação

Em suma, foi possível compreender o lado administrativo de uma organização e que funcionalidades devem ser necessárias a um TMS. Plataformas como a FORINSIA apresentam um grau de importância significativo, uma vez que, permite diminuir o tempo de planeamento em todas as fases do processo de gestão da formação, responder facilmente às exigências dos colaboradores e clientes de uma empresa sobre formações específicas, adaptar rapidamente e com menor esforço às mudanças que a realidade formativa impõe numa empresa, exportar informação para formatos que facilitem as tarefas da formação e receber e tratar com facilidade as inscrições dos utilizador.

2.4 Conceitos

Tendo compreendido a importância de um SGA (explicado em 2.2) e de um TMS (explicado em 2.3), a secção 2.5 apresenta alguns exemplos arquiteturais de LMS. Contudo, para compreender essas estruturas e toda a sua dinâmica, foi desenvolvida esta secção para a interiorização de alguns conceitos, nomeadamente, *Web Services*, API, REST, *Training Plan/Learning Plan* e *Learning Program*.

2.4.1 *Web Services*

Os *Web Services* permitem converter as soluções existentes em aplicações e tornar as suas funcionalidades públicas, uma vez que, são componentes de *software* que comunicam através de protocolos (Portela & Queirós, 2018). Contudo, os *Web Services* evoluíram e as novas

tecnologias passaram a utilizar APIs e protocolos do tipo *Representational State Transfer* (REST) (Portela & Queirós, 2018).

2.4.2 API

Uma *Application Programming Interface* (API) é um “conjunto de rotinas e padrões estabelecidos por um *software* para que as aplicações que não pretendem envolver-se em detalhes da sua implementação, mas apenas usar os seus serviços, possam utilizar as funcionalidades desse *software*” (Portela & Queirós, 2018). De modo geral, uma API contém funções acessíveis apenas por programação e que permitem utilizar características do *software* menos expectáveis ao consumidor final.

No contexto do desenvolvimento Web, uma API é um conjunto pré-definido de mensagens de pedido e resposta HTTP (Portela & Queirós, 2018), geralmente expressas no formato *JavaScript Object Notation* (JSON). Ao usufruir de uma aplicação ou site, o programador passa a ter acesso, via API, a outros sistemas e aplicações (Portela & Queirós, 2018).

2.4.3 Arquitetura REST

Quanto ao protocolo de comunicação REST, é caracterizado por não apresentar estado e separar as camadas do cliente e do servidor (Portela & Queirós, 2018). “A arquitetura REST pressupõe que as implementações do cliente e servidor sejam efetuadas de forma independente, sem que um tenha conhecimento do outro” (Portela & Queirós, 2018). Isto significa que o código do lado do cliente pode ser alterado sem que isso afete a operação do servidor e vice-versa.

2.4.4 REST API

A utilização de API com REST apresenta benefícios na eficiência, gestão de processos, diversidade de aplicações, interoperabilidade, entre outros (Portela & Queirós, 2018). As API estão disponíveis na *cloud* através de *link* (*Uniform Resource Identifier* – URI). Diferentemente dos *Web Services*, a comunicação com uma API é mais simples porque temos de comunicar apenas com a interface (API) (Portela & Queirós, 2018). Na Figura 3, é possível compreender como é efetuada uma chamada a uma API.

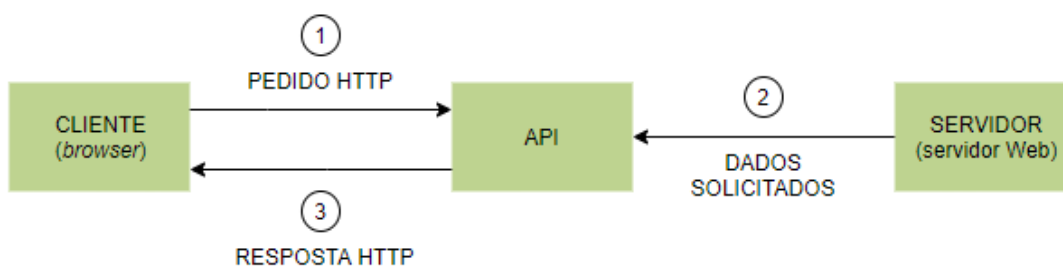


Figura 3 - Chamada API. Adaptado de Filipe e Ricardo (2018).

Como ilustrado, o cliente (*browser*) faz um pedido à API que tem ligação direta com o servidor. Para responder ao pedido, por sua vez, o servidor Web envia os dados solicitados e a API encaminha os mesmos, através da resposta, ao cliente.

2.4.5 Training Plan/Learning Plan

Uma das particularidades que a plataforma Captivate Prime oferece em comparação com as apresentadas em 2.5.2 e 2.5.3, são os planos de formação (em inglês, *Training Plans/Learning Plans*).

Um *learning plan* é um conjunto de regras que inscreve alunos em formações específicas com base em determinados critérios (Learning Programs, 2020). Um *learning plan* permite a um administrador atribuir automaticamente cursos ou certificações com base na ocorrência de certos eventos, como a entrada de um novo colaborador ou alteração na designação ou localização dos colaboradores (Learning Programs, 2020).

No contexto deste projeto, um exemplo de um *learning plan* seria quando a organização contrata um novo colaborador e atribui automaticamente os programas de iniciação. Da mesma forma, se um colaborador for promovido, o mesmo é inscrito automaticamente nos programas de iniciação para o cargo específico.

2.4.6 Learning Program

Learning programs são um conjunto de cursos ou atividades (exercícios elaborados nas sessões/cursos) criados pelos formadores ou administradores, que atendem a metas específicas dos utilizadores interessados (maioritariamente formandos).

2.5 Arquiteturas de Sistemas de Gestão de Aprendizagem baseados na Cloud

Nesta secção serão estudados quatro sistemas de gestão de aprendizagem utilizados em setor educacional (escolas e universidades) e empresarial. São também estudadas algumas API

disponibilizadas pelos respectivos autores de modo a compreender a lógica de negócio de um LMS. Adicionalmente, são analisadas arquiteturas de modo a elaborar a mais adequada para a plataforma deste projeto.

2.5.1 TalentLMS

TalentLMS é exemplo de uma ferramenta e-learning baseada na *cloud*. O TalentLMS permite poupar tempo, uma vez que, é possível reutilizar o material de formação existente (Pappas, TalentLMS Reviews 2020). Esta plataforma oferece uma REST API que permite integrações personalizadas das suas operações, sendo compatível com quase todos os sistemas e linguagens de programação de terceiros (*third party system*) (Pappas, TalentLMS Reviews 2020).

Assim, com recurso à interface REST API da TalentLMS, foi possível aceder ao *endpoint* (URL) com maior relevância para o desenvolvimento deste projeto. A Figura 4 ilustra uma adaptação da resposta HTTP em JSON com base nos dados da Tabela 2.

Tabela 2 – Dados Pedido REST para *user* com id = 1

URI	Operação na Base de dados	Parâmetro URL	Método HTTP
https://samples.talentslms.com/api/v1/users/{userID}	Nenhuma (apenas lê)	{userID}	GET



Figura 4 - Pedido REST para *user* com *id* = 1. Adaptado de Filipe e Ricardo (2018) e TalentLMS API (TalentLMS, 2020)

No contexto deste projeto, com base no objeto apresentado em cima, são relevantes os seguintes valores:

- Propriedades com os seguintes nomes: “*id*”, “*login*”, “*created_on*”, “*last_updated*”, entre outros;
- Propriedade com o nome “*login_key*” que utiliza um método de encriptação seguro para proteger a *key* (*password*) do utilizador;

- Propriedade com o nome “courses” que contém um *array* de tamanho 2. Este *array* apresenta 2 objetos que indicam que o utilizador com “id” igual a 1 é instrutor nos cursos com “id” igual a 1 e 19.
- Propriedade com o nome “groups” que contém um *array* de tamanho 1. Este objeto com “id” igual a 1 representa o grupo em que o utilizador se encontra. Este certamente conterá outros utilizadores que trocam informações. Com isto é evidente que na base de dados as tabelas USER (utilizador) e GROUP (grupo) apresentam algum tipo de relação;
- Propriedade com o nome “certifications” que significa que o utilizador com “id” igual a 1 obteve certificação no “course_id” igual a 12. Com isto é evidente que na base de dados as tabelas USER e COURSE (curso) apresentam algum tipo de relação.

Em síntese, foi elaborada a Tabela 3 que contém as propriedades mais importantes no desenvolvimento deste projeto segundo a análise efetuada à Figura 5.

Tabela 3 – Propriedades relevantes

Propriedades		
id	user_type	Login_key
login	status	array certifications
first_name	desactivation_date	array courses
last_name	created_on	array groups
email	avatar (imagem)	array certifications

Em suma, o TalentLMS é um exemplo de um *software* baseado na *cloud* seguro e encriptado para todos os tipos de comunicações, oferecendo benefícios como mobilidade empresarial, empenho dos colaboradores e previsão de custos (Arterburn, 2019). Isto capacita os utilizadores a configurar, fornecer e acompanhar o processo de autoaprendizagem. O TalentLMS também permite criar cursos de aprendizagem baseados em determinadas habilidades (*skills*), os quais podem aumentar o potencial do mercado do sistema de gestão de aprendizagem global conforme ilustrado na Figura 2.

2.5.2 Intuo LMS

O Intuo LMS é um LMS baseado na *cloud*, que dá suporte à gestão das habilidades (em inglês, *skills*) dos colaboradores de uma organização (INTUO Learn). O sistema foca particularmente na monitorização do desenvolvimento profissional, seja em apoio à formação geral ou como parte do processo de integração (INTUO Learn).

O Intuo LMS permite obter relatórios e análises de progresso detalhadas que mostram aos formadores e administradores da organização uma imagem nítida das *skills* de cada colaborador

e cliente, permitindo tomar decisões baseadas em dados estatísticos sobre onde atribuir as melhores pessoas para obter o melhor resultado (INTUO Learn). Assim, o Intuo LMS faz parte de uma plataforma de capacitação de talentos.

O objetivo desta plataforma é apoiar o departamento dos recursos humanos na atração, retenção e participação dos melhores talentos (colaboradores de uma organização), bem como apoiar os líderes das equipas de desenvolvimento na evolução das capacidades pessoais e profissionais dos seus membros de equipa oferecendo formação (INTUO Learn). A Intuo LMS apresenta semelhanças com as Plataformas de Gestão da Formação (seção 2.3) porque apesar de ser um SGA, foca-se muito na monitorização dos processos de formação.

Após um estudo sobre os *endpoints* disponibilizados pela API desta plataforma, em seguida são analisados os mais importantes no contexto deste projeto. Na Tabela 4 é apresentado um resumo dos *Web Services* estudados.

Tabela 4 – *Web Services* estudados através da Intuo API

URI	Operação na Base de dados	Parâmetro URL	Método HTTP	Resposta (BODY)
https://apitest.intuo.io/public/skills/{id}	Nenhuma (apenas lê)	{id}	GET	Figura 5
https://apitest.intuo.io/public/reporting/users/{id}	Nenhuma (apenas lê)	{id}	GET	Figura 6
https://apitest.intuo.io/public/student_groups	Nenhuma (apenas lê)	-	GET	Figura 7

Na Figura 5 é apresentada a resposta HTTP do primeiro *Web Service* estudado em formato JSON. Este foi obtido utilizando os dados apresentados na Tabela 4.

```

200 OK Metadata
{
  "skill": {
    "id": 63081,
    "name": "Team development",
    "color": "#eeeeee"
  }
}

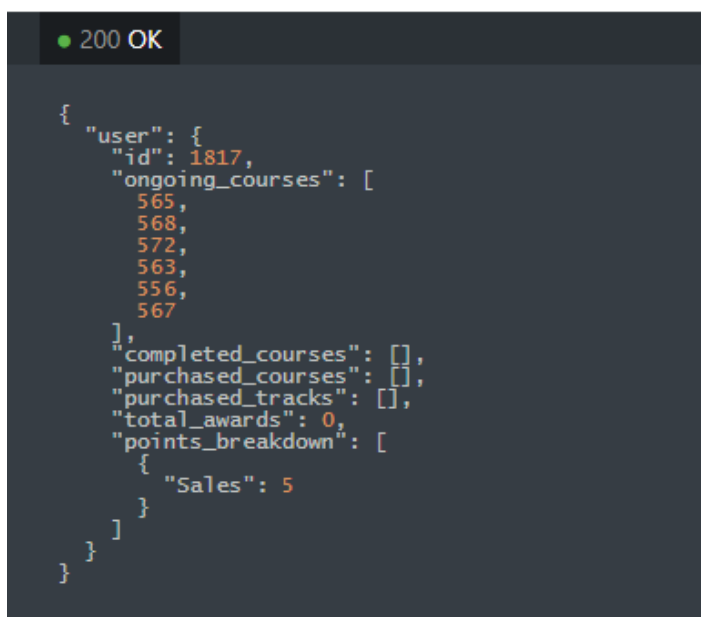
```

Figura 5 – Resposta HTTP em JSON dos dados da *skill* com id = 63081 (INTUO)

As *skills* permitem que os colaboradores alinhem os seus comportamentos às expectativas organizacionais. Para este objeto foi concluído que uma *skill* apresenta um “id” e um “name”.

Um conjunto de *skills* serve para identificar as capacidades, características e os conhecimentos de um colaborador. Este tipo de estatística ajuda as organizações a definir ou elevar as expectativas de desempenho dos seus colaboradores. Para este projeto, será necessária uma tabela SKILL.

Na Figura 7 é apresentada a resposta HTTP em formato JSON do segundo Web Service estudado, tendo por base os dados apresentados na Tabela 4.



```
● 200 OK
{
  "user": {
    "id": 1817,
    "ongoing_courses": [
      565,
      568,
      572,
      563,
      556,
      567
    ],
    "completed_courses": [],
    "purchased_courses": [],
    "purchased_tracks": [],
    "total_awards": 0,
    "points_breakdown": [
      {
        "Sales": 5
      }
    ]
  }
}
```

Figura 6 - Resposta HTTP em JSON dos dados do report do user com id = 1817 (INTUO)

O objeto da imagem anterior mostra o modo como se monitoriza a atividade de um utilizador da plataforma, isto é, dos cursos em progresso, completos, comprados e da soma dos pontos ganhos até ao momento. Isto possibilita aos *Administrators* gerar relatórios para procurar e controlar as atividades do “user” (colaborador).

Assim, para o desenvolvimento da plataforma deste projeto será necessário o relacionamento entre uma tabela REPORTING e uma tabela USER. Na Tabela 5 são apresentadas as propriedades que se devem destacar nessa relação.

Tabela 5 – Propriedades relevantes do objeto da Figura 6

Propriedades
id
ongoing_courses
completed_courses

Por último, na Figura 7 é destacada a resposta HTTP em formato JSON do estudo realizado para o terceiro Web Service.

```
● 200 OK
{
  "student_groups": [
    {
      "id": 121,
      "name": "MNGT QUA-ENV",
      "color": "#40b1d0",
      "user_ids": []
    },
    {
      "id": 122,
      "name": "MNGT QES",
      "color": "#40b1d0",
      "user_ids": []
    },
    {
      "id": 123,
      "name": "MNGT QES+",
      "color": "#40b1d0",
      "user_ids": [
        267,
        270,
        271,
        272,
        275,
        276,
        277
      ]
    }
  ]
}
```

Figura 7 – Resposta HTTP em JSON com dados dos grupos de utilizadores da plataforma (INTUO)

A tabela “STUDENT_GROUPS” ajuda os *Administrators* a selecionar colaboradores de uma organização com base nos seus atributos, tornando possível a atribuição de conteúdos de aprendizagem aos mesmos. Para além disso, os *Administrators* conseguem gerar relatórios sobre o progresso dos utilizadores cujo “id” aparece no *array* “user_ids”.

Por fim, com base no objeto da imagem anterior, é perceptível que uma tabela GROUP deva apresentar algum tipo de relação com uma tabela USER. Isso permite identificar os utilizadores pertencentes a um grupo através dos seus “user_ids”.

2.5.3 PowerSchool Student Management System (SIS)

PowerSchool é um sistema de informação de estudantes que permite aos professores tomar decisões que afetam o desempenho do aluno, criando um ambiente colaborativo para pais, professores e alunos. Este sistema é utilizado pelas escolas públicas de North Carolina para analisar, tratar e armazenar dados de estudantes, de forma segura, através da Internet (Cameron, 2011).

Este sistema mantém mais de 1.5 milhões de registos de alunos e oferece recursos para professores e administradores gerirem efetivamente os processos escolares (Cameron, 2011). Isto reforça a aprendizagem personalizada, uma vez que, é possível adaptar o caminho pedagógico de um aluno aos seus interesses, habilidades e objetivos (Cameron, 2011). Contudo, o que distingue este sistema dos outros, é o portal dos pais (PowerSchool Parents) (Cameron,

2011). Este portal permite consultar as notas e tarefas dos alunos. No contexto deste projeto, é do interesse consultar as classificações e tarefas dos colaboradores e clientes.

Como abordado nas secções anteriores, atualmente, existem programas relativamente simples na Internet que permite aos formadores partilhar não somente os seus apontamentos, mas também as suas tarefas, leituras complementares, vídeos instrutivos e outros textos que podem ser utilizados para promover a aprendizagem dos utilizadores (Cameron, 2011). Consultar as classificações dos seus educandos é a funcionalidade que atrai os encarregados de educação a utilizar a plataforma apresentada nesta secção.

Assim, a palavra chave para descrever um Sistema de Informação de Estudantes (*Student Information System, SIS*) é a integração, uma vez que, permite a inclusão de componentes que são considerados como aplicações independentes (Cameron, 2011). O PowerSchool integra componentes padrão, nomeadamente, geração de cronogramas numa única solução através de uma UI consistente. Esses componentes partilham a mesma base de dados e as respetivas informações ficam imediatamente disponíveis, de igual forma para todos, com segurança (Cameron, 2011).

Face ao exposto, a Figura 8 ilustra os componentes que compõe a arquitetura do PowerSchool SIS.

PowerSchool Architecture

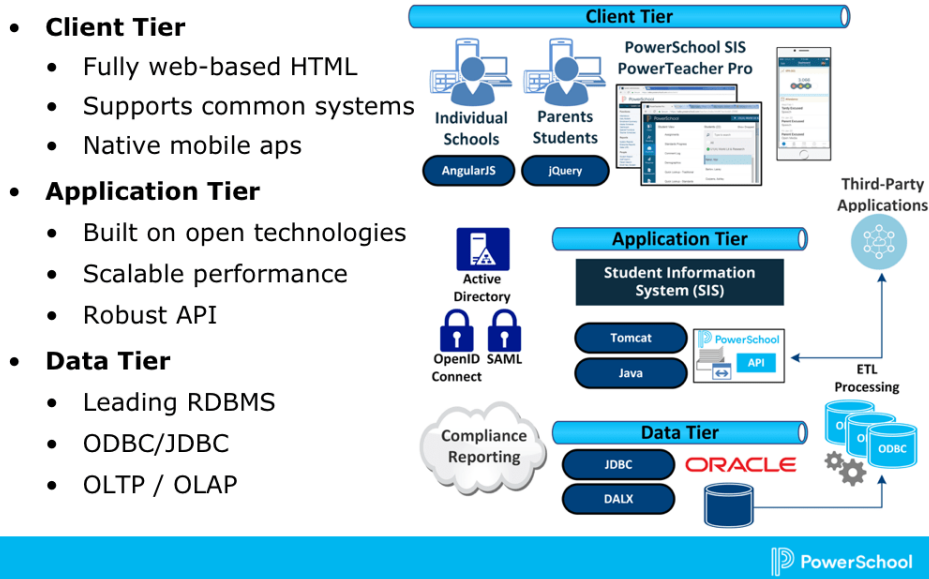


Figura 8 – Visão geral da arquitetura do PowerSchool SIS (PowerSchool)

O sistema em análise é baseado na arquitetura em 3 camadas que é hierarquizada da seguinte forma (Three-tier architectures, 2019):

- Camada do Cliente (*Client Tier*);
- A camada de Aplicação (*Application Tier*);
- A Camada de Dados (*Data Tier*).

Este tipo de arquitetura foi desenvolvido devido à regularidade com que é utilizado o sistema cliente-servidor em aplicações (PowerSchool). As arquiteturas em 3 camadas oferecem muitos benefícios para os ambientes de desenvolvimento e produção, pois permitem dividir em partes distintas (modularização) as camadas referidas anteriormente (Three-tier architectures, 2019). Isto permite que as equipas de desenvolvimento atualizem partes da aplicação sem afetar as restantes (Three-tier architectures, 2019). Essa flexibilidade diminui substancialmente os tempos do ciclo de desenvolvimento, oferecendo às equipas a capacidade de substituir ou atualizar camadas independentes sem afetar as outras partes do sistema (o portal PowerSchool Students não deve afetar o portal PowerSchool Parents e vice-versa) (Three-tier architectures, 2019).

A camada do cliente apresenta uma UI que inclui HTML com AngularJS e jQuery (PowerSchool). Neste sistema, é aplicado o padrão MVC (PowerSchool). Este é utilizado em projetos devido à arquitetura que possui, uma vez que, possibilita a divisão dos mesmos em camadas definidas, onde cada uma delas, o *Model*, *Controller* e *View*, executam unicamente aquilo para a qual foram definidas (Three-tier architectures, 2019). A aplicação apresenta ainda um *design web* responsivo, técnica utilizada para responder à forma como os elementos que compõem um website se dispõem automaticamente à largura do ecrã do dispositivo no qual está a ser visualizado.

A camada da aplicação é construída sobre tecnologias *open-source*, nomeadamente, Apache Tomcat e Java (PowerSchool). Tomcat, vinculado à linguagem de programação Java, é um servidor que permite a execução de aplicações para web (PowerSchool). Esta camada utiliza ainda processamento de transações online (*Online Transaction Processing*, OLTP) para distribuição de informação geral da *web* e processamento analítico online (*Online Analytical Processing*, OLAP) para relatórios (PowerSchool). Além disso, a camada possui uma API RESTful para integrações (PowerSchool).

A camada de dados utiliza o Oracle 12c Enterprise Edition (PowerSchool), o principal sistema de gestão de base de dados relacional, que é compatível com dois tipos: *Open Database Connectivity* (ODBC) e *Java Database Connectivity* (JDBC). A ideia do primeiro é desenvolver uma solução comum de acesso utilizando SQL, de forma a que uma aplicação cliente pudesse aceder à base de dados de fornecedores (Milener, Hamilton, & Rabeler, 2017). Quanto ao segundo, é um conjunto de classes e interfaces em Java, que proporcionam uma interface similar à ODBC para bases de dados SQL (Souza, 2007). Logo, ODBC é utilizada em plataformas Windows e JDBC em qualquer tipo de sistema.

Por fim, a camada de dados para obter um desempenho escalável, utiliza OLAP e OLTP (PowerSchool). Quanto ao primeiro, é uma categoria de ferramentas de *software* que fornece

análise de dados para decisões de negócio e permitem aos utilizadores analisar informações da base de dados de vários sistemas (Oliveira, 2018). Quanto ao segundo, suporta aplicações orientadas a transações numa arquitetura de três camadas. Este administra as transações diárias de uma organização (Oliveira, 2018). Logo, OLTP para atividades de base de dados gerais e OLAP para descrever a atividade da base de dados.

2.5.4 Adobe Captivate Prime

O Adobe Captivate Prime é uma solução corporativa para sistemas de gestão de aprendizagem. Este oferece recursos de gestão de aprendizagem e permite a participação online através de um browser e ligação. Com um serviço baseado na *cloud*, o Adobe Captivate Prime permite que o administrador adicione e registe utilizadores numa conta, crie e atribua cursos de e-learning e programas para utilizadores (Adobe, 2017). Os autores dos cursos podem carregar para o Captivate Prime vários formatos de conteúdo, nomeadamente, PowerPoint, PDF, Word, vídeo, entre outros (Adobe, 2017).

2.5.4.1 Arquitetura da Aplicação

Adobe Captivate Prime é uma solução baseada na *cloud* que permite a independência dos processos, como apresentação, processamento de aplicações e gestão de dados (Adobe, 2017). Estes processos são executados em vários servidores conforme ilustrado na Figura 9: Business Logic Servers, Learning Record Servers, entre outros. Cada servidor oferece os respetivos serviços conforme as necessidades dos utilizadores do LMS, que incluem *Administrators*, *Trainers* e *Trainees*. Exemplos de funções fornecidas pelo Captivate Prime são a criação de conteúdo, gestão de utilizadores e grupos, autorização e autenticação, entre outras (Adobe, 2017).

Na Figura 9 são ilustradas as componentes que compõe a arquitetura da aplicação.

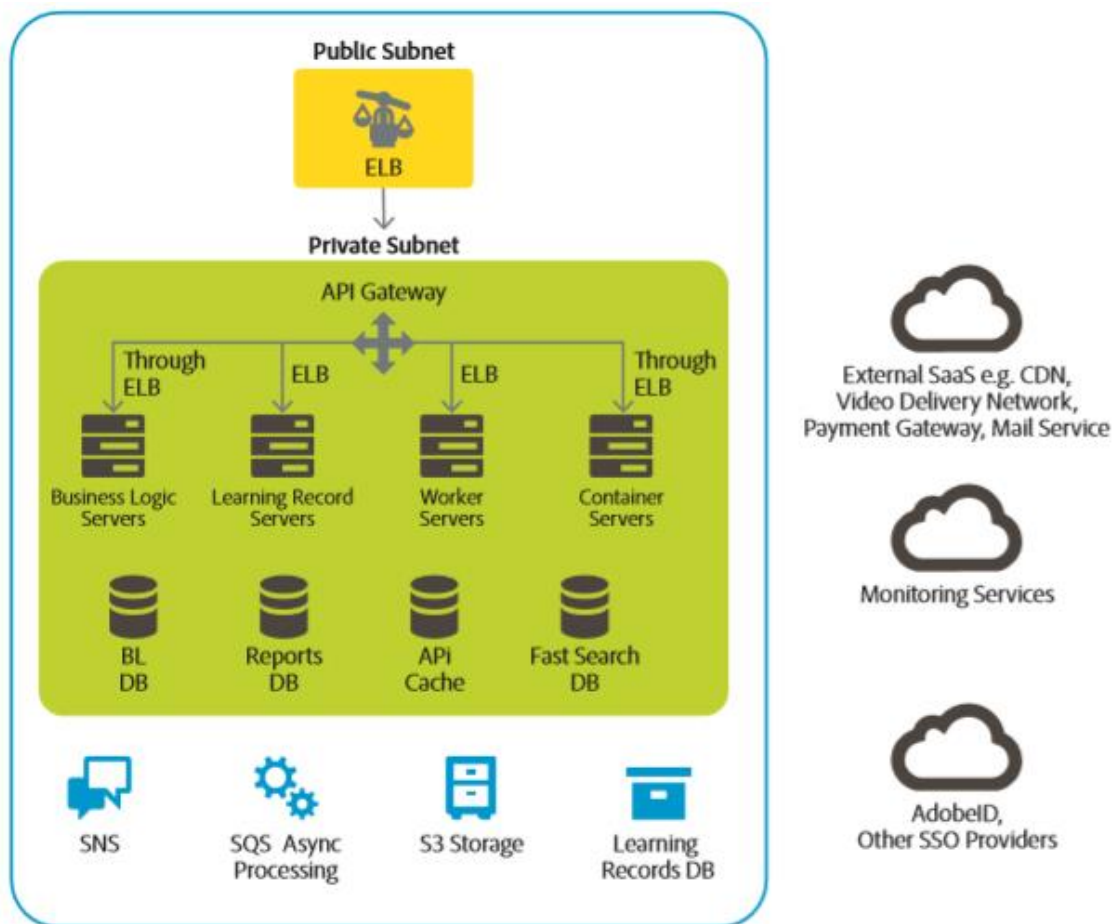


Figura 9 – Arquitetura da aplicação Adobe Captivate Prime (Adobe, 2017)

Para que o Adobe Captivate Prime funcione, são necessárias 6 componentes (Adobe, 2017):

- Adobe Captivate Prime Business Logic Server – permite criar e gerir utilizadores, inscrições, grupos de utilizadores e *learning objects* (cursos, programas e certificações);
- Adobe Captivate Prime Learning Record – gere os relatórios de aprendizagem enquanto os alunos completam cursos;
- Adobe Captivate Prime Worker Server – executa todas as tarefas assíncronas, como por exemplo, conversão de conteúdo do curso, importação de listas de utilizadores a partir de ficheiros Excel e geração de relatórios grandes e específicos e/ou personalizados;
- API Gateway Server – gere o acesso dos utilizadores aos recursos e determina a autenticação do utilizador, como por exemplo, autores não podem criar cursos, no entanto, administradores tem permissão para adicionar utilizadores;
- Containers Servers – hospeda serviços múltiplos, API públicas e OAuth;

- Fluidic Player – permite que os conteúdos de aprendizagem sejam reproduzidos nos dispositivos dos utilizadores.

2.5.4.2 Fluxo de Dados

O Adobe Captivate Prime suporta 5 grupos de utilizadores que interagem com o Sistema, onde cada qual consome e fornece diferentes dados (Adobe, 2017):

- *Administrators* – tem acesso aos dados dos utilizadores via CSV ou adicionam utilizadores manualmente preenchendo campos, como nome ou e-mail;
- *Authors* – criam cursos de e-learning sempre que importam conteúdo novo, nomeadamente, ficheiros PDF, vídeos, ficheiros Word, entre outros;
- *Trainees* – fazem cursos mediante os seus interesses ou com base nos interesses dos seus administradores. Assim, o Adobe Captivate Prime guarda a interação entre aluno e curso permitindo desenvolver gráficos que permite aos RH de uma organização acompanhar a evolução dos seus colaboradores e, posteriormente, gerar relatórios de acompanhamento;
- *Managers* – visualizam unicamente esses gráficos com dados estatísticos personalizados sobre os membros das suas equipas.

A Figura 10 ilustra todo o processo de fluxo de dados no sistema Captivate Prime, incluindo servidores onde os dados são armazenados e como os mesmos são consumidos. Cada linha apresenta uma cor que descreve o fluxo de dados que entram e saem do sistema (Adobe, 2017).

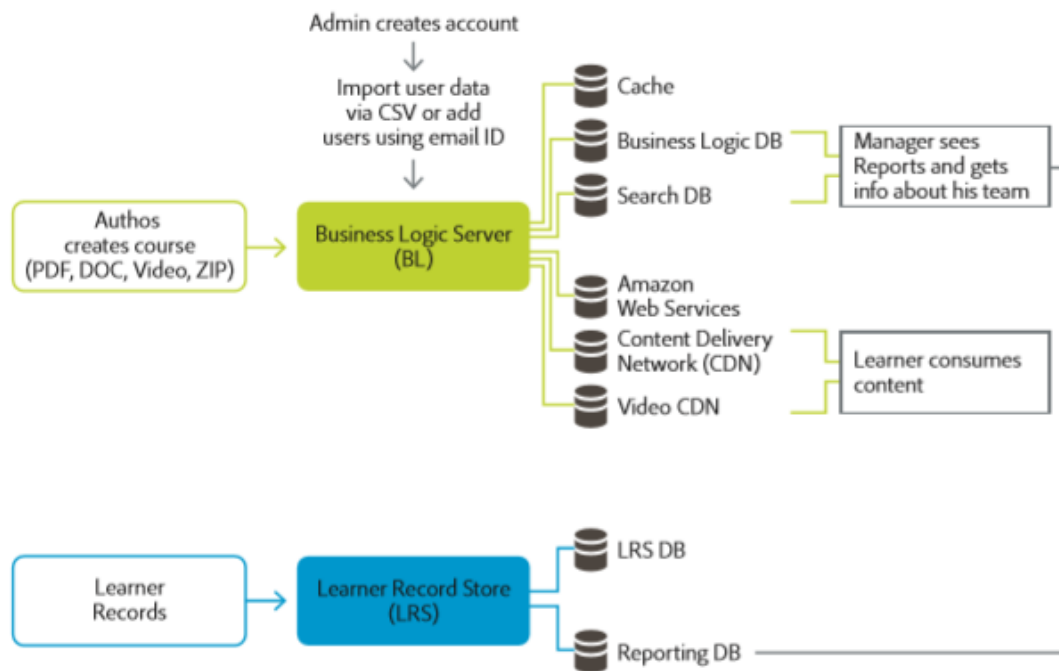


Figura 10 – Fluxo de dados da Adobe Captivate Prime (Adobe, 2017)

Reforçar que qualquer ligação com o sistema analisado nesta secção é enviada por HTTPS (*HyperText Transfer Protocol Secure*) utilizando *Secure Sockets Layer* (SSL) (Adobe, 2017). O HTTPS é um protocolo que permite que o computador troque informações com o servidor que hospeda o website. O SSL permite que uma aplicação do tipo cliente-servidor comunique pela rede de forma segura, protegendo a integridade e veracidade do conteúdo que circula na Internet. Qualquer comunicação com terceiros também é enviada por HTTPS (Adobe, 2017).

2.5.4.3 Arquitetura de Segurança

O Adobe Captivate Prime encontra-se hospedado nos servidores Web da Amazon (Amazon Web Services, AWS) numa Amazon Virtual Private Cloud (Amazon VPC). Qualquer conteúdo que seja fornecido pelo utilizador (nomeadamente, ficheiros, imagens, entre outros) fica disponível por meio de uma camada de autorização e apenas pode ser acedido por utilizadores devidamente autorizados (Adobe, 2017).

Por fim, sobre as bases de dados do Captivate Prime, encontram-se na Amazon VPC e são acedidas apenas por servidores de aplicações autorizadas. Estas base de dados contêm algumas camadas com códigos de segurança que reforçam a não manipulação dos dados, bem como código que ajuda a impedir o acesso de utilizadores não autorizados (Adobe, 2017). Um utilizador da Captivate Prime apenas e somente tem acesso à conta que lhe pertence (Adobe, 2017).

2.5.5 Comparações e Conclusões sobre os SGA Analisados

Em cada um dos exemplos desta secção (2.5), observamos soluções simples, no entanto, parcialmente distintas. Em seguida, são apresentados os pontos mais relevantes após uma análise estruturada e detalhada:

- As plataformas TalentLMS e Intuo LMS mostram um modelo de domínio idêntico, embora a primeira apresente um número de funcionalidades significativamente maior;
- Um sistema que utilize a arquitetura em 3 camadas (PowerSchool) é considerado um sistema monolítico do lado do servidor. Isso significa que o código é mantido no mesmo servidor e só funciona se for implantado como uma unidade única. Assim, qualquer alteração, por mais mínima que seja, requer que toda a aplicação seja reconstruída e reimplantada;
- Contrariamente ao ponto anterior, a arquitetura de micro serviços (Adobe Captivate Prime) permite que a aplicação seja dividida e projetada como serviços independentes, pouco acoplados e modelados em torno de um modelo de negócios personalizado. O ponto mais importante é que cada serviço possui um limite de negócios que pode ser desenvolvido, testado, implantado, monitorizado, dimensionado de modo independente e desenvolvidos em diferentes linguagens de programação;
- Adobe Captivate Prime apresenta os mesmos grupos de utilizadores identificados na secção 2.3.3, bem como a melhor infraestrutura.

2.6 User Interface e Funcionalidades de SGA e TMS

Uma UI é o termo utilizado para definir como uma aplicação ou *software* interage com o utilizador. De forma a compreender como se encontram dispostas visualmente as principais páginas e funcionalidades de um LMS e de uma plataforma de gestão da formação, em seguida, são apresentados alguns estudos efetuados sobre as aplicações. Mediante esses estudos são elaboradas conclusões permitindo clarificar como podem ser abordadas essas mesmas análises no desenvolvimento deste projeto. As observações recaem sobre barra de navegação e página inicial após utilização da plataforma Adobe Captivate Prime e pesquisa de imagens da TalentLMS.

2.6.1 Barra de Navegação

Uma barra de navegação é considerada como o elemento da UI mais importante, uma vez que, contém hiperligações para outras secções do site e apresenta dados estatísticos. Com base no estudo realizado nas secções 2.3 e 2.5, foram escolhidas as seguintes barras de navegação da TalentLMS (Figura 11) e FORINSIA (Figura 12).

Figura 11 – *Navbar* TalentLMS (Pappas, TalentLMS Overview)



Figura 12 - *Navbar* FORINSIA de um administrador

Face ao exposto, todas as barras de navegação apresentam o nome do utilizador. No caso particular da TalentLMS, esse campo é uma hiperligação para uma página onde se pode visualizar e editar as informações do utilizador autenticado. Esta barra apresenta ainda o perfil de utilizador.

Relativamente à FORINSIA, são destacados os seguintes menus: Utilizadores (área de gestão de utilizadores e instituições), Cursos, Inquéritos e Utilizador (área pessoal).

Outra comparação é a área de pesquisa por texto na Figura 11 que permite ao utilizador procurar conteúdo na aplicação sem necessidade de carregar em várias hiperligações.

2.6.2 Página Inicial com Perfil de *Administrator*

A página inicial é a primeira que um utilizador autenticado visualiza após o processo de autorização e autenticação. Esta página é importante porque deve “entusiasmar” e atrair o utilizador a navegar pelas páginas internas do site ou aplicação.

Conforme ilustrado na Figura 13, a TalentLMS apresenta hiperligações para funcionalidades como USERS, COURSES, GROUPS, USER TYPES, entre outros. Para além disso, é apresentado um gráfico que indica o número de utilizadores que utilizam a plataforma e os cursos completos. Esta filtragem pode ser efetuada por dia, semana ou mês.

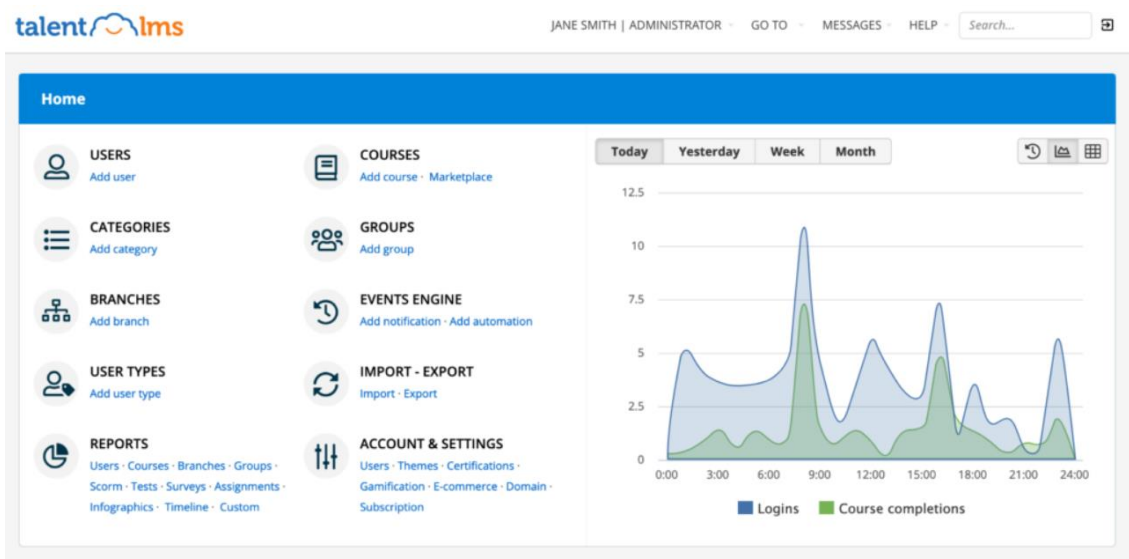


Figura 13 – Homepage de um administrador TalentLMS (Pappas, TalentLMS Overview)

Na Figura 14 é apresentado o *layout* para o Adobe Captivate Prime. Apesar de manter a identidade de destacar as funcionalidades mais importantes (como por exemplo, adicionar curso e gerir *skills*), a sua página inicial não apresenta gráficos. Em contrapartida, possui uma hiperligação (“View Reports”) para uma página com gráficos personalizados. Para além disso, apresenta um Menu fixo à esquerda.

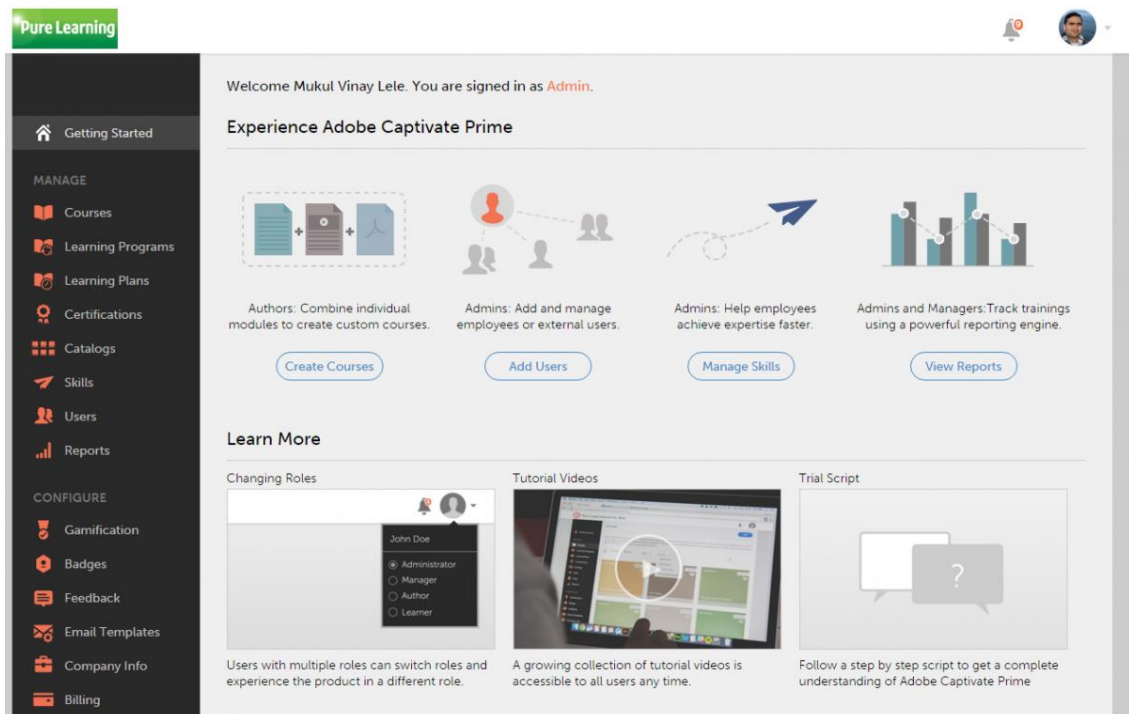


Figura 14 – Homepage de um administrador Adobe Captivate Prime (Pappas, Adobe Captivate Prime)

Em suma, o Adobe Captivate Prime apresenta um *layout* significativamente melhor que o TalentLMS, pois com menu não existe necessidade de voltar à página inicial tornando tudo um pouco mais funcional e menos confuso.

2.6.3 Gestão de Inscrições

A Figura 15 mostra a página de gestão de inscrições. Esta é uma das funcionalidades principais num TMS, uma vez que, permite gerir os utilizadores previstos, inscritos e que compareceram à formação. Para este projeto, esta funcionalidade deve permitir a filtragem de cursos por instituições.

Nome	Curso	Ação	Data início processo	Data início Ação	Localidade	Documentos	Pagamento	Confirmado	Inscrito
ABEL DE MOURA RODRIGUES	Teste (msg) (Mensagens)	1	07-09-2012	01-09-2007		✓	!	✗	✗
ABEL FERNANDO MARTINS TEIXEIRA	Testes Módulo Avaliação (Testes Módulo Avaliação 01)	Testes do Módulo de Avaliação	06-09-2012	01-04-2009		✓		✗	✗
ABEL FERNANDO MARTINS TEIXEIRA	Teste Ticket 270 (2012 empresas c/empresa)	4	08-10-2012	18-10-2012	Leça da Palmeira	!	!	✗	✗
Abel Leitão Buco	Teste (msg) (Mensagens)	1	07-09-2012	01-09-2007		✓	!	✗	✗
Abel Leitão Buco	Testes Módulo Avaliação (Testes Módulo Avaliação 01)	Testes do Módulo de Avaliação	06-09-2012	01-04-2009		✓		✗	✗
ABEL PEREIRA TEIXEIRA	Formação Forfnia (2011)	2	08-10-2012	03-11-2011	São Mamede Coronado	!		✗	✗
ABEL PEREIRA TEIXEIRA	Teste Ticket 270 (2012 empresas c/empresa)	4	08-10-2012	18-10-2012	Leça da Palmeira	!	!	✗	✗
ABEL PEREIRA TEIXEIRA	Teste Ticket 270 (2012 empresas)	5	08-10-2012	18-10-2012	Leça da Palmeira	!	!	✗	✗

Figura 15 - Gestão de inscrições na plataforma FORINSIA

2.7 Comparações e Conclusões sobre Funcionalidades

Com base no estudo efetuado nas secções 2.3 e 2.5, foi elaborada a Tabela 6 que expõe uma lista de funcionalidades oferecidas pelos LMS e TMS.

Tabela 6 – Funcionalidades oferecidas pelos sistemas de gestão de aprendizagem e sistema de gestão de formação

Funcionalidades	Plataformas				
	TalentLMS	Intuo	PowerSchool (SIS)	Captivate Prime	FORINSIA
Cursos	Sim	Sim	Sim	Sim	Sim
Campos personalizados	Sim	Sim	Não	Sim	Sim
Relatórios administrativos	Sim	Sim	Sim	Sim	Sim
Catálogo de cursos	Sim	Sim	Sim	Sim	Sim

Funcionalidades	Plataformas				
	TalentLMS	Intuo	PowerSchool (SIS)	Captivate Prime	FORINSIA
Activity History (utilizadores)	Não	Não	Sim	Sim	Sim
Relatórios personalizados	Sim	Sim	Sim	Sim	Sim
Importação/Exportação de dados	Sim	Não	Sim	Sim	Sim
Permissões baseadas em roles	Sim	Sim	Sim	Sim	Sim
Learning Plans	Não	Não	Não	Sim	Não
Classificação	Sim	Sim	Sim	Sim	Sim
Levels	Sim	Não	Não	Sim	Não
Points	Sim	Sim	Não	Sim	Não
Planos individuais	Sim	Sim	Não	Sim	Sim
Página inicial personalizada	Sim	Não	Não	Sim	Sim
Skills	Não	Sim	Não	Sim	Não
Notificações	Sim	Sim	Sim	Sim	Sim
Certificações	Sim	Sim	Não	Sim	Sim
Certificações personalizadas	Não	Não	Não	Sim	Sim
Dashboard	Sim	Sim	Sim	Sim	Sim
Gestão de clientes	Não	Não	Não	Não	Sim
Arquiva documentos	Não	Não	Não	Sim	Sim
Gestão de salas de aulas	Não	Não	Não	Não	Sim
Presenças	Sim	Sim	Sim	Sim	Sim
Sumários	Não	Não	Sim	Não	Sim
Autenticação	Sim	Sim	Sim	Sim	Sim
Gestão de candidaturas	Não	Não	Não	Não	Sim
Monitorizar presenças	Sim	Não	Sim	Sim	Sim
Blended Learning	Sim	Sim	Não	Sim	Sim
Registo histórico da formação externa por formando	Não	Não	Não	Não	Sim

Funcionalidades	Plataformas				
	TalentLMS	Intuo	PowerSchool (SIS)	Captivate Prime	FORINSIA
Inscrição automatizada	Sim	Sim	Sim	Sim	Sim
Auto inscrição	Sim	Sim	Sim	Sim	Sim
Dupla confirmação de presenças dos formandos	Não	Não	Não	Não	Não

Através da tabela, a plataforma FORINSIA apresenta algumas funcionalidades que diferem dos 4 LMS, como por exemplo, registo histórico da formação externa por formando, gestão de salas de aula, gestão de clientes, certificações personalizadas (apenas o Adobe Captivate Prime oferece). Contudo, também existem funcionalidades oferecidas pelos 4 LMS que a plataforma de gestão da formação não tem, nomeadamente, *skills*, *learning plans*, *points* e *levels*.

Como é possível observar, os LMS detêm funcionalidades que podem ser aplicadas em contexto de formação em sala. Um dos juízos principais a formular após observação da tabela anterior, é precisamente esse, que o e-learning não depende da formação em sala, no entanto, a complementaridade entre os dois termos existe. Apesar disso, o LMS tem funcionalidades que ajudam a formação em sala, mas o contrário não se verifica.

Em síntese, para desenvolver uma plataforma de gestão da formação é necessário conhecer os processos administrativos de uma organização que permitem desenvolver funcionalidades específicas, no entanto, será necessário sincronizar os dados do LMS com a plataforma de gestão da formação, pois só assim é possível obter mais informação e veracidade aquando a realização de gráficos, relatórios e certificados personalizados.

3 Análise de Valor

Nesta secção é apresentado o processo de análise de valor relacionado com o produto em desenvolvimento, bem como os métodos, ferramentas e técnicas utilizadas para identificar o valor do produto e as funcionalidades principais que geram valor para o cliente ou utilizador. O modelo escolhido para elaborar esta análise de valor é o *Value Engineering* (VE). Em resumo, o *Value Analysis* (VA) refere-se à análise de um produto, serviço ou processo que já existe, enquanto que o VE se refere à mesma análise aplicada ao produto, serviços ou processos administrativos que estão a ser projetados e não foram finalizados (Rich & Holweg, 2000).

O *Value Analysis* (VA) é uma abordagem projetada para redução de custos que analisa a composição do material de um produto e design de produção, para que modificações e melhorias possam ser efetuadas e que não reduzam o valor do produto para o cliente ou utilizador (ou seja, a qualidade para fins não deve ser comprometida) (Rich & Holweg, 2000). Portanto, o valor e a qualidade de um produto devem ser mantidos, ou aperfeiçoados, a um custo reduzido.

O foco principal da abordagem VA é, portanto, a gestão da funcionalidade para gerar valor para o cliente (Rich & Holweg, 2000). Se uma organização pretende reduzir os custos de produção, deve procurar custos ou partes do produto que não acrescentam valor funcional ao cliente (Rich & Holweg, 2000).

Assim, o objetivo do VA é formal e sistemático. Algumas das fases mais importantes que podem ser implementadas são representadas na Figura 16: Orientação, Análise Funcional (que também inclui a fase de Identificação Funcional), Criação de Alternativas, Análise e Avaliação, Implementação.

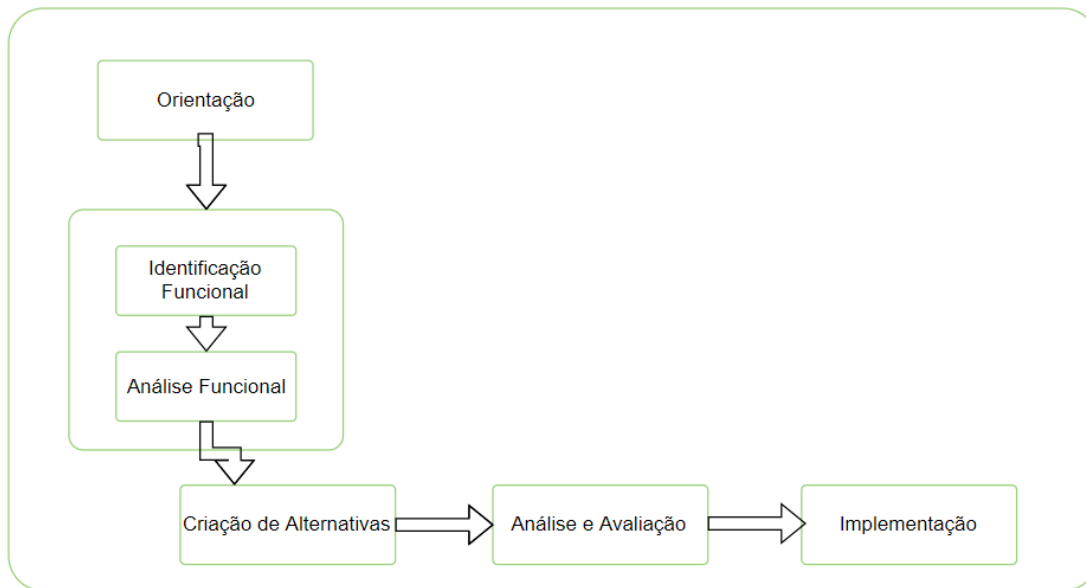


Figura 16 – Processo Análise de Valor. Adaptado de Nick Rich e Matthias Holweg (Rich & Holweg, 2000)

A primeira etapa é a fase de Orientação, que tem como objetivo identificar e selecionar o produto ou serviço que irá ser desenvolvido permitindo entender melhor o problema. A fase da Análise Funcional é um componente essencial do VA, é uma técnica para identificar e entender as necessidades do produto ou serviço (o que faz, o que deve fazer). A terceira etapa, Criação de Alternativas é uma oportunidade para criar ideias num curto período de forma a produzir meios alternativos para executar as funções associadas ao produto ou serviço. A fase de Análise e Avaliação é a quarta fase do processo de VA, aqui as ideias são sistematicamente avaliadas, priorizadas e pré-selecionadas quanto ao seu potencial de economia de custo e/ou valor geradas a partir da fase Criativa. A última fase, de Implementação, tem como objetivo a descrição do desenvolvimento do produto.

3.1.1 Orientação

Para a primeira fase do processo de VA foi aplicado o modelo *Fuzzy Front End* (FFE), que é geralmente considerado como uma das maiores oportunidades de melhoria do processo de inovação global. A atenção está cada vez mais focada nas atividades de *front-end* que precedem este processo formal e estruturado, a fim de aumentar o valor, a quantidade e a probabilidade de sucesso de conceitos de alto lucro ao entrar no desenvolvimento de produtos e comercialização. Assim, neste processo é possível identificar três etapas: o FFE, o *New Product Development* (NPD) e a comercialização.

Peter Koen desenvolveu o *New Concept Development* (NCD) a fim de proporcionar uma linguagem comum para descrever os componentes do *Fuzzy Front End*, apresentado na Figura 17.

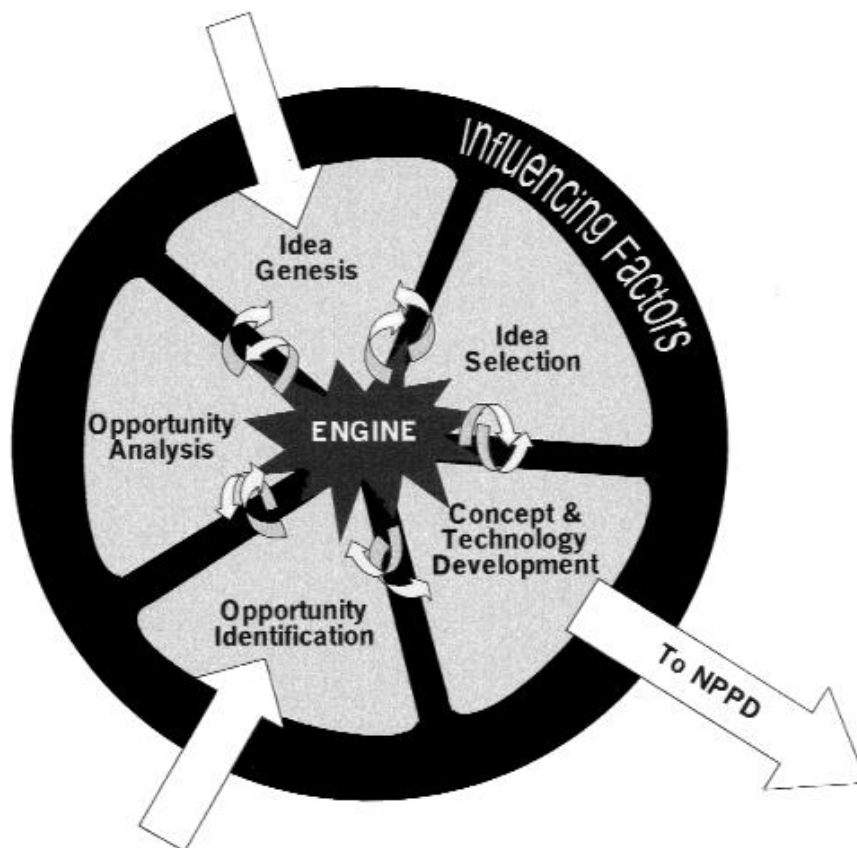


Figura 17 – *New Concept Development* (Koen, 2001)

O modelo NCD consiste em três partes principais:

- Fatores de influência;
- O Motor que representa a liderança, a cultura e a estratégia de negócios da organização impulsionam os cinco elementos principais;
- Os cinco elementos do FFE – Identificação de Oportunidades, Análise de Oportunidades, Geração de ideias, Seleção de Ideias e Definição de Conceito.

Observando a Figura acima, muitas características do modelo NCD destacam-se, nomeadamente, as partes internas também chamadas de elementos, em oposição a processos. O modelo apresenta ainda uma forma circular, que tem o propósito de sugerir que as ideias devem fluir, circular e iterar entre os cinco elementos. Seguidamente, são apresentados os elementos do NCD no contexto deste projeto.

3.1.1.1 Fatores de influência

O FFE existe num ambiente com fatores de influência. Os fatores são as capacidades organizacionais da empresa, a influência de clientes e concorrentes, e a tecnologia (é um

facilitador quando se pode utilizar repetidamente um produto ou serviço). Os recursos organizacionais determinam se e como as oportunidades são identificadas e analisadas, como as ideias são selecionadas e geradas, e como os conceitos e tecnologias são desenvolvidos. O mundo exterior, a política, os regulamentos ambientais, as leis e as tendências socioeconômicas também afetam o FEE.

Face ao exposto no parágrafo anterior, os fatores de influência a considerar segundo o contexto deste projeto são as capacidades organizacionais da empresa, a influência dos seus clientes e concorrentes, a tecnologia bem como as ferramentas que existem no mercado e melhor se adequam às necessidades do projeto.

A ALERT, empresa que se insere na área da saúde, recolhe ininterruptamente dados de fontes externas através de notícias (televisão ou aplicações que atualizam ao minuto) ou até mesmo, conferências. Combinar estes dados permite originar informação que determina se existe necessidade de mudar, solucionar algum problema ou aproveitar oportunidades que impactem os produtos da empresa na área da saúde. Posto isto, é necessário acrescentar, eliminar, modificar ou atualizar funcionalidades dos produtos da ALERT. Assim, estes fatores de influência externos têm impacto no contexto deste projeto uma vez que adicionar, remover, atualizar ou modificar funcionalidades, leva a que os clientes e utilizadores obtenham nova formação sobre os produtos. Logo, utilizando este projeto será possível elaborar planos de formação para todos os interessados. Além disto, a capacidade que a empresa demonstra aquando da necessidade de se adaptar às alterações provenientes do mundo exterior, respeitando as normas de saúde impostas, permite mostrar-se confiável.

Os clientes da ALERT são considerados como um outro fator de influência uma vez que a sua adesão e satisfação são indispensáveis para que a empresa progrida diariamente. Atualmente, a ALERT possui uma lista de clientes diversificada a nível internacional, nomeadamente, hospitais, clínicas de saúde (públicas ou privadas), entre outras instituições de saúde. Assim, para que seja possível atender às necessidades dos seus colaboradores, para que estes demonstrem satisfação e confiança relativamente à capacidade de resposta do produto, é necessário que exista contacto diário com os clientes e utilizadores.

No seguimento do parágrafo anterior, a ALERT não é a única empresa que possui produtos clínicos e, por esse motivo, é necessário prestar atenção aos concorrentes. O mercado onde a empresa se encontra inserida é bastante competitivo e as grandes vantagens da utilização de sistemas de informação (SI) deste tipo são, a facilidade, velocidade e quantidade e informação relativa aos utentes que os profissionais de saúde podem obter; os sistemas de apoio nas decisões, que suportam as decisões clínicas mostram a relevância de um bom SI para obter níveis de qualidade altos. Quando não se tem um acesso adequado a dados, praticamente não se podem tomar decisões sobre os diagnósticos e nos procedimentos, pois pode resultar em problemas graves para os pacientes.

Desta forma, processar sistematicamente dados e informação incrementa a qualidade e a eficiência no cuidado de saúde. Não obstante, é necessário estar ao corrente de modificações

nas legislações, regras e nos padrões de saúde uma vez que tem impacto direto nos produtos da ALERT. Estes fatores são a principal razão pela qual a empresa necessita de estar em constante monitorização relativamente a eventos externos para que seja possível alterar as funções dos seus produtos com rapidez, sem comprometimento e eficácia.

Em último, com o avançar dos tempos é perceptível a evolução tecnológica e uma vez que os produtos clínicos dependem da mesma, esta é considerada outro fator de influência externa. Geralmente, para que não exista necessidade de alterar sucessivamente os SI na sua totalidade, pode ser necessário principiar métricas a nível arquitetural ou de design para que no futuro, estes sistemas continuem a responder de forma adequada e rápida aos profissionais de saúde. Assim, é necessário estar sempre atento à chegada de novas tecnologias.

3.1.1.2 Motor

Como referido anteriormente, o Motor que representa a capacidade de liderança, a cultura e a estratégia de negócios da organização impulsionam os cinco elementos principais do NCD (Koen, 2001).

Sendo que a missão da ALERT é “melhorar a saúde e prolongar a vida, alcançar rentabilidade para benefício da sociedade e inspirar outros para a excelência através do nosso exemplo” (Computing, 2019), e com base naquilo que tem vindo a ser mencionado, a empresa é dividida por equipas de tamanho semelhante, que em conjunto cooperam para que os produtos se encontrem operacionais e ofereçam funcionalidades que permitam auxiliar no cuidado de saúde dos utentes. Não obstante, cada equipa é composta por um líder que através de reuniões define tarefas e prazos de entrega de forma a que o produto final seja melhorado e atenda às exigências do cliente.

3.1.1.3 Identificação de oportunidade

A identificação de oportunidade ocorre quando a organização reconhece a necessidade de desenvolver um produto que corresponda aos desenvolvimentos tecnológicos dos consumidores e do mundo, e a ameaça competitiva nesta área quando outras empresas decidem investir no mesmo (Koen, 2001).

A ALERT apresenta um vasto leque de produtos clínicos que apoiam na tomada de decisão relacionada com a própria vida. Como consequência, é necessário assegurar a formação de conteúdos relativos a esses produtos e, para isso, destaca-se o produto ALERT® eLEARNING (um SGA) que oferece aos utilizadores uma formação completa em diversos produtos clínicos ALERT®, que através da ligação à Internet podem consultar conteúdos sobre os mesmos. Esses conteúdos expõem o funcionamento do produto clínico.

Paralelamente à formação realizada através do ALERT® eLEARNING, as metodologias de formação em sala continuam a obter uma presença significativa nos projetos da ALERT, onde a

documentação em papel contínua a ser um dos recursos usados para meio de registo dessas atividades. Neste sentido, a empresa identificou uma nova oportunidade para desenvolver um produto independente que suporte os processos de formação em sala. Estes processos devem ser informatizados e otimizados para que se apresentem mais ágeis, eficazes e produtivos, visando funcionalidades de planeamento, gestão de inscrições, confirmação de presenças dos formandos, avaliação de aprendizagem e monitorização da formação. Assim, apesar da disponibilização de conteúdos proporcionada pelo ALERT® eLEARNING, é necessário um outro produto que, por exemplo, possibilite a criação de planos de formação para que os consultores da ALERT possam dar formação presencial aos seus colaboradores.

Em suma, visualizar documentos ou assistir a vídeos que explicam o funcionamento dos produtos ALERT nem sempre é suficiente para os colaboradores e, por este motivo, a identificação desta nova oportunidade permitirá eliminar esse problema ao elaborar planos de formação em sala mais diversificados, menos propenso a erros e maior rapidez aquando a alteração de detalhes.

3.1.1.4 Análise da oportunidade

A análise da oportunidade surge quando a empresa analisa ao pormenor aquilo em que os consumidores se encontram realmente interessados (Koen, 2001).

Segundo Woodall (Conceptualising 'Value for the Customer', 2003) afirmou: “o *cliente value* (VC) é qualquer percepção pessoal que surge da associação de um cliente com a oferta de uma organização e pode ocorrer como redução do sacrifício; presença de benefícios (percebidos como atributos); o resultado de qualquer combinação ponderada de sacrifício e benefício (determinada e expressada racional ou intuitivamente); ou uma agregação, horas extras, de qualquer um ou de todos esses itens.”. A partir disto, observa-se que o valor é a percepção do consumidor sobre o que ele recebe e o que ele fornece. O que ele recebe pode ser um benefício e o que ele fornece um custo ou sacrifício.

Assim, o *perceived value* é a avaliação do consumidor sobre as capacidades de um produto ou serviço para atender às suas necessidades e expectativas. De maneira a compreender o valor de um produto ou serviço e como os mesmos são identificados como oportunidade para o cliente, Woodall definiu uma lista de indicadores presentes na Tabela 7 (Conceptualising 'Value for the Customer', 2003), que segundo Zeithaml (1988), “é a avaliação geral do consumidor da utilidade de um produto com base em percepções do que é recebido e do que é dado”.

Tabela 7 – Benefícios e Sacríficos. Adaptado de Woodall (2003)

BENEFÍCIOS		SACRÍFICIOS
Atributos	Resultados	
Qualidade procurada	Benefícios funcionais	Preço
Qualidade do produto	Utilidade	Preço do mercado

BENEFÍCIOS		SACRÍFIÇOS
Atributos	Resultados	
Qualidade	Função de utilização	Custos monetários
Qualidade do serviço	Função estética	Finanças
Qualidade técnica	Benefícios operacionais	Custos
Qualidade funcional	Economia	Custos de utilização
Qualidade de desempenho	Benefícios logísticos	Custos percebidos
Desempenho do serviço	Benefícios do produto	Custos de pesquisa
Serviço	Benefícios estratégicos	Custos de aquisição
Suporte do serviço	Benefícios financeiros	Custos de oportunidade
Aspetos especiais do serviço	Resultado para o cliente	Custos de distribuição e instalação
Serviços adicionais	Benefícios sociais	Custos de reparação
Solução principal	Segurança	Custos de formação e de manutenção
Personalização	Conveniência	Custos não monetários
Confiança	Satisfação	Custos não financeiros
Características do produto	Apreciação por parte dos utilizadores	Custos de relação
Atributos do produto	Conhecimento, humor	Custos psicológicos
Características	Expressão própria	Tempo
Desempenho	Benefícios pessoais	Recursos humanos
	Associação com grupos sociais	Esforço
	Entusiasmo afetivo	

Tendo em conta a tabela anterior, foram utilizados os seus valores para elaborar uma segunda em específico para a análise de oportunidade. A Tabela 8 encontra-se dividida entre benefícios e sacrifícios para três áreas (produto, serviço e relação com o cliente). No enquadramento deste projeto, os principais clientes serão os colaboradores internos da ALERT e os utilizadores do produto.

Tabela 8 – *Perceived Value*

	Produto	Serviço	Relação
Benefícios	Garantias de qualidade e segurança;	Garantir a qualidade do serviço;	Redução do tempo e custos
	Melhorar o desempenho e a experiência dos colaboradores e utilizadores aquando a utilização dos produtos ALERT®;	Flexibilidade de adaptação aos pedidos dos clientes;	associados aos processos de formação;
	Melhorar os processos da formação em sala.	Flexibilidade de resposta;	Aumento da satisfação do cliente;
		Maior rapidez na procura do conteúdo;	Aumentar a confiança nos produtos
		Serviço prestado de confiança	ALERT®;
			Fortificação da imagem da ALERT;
			Supervisão;
			Assistência.
Sacríficos	Custos de tempo relacionados com a pesquisa, implementação e desenvolvimento;	Custos mão-de-obra e tempo.	Custos de formação dos colaboradores no uso do serviço.
	Custos mão-de-obra.		

Verificando a tabela acima, no que diz respeito ao Produto e o valor que representa para o cliente, o benefício principal é a sua finalidade uma vez que facilitará nas diversas atividades de formação em sala, como por exemplo, através da sua conta pessoal, o formando poderá registar a sua presença na formação além da requerida pelo formador. Desta forma é possível garantir um *double check* do número *real* de formandos presentes na ação de formação. A partir do trabalho efetuado na sala de aula, o formador recolhe informação muito diferente em relação aos seus formandos e que, obrigatoriamente, formaliza em registos estruturados: informação qualitativa e informação quantitativa. Todos os benefícios enunciados na Tabela 8 relativamente ao Produto, tem um impacto positivo na empresa, pois um dos pontos fundamentais e cruciais para o reforço e sustentação das equipas dentro de uma organização incide sobre as formações aos colaboradores. Sobre os sacrifícios, a empresa necessita de investir quer em tempo quer na mão-de-obra associada à pesquisa e desenvolvimento do produto.

Relativamente ao valor associado ao Serviço prestado pelo produto, devido à constante necessidade de alterar planos de formação e necessidade de fornecer às partes interessadas, é

importante que exista uma área para efetuar operações de gestão e visualização desses planos, facilitando e tornando mais prático todo o processo. A investigação associada ao produto tem por base detetar erros e compreender como os mesmos podem ser melhorados e/ou corrigidos de forma a desenvolver o melhor *software*. Com este serviço pretende-se beneficiar o cliente: através da sua flexibilidade de resposta e do seu serviço de confiança; redução de tempo na procura de conteúdo no decorrer da formação; flexibilidade de adaptação aos pedidos dos clientes antes, durante e depois da formação. Ainda assim, existe custo de tempo e mão-de-obra.

Finalmente, a Relação entre cliente e produto sai beneficiada uma vez que existe supervisão e assistência contínua nos produtos e serviços prestados, o que estabelece um relacionamento confiável. Outro benefício passa pela redução do tempo despendido nos processos de formação e conseqüentemente, redução dos custos associados. Além disso, a satisfação do cliente permite fortificar a imagem da ALERT para o mundo exterior. Contudo, os custos de formação dos colaboradores sobre o novo produto é um sacrifício, pois a ALERT tem como política formar todos os seus colaboradores aquando a disponibilização de novos produtos.

Em resumo, a proposta de valor apresentada permitirá reduzir tempo, aumentar a qualidade e interatividade entre as equipas de desenvolvimento da ALERT e nas ações de formação do cliente tendo em conta todos os benefícios associados em comparação com os sacrifícios.

3.1.1.5 Ideia

Face ao exposto nas fases anteriores do modelo *New Concept Development*, a ideia associada a este projeto consiste na disponibilização de um *software* exclusivo para a gestão de processos de formação de colaboradores e clientes da ALERT. Adicionalmente, é possível acompanhar, monitorar e responder com maior prontidão às necessidades das equipas de desenvolvimento e clientes da ALERT. Tudo isto permitirá aumentar a qualidade e gestão dos produtos ALERT® e, posteriormente, “melhorar a saúde e prolongar a vida, alcançar rentabilidade para benefício da sociedade e inspirar outros para a excelência” (Computing, 2019).

3.1.1.6 Conceito

Como conclusão do NCD, o conceito do projeto pode ser definido como o desenvolvimento de uma plataforma Web com a finalidade de providenciar diferentes tipos de operações para a gestão da formação em sala, como por exemplo, inscrições/matrículas em diferentes cursos de aprendizagem, horários dos professores, a assiduidade de cada aluno, entre outros. Neste sentido, a plataforma deve informatizar os principais processos do ciclo da formação, no contexto de sala de aula, sincronizar as informações com o produto ALERT® eLEARNING e garantir uma maior eficiência na sua gestão e produtividade, reduzindo os custos associados aos mesmos.

3.1.2 Identificação e Análise Funcional

A próxima etapa na *Value Analysis* é a etapa de Identificação e Análise Funcional, onde será efetuada a análise do produto, com base na identificação dos requisitos e funções mais importantes que este oferece ao cliente (Rich & Holweg, 2000). Uma função pode ser definida como a utilidade e o valor de estima (*esteem value*) que esta transmite ao cliente (Rich & Holweg, 2000). Essas funções fazem o produto funcionar eficazmente ou contribuem para a venda do produto (Rich & Holweg, 2000).

Esta etapa encontra-se dividida em duas partes, isto é, a Identificação Funcional e a Análise Funcional. Na primeira, são identificados e definidos os requisitos mais importantes do produto, enquanto que na segunda é efetuada uma análise a esses requisitos, bem como uma comparação entre as diversas funções com o intuito de as avaliar.

3.1.2.1 Identificação Funcional

Nesta etapa, pretende-se dar resposta à questão “Quais são as funções que este produto deve responder?” (Rich & Holweg, 2000). Para tal é necessário efetuar algumas atividades que primeiramente podem parecer demoradas, mas cruciais no levantamento de funções, pois o trabalho com elas geralmente fornece informações muito valiosas sobre o valor e a funcionalidade do produto (Rich & Holweg, 2000). Como referido, esta etapa do processo de VA é morosa, no entanto, um pouco agradável porque ajuda a compreender que tipo de funções o cliente exige num produto e conseqüentemente, impedir uma análise precipitada que pode originar no produto errado (Rich & Holweg, 2000).

Para determinar as principais funcionalidades do produto, recorre-se à utilização do *Quality Function Deployment* (QFD) (Quality Function Deployment, 2007). O QFD é uma das ferramentas da qualidade que tem como objetivo principal permitir incorporar as necessidades do cliente no desenvolvimento deste projeto (Quality Function Deployment, 2007). Por assim dizer, o QFD é uma ferramenta que permite “ouvir” a voz do cliente e ordená-la de modo a facilitar a análise das suas necessidades que são transformadas em requisitos para a melhoria do produto na forma de especificações técnicas do mesmo.

O QFD consegue atingir estes resultados (valor, reputação e aumento do negócio para o cliente) através de uma análise pormenorizada aos requisitos propostos pelo cliente (Quality Function Deployment, 2007). Cada requisito é segmentado e são identificadas formas que permitam atingir esse segmento (Quality Function Deployment, 2007).

Na prática, o processo QFD é realizado através de matrizes que desdobram as necessidades do cliente e os requisitos técnicos relacionados a ela. Assim, para iniciar o QFD, é elaborada uma *Customer Chain* (Quality Function Deployment, 2007), onde se identifica a cadeia de clientes do produto. A Figura 18 apresenta a *Customer Chain* para o produto em desenvolvimento.



Figura 18 – *Customer Chain*

Observando a figura anterior, é perceptível que o cliente principal do produto são os colaboradores ALERT embora indiretamente existam outros clientes, como por exemplo, a própria empresa ALERT. Adicionalmente, os próprios clientes ALERT também são considerados clientes do produto em desenvolvimento. Contudo, não se pode esquecer os profissionais de saúde que também são clientes finais do produto de formação.

Tendo em conta que no parágrafo anterior foram identificados os nossos clientes, é necessário proceder ao levantamento dos requisitos de qualidade mais importantes impostos por eles. Assim, com base nas características gerais dos requisitos definidos na secção 4.2 deste documento, foram identificados os seguintes requisitos de qualidade do cliente (“*Whats*”):

- Bom desempenho;
- Segurança de acesso;
- Troca segura de dados;
- Alta usabilidade da interface gráfica;
- Qualidade e integridade dos planos de formação;
- Redução de tempo.

Para cada um destes requisitos de qualidade do cliente foi necessário avaliar o seu peso/grau de importância. Assim, foram realizados questionários para colaboradores da empresa em relação a esse mesmo grau de importância, onde foi pedido para atribuírem uma pontuação a cada requisito proposto, sendo que a soma total da pontuação dada por cada elemento teria de ser igual a 100. Com todas as respostas recolhidas, procedeu-se ao cálculo da média atribuída a cada critério. Com valores arredondados, seguem os resultados:

- Bom desempenho - 20;
- Segurança de acesso – 15;
- Troca de dados segura - 15;
- Alta usabilidade da interface gráfica - 10;
- Qualidade e integridade dos planos de formação – 40.

Além dos requisitos de qualidade do cliente, também foi necessário definir as características técnicas do produto (“Hows”):

- Autenticação;
- Autorização;
- Gerir Learning Plan;
- Gerir inscrições;
- Tempo de transação dos dados;
- Interface gráfica responsiva e intuitiva.

3.1.2.2 Análise Funcional

Seguidamente, será elaborada uma “Casa de Qualidade” com base nos requisitos identificados na secção 3.1.2.1. Referir que esta “Casa de Qualidade” apresenta a comparação entre os requisitos de qualidade e os requisitos funcionais e avalia a relação (positiva ou negativa) entre os mesmos (Quality Function Deployment, 2007). São também considerados ainda os graus de importância associados aos requisitos de qualidade do cliente aquando a realização da “Casa de Qualidade” (Quality Function Deployment, 2007). Por último, são identificados os valores/objetivos a atingir e um grau de dificuldade para cada um dos requisitos funcionais (Quality Function Deployment, 2007). A Figura 19 apresenta a “Casa de Qualidade” no contexto deste projeto.

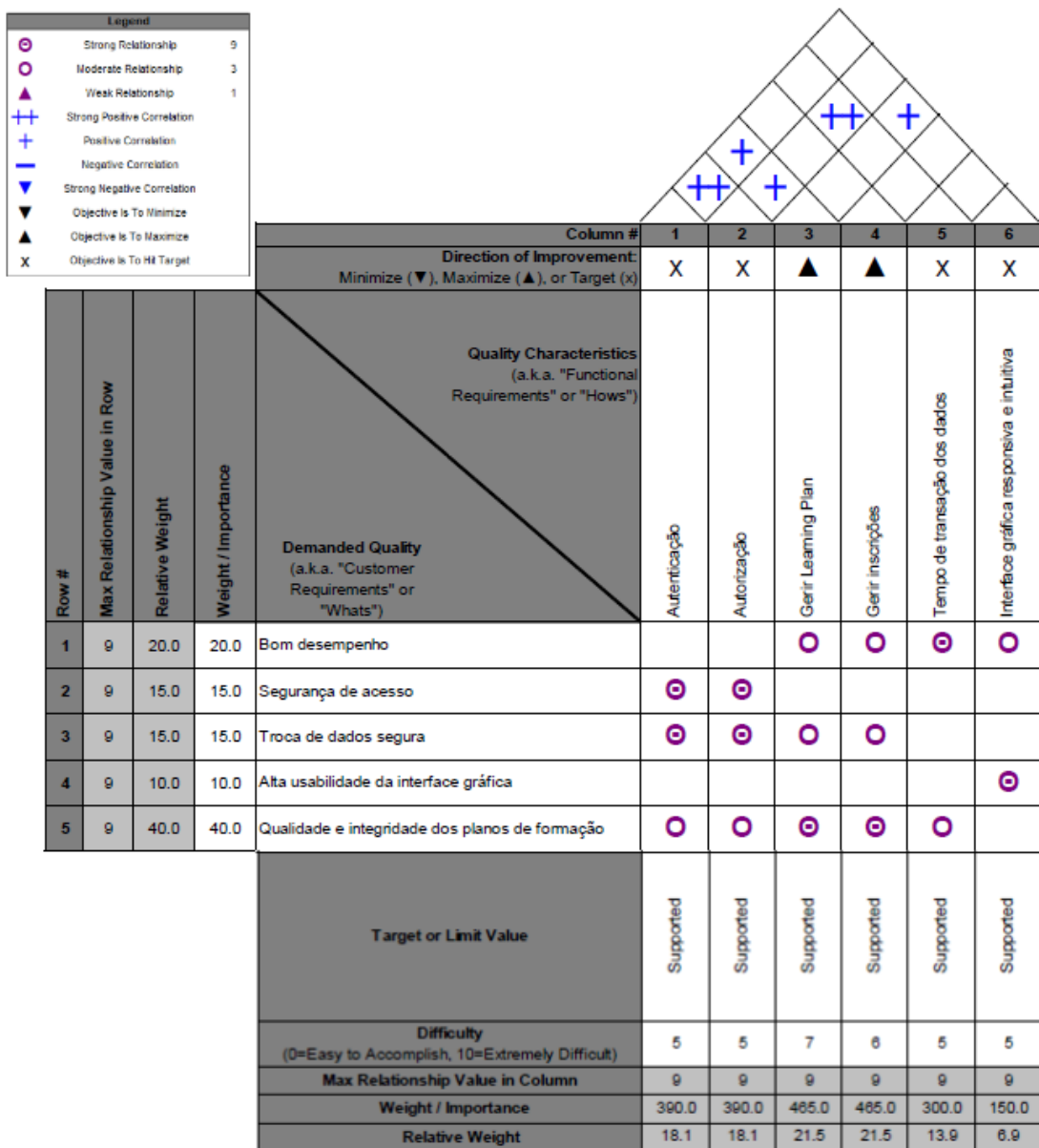


Figura 19 – Casa de Qualidade

Com base nos resultados da aplicação técnica QFD e da Cada de Qualidade apresentada na figura anterior, foi possível concluir que:

- De todos os requisitos de qualidade do cliente, a “qualidade e integridade dos planos de formação” é o que apresenta maior importância, sendo que a “troca de dados segura”

surge em seguida. Com isto, o cliente pretende que o produto seja desenvolvido garantindo a qualidade e integridade dos dados dos planos de formação (*learning plan*), isto é, que chegue ao destino sem perda de informação e apresente um bom tempo de transação dos dados. Contudo, não se deve menosprezar os restantes requisitos uma vez que também são essências no desenvolvimento do produto;

- No que toca às comparações entre requisitos de qualidade e requisitos funcionais do produto, foram definidos três tipos de graus de força com base na relação entre eles: relação forte, relação moderada e relação fraca. Com um grau de relação forte, o grau mais importante, destacam-se as seguintes relações:
 - Bom desempenho e tempo de transação de dados;
 - Segurança de acesso e autenticação e autorização;
 - Troca de dados segura e autenticação e autorização;
 - Alta usabilidade da interface gráfica e interface gráfica responsiva e intuitiva;
 - Qualidade e integridade dos planos de formação e *gerir learning plan* e inscrições.
- Em comparação com as relações destacadas no ponto anterior, existem outras, no entanto, o seu grau de importância é mais baixo;
- Relativamente aos requisitos funcionais, é possível destacar as seguintes correlações:
 - Correlação fortemente positiva entre a autenticação e a autorização, pois a complementaridade entre ambos aumenta a segurança do sistema;
 - Correlação positiva entre a autorização e *gerir learning plan* de forma a garantir que nem todos os utilizadores devem ter acesso a essa administração.
- Acima de cada requisito é marcado se a característica deve ser mais baixa (seta para baixo), mais alta (seta para cima) ou nenhuma (sem seta). No contexto deste projeto, as características *gerir learning plan* e *gerir inscrições* devem ser altas pelo que ambas apresentam seta para cima. Com base nelas, foi possível determinar a seguinte correlação:
 - Correlação fortemente positiva entre *gerir learning plan* e *gerir inscrições*. Isto acontece não só devido à direção das suas setas, mas também porque *gerir learning plan* afeta *gerir inscrições*.
- Não são identificadas correlações negativas entre os diferentes requisitos funcionais;

- De forma a garantir a melhor qualidade do produto, todos os requisitos funcionais apresentados devem ser implementados. Tendo em consideração os níveis de dificuldade associados, destaca-se a complexidade de execução para gerir *learning plans*.

Por último, com base na análise dos requisitos do cliente e requisitos funcionais, seguidamente, procede-se à classificação das funções por comparação em pares para julgar qual delas é preferida, ou possui um valor e/ou pontuação final mais elevada, sendo que não se deve permitir empates (Rich & Holweg, 2000). Para isso, foi necessário recorrer ao método “Pairwise Comparison” que permite utilizar uma tabela onde cada função é comparada entre si quanto à sua importância (Rich & Holweg, 2000). Assim, foram considerados os seguintes valores (Rich & Holweg, 2000):

- 1 ponto – diferença de importância menor, ou seja, o primeiro termo é um pouco mais importante/melhor que o segundo;
- 2 pontos – diferença de importância média, ou seja, o primeiro termo é significativamente mais importante/melhor que o segundo;
- 3 pontos – diferença de importância maior, ou seja, o primeiro termo é muito mais importante que o segundo.

A Figura 20 apresenta a tabela com as comparações entre as três funções.

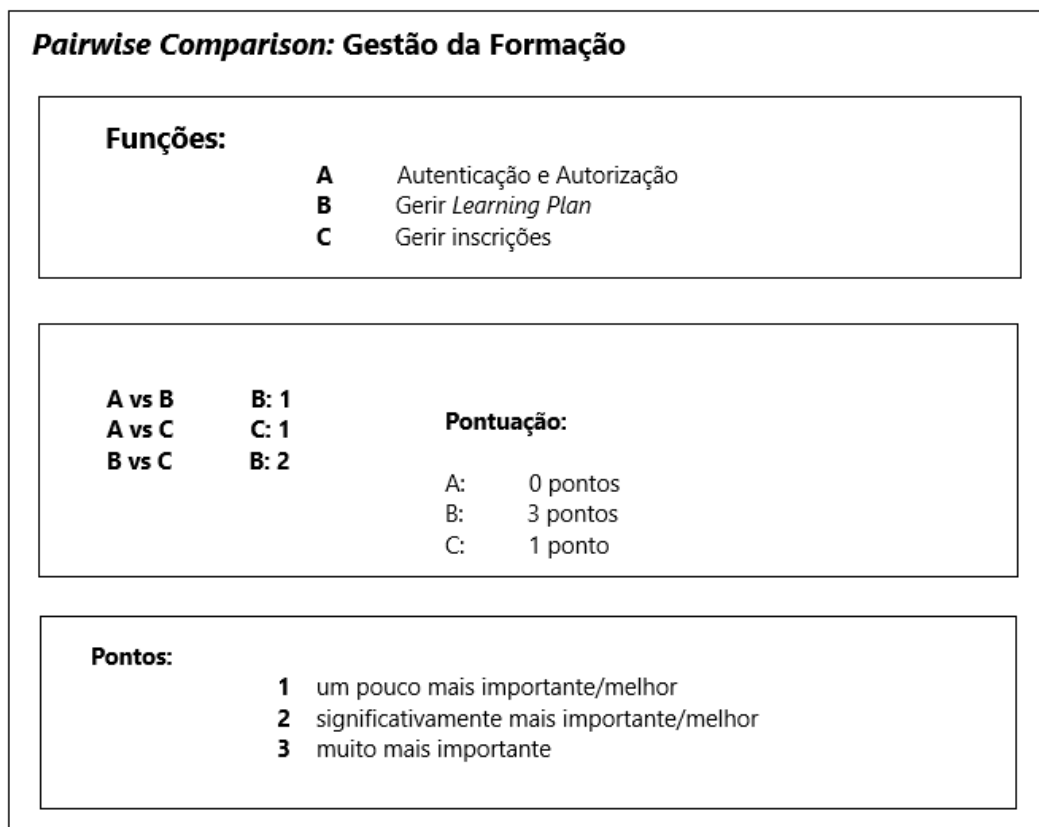


Figura 20 – *Pairwise Comparison*: Gestão da Formação

Tendo em conta os resultados obtidos na figura anterior é possível objetar que a função que apresenta uma maior importância é a função Gerir *Learning Plan* uma vez que a mesma obteve uma pontuação de três pontos. Em segundo lugar, encontra-se a função Gerir Inscrições com uma pontuação de um ponto. Em último lugar, a função Autenticação e Autorização obteve zero pontos indicando que a sua importância é muito menos importante que as outras duas, no entanto, merece a mesma atenção no seu desenvolvimento.

3.1.3 Criação de Alternativas

Nesta secção serão identificadas as ferramentas existentes que auxiliam nos processos de gestão da formação. Sendo o processo de gestão da formação complexo, foram identificadas as seguintes alternativas (previamente estudadas nas secções 2.3, 2.5 e 2.6 do presente documento):

- Talent;
- Intuo;
- PowerSchool

- Adobe Captivate Prime;
- FORINSIA.

3.1.4 Análise e Avaliação

No decorrer desta fase (Análise e Avaliação), é utilizado o método *Analytic Hierarchy Process* (AHP), como uma ferramenta de auxílio multicritério à decisão do melhor objetivo (Nicola, 2019). A tomada de decisão é um processo cognitivo, que tem como objetivo encontrar a alternativa mais eficiente (tendo em consideração uma série de critérios) de entre a gama de possibilidades descritas na secção 3.1.3.

Assim, foram identificados os seguintes critérios para avaliação das ferramentas:

- Suporte – este critério centra as suas forças no apoio ao cliente no momento da configuração, bem como durante uma execução normal para resolver problemas relacionados com o funcionamento e a manutenção;
- Migração de dados – devido à quantidade de dados armazenados nas alternativas expostas pode existir a necessidade de querer mover esses dados para outra alternativa (plataforma) e, por esta mesma razão, uma plataforma eficaz deve suportar uma ampla gama de formatos. Além disso, deve tornar o processo de migração de dados o mais simples e direto;
- Customização e personalização – recursos personalizados, ou seja, deve ser possível escolher as funcionalidades que melhor se adequam à gestão da formação de uma organização, por exemplo, relatórios pré-criados que cobrem uma variedade de métricas. Contudo, o segredo para o sucesso de uma formação não presencial é a personalização, pois os alunos necessitam de ferramentas e recursos de formação não presencial que correspondam às suas metas (por exemplo, os alunos podem escolher atividades e avaliações de formação com base naquilo que eles necessitam de aprender permitindo-lhes melhorar continuamente sem ter que acompanhar outros interessados, ou ao contrário, os formadores escolhem recursos de formação não presencial para um número de alunos específico e por base numa meta conjunta).
- Integração e utilização – os utilizadores de uma plataforma podem querer comunicar com outras de nível empresarial. Nesses cenários, recursos de integração podem ampliar o alcance do público-alvo e a sua utilização deve ser facilitada.

De forma a iniciar a fase de análise e avaliação, a Figura 21 mostra a decomposição do problema/objetivo numa hierarquia de critérios, que posteriormente serão analisados e comparados entre si, de modo independente (Saaty, 2008). O problema a solucionar é a escolha de uma das ferramentas.

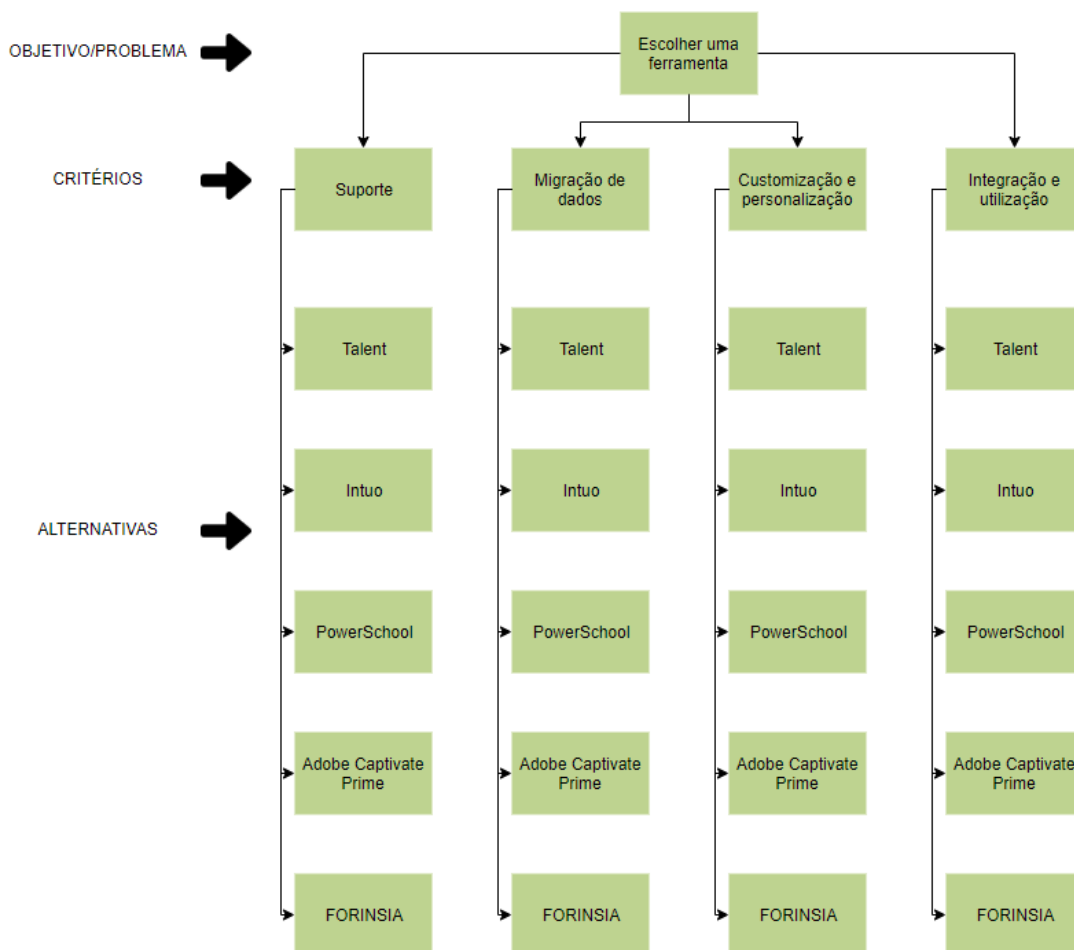


Figura 21 - Árvore de divisão hierárquica

Como verificado na figura anterior, existem quatro critérios e cada um possui seis alternativas. Estas ferramentas (identificadas como alternativas) foram definidas na secção 3.1.3 do *Value Analysis*.

Assim, de modo a passar à etapa seguinte, isto é, à comparação de critérios e alternativas, necessitamos de uma escala de números que indique “quantas vezes um elemento pode ser mais importante ou dominante que outro em relação ao critério ou propriedade à qual são comparados” (Saaty, 2008). Para efetuar a comparação entre elementos, é mais comum a utilização da escala de importância proposta por Saaty (Saaty, 2008), atribuindo valores que variam entre 1 e 9, conforme apresentado na Tabela 9.

Tabela 9 – Escala fundamental – Níveis de importância de comparações (Nicola, 2019)

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2, 4, 6, 8	Valores intermediários	Quando se procura uma condição de compromisso entre duas definições

Tendo em conta os valores da tabela anterior, a Tabela 10 apresenta as prioridades atribuídas a cada um dos critérios quando comparados entre cada um.

Tabela 10 – Comparação de prioridades dos critérios (matriz de prioridades)

	Suporte	Migração de dados	Customização e personalização	Integração e utilização
Suporte	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$
Migração de dados	3	1	3	3
Customização e personalização	2	$\frac{1}{3}$	1	2
Integração e utilização	2	$\frac{1}{3}$	$\frac{1}{2}$	1

Como é possível verificar, o critério de migração de dados, quando comparado com os restantes critérios, é o que apresenta maior nível de importância e/ou maior prioridade (por exemplo, o critério migração de dados apresenta um grau de importância ligeiramente superior quando comparado com o critério de customização e personalização). Realisticamente, este critério é o mais importante uma vez que a ferramenta deve permitir a migração de grandes quantidades de dados.

Em terceiro lugar, aparece o critério de integração e utilização. De facto, como explicado em parágrafos anteriores, quando comparado com o critério de customização e personalização é ligeiramente desfavorecido uma vez que as características que apresenta não chegam para as

que são disponibilizadas pelo critério de customização e personalização. Por último, o critério suporte que apesar da sua prioridade, não deixa de ser menos importante para escolher uma ferramenta.

De modo a dar continuidade ao processo AHP, a Tabela 11 apresenta a soma de todas as colunas da tabela anterior. Esta fase é extremamente importante uma vez que influencia os cálculos de fases posteriores.

Tabela 11 – Somatório das colunas de cada critério

	Suporte	Migração de dados	Customização e personalização	Integração e utilização
Suporte	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$
Migração de dados	3	1	3	3
Customização e personalização	2	$\frac{1}{3}$	1	2
Integração e utilização	2	$\frac{1}{3}$	$\frac{1}{2}$	1
Soma	8	2	5	$\frac{13}{2}$

A Tabela 12 apresenta a normalização dos valores da tabela anterior, isto é, a igualdade dos valores de cada um dos critérios a uma mesma unidade. Para realizar estes cálculos, foi efetuada a divisão do valor atribuído a cada critério pela soma total da sua respetiva coluna.

Tabela 12 – Comparação de prioridades dos critérios normalizados

	Suporte	Migração de dados	Customização e personalização	Integração e utilização
Suporte	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{10}$	$\frac{1}{13}$
Migração de dados	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{6}{13}$
Customização e personalização	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{4}{13}$
Integração e utilização	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{10}$	$\frac{2}{13}$

Em seguida, foi obtido o vetor de prioridades que tem como objetivo único identificar a ordem de importância de cada um dos critérios (Nicola, 2019). A Tabela 13 apresenta a média aritmética dos valores de cada uma das linhas obtidas na tabela anterior.

Tabela 13 – Prioridade dos critérios

	Suporte	Migração de dados	Customização e personalização	Integração e utilização	Prioridade (vetor próprio)
Suporte	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{10}$	$\frac{1}{13}$	0.12
Migração de dados	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{6}{13}$	0.48
Customização e personalização	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{4}{13}$	0.23
Integração e utilização	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{10}$	$\frac{2}{13}$	0.17

Com base na tabela anterior é possível retirar o seguinte vetor de prioridades (ou vetor próprio):

$$\begin{bmatrix} 0,12 \\ 0,48 \\ 0,23 \\ 0,17 \end{bmatrix}$$

Assim, o critério de migração de dados é aquele que apresenta maior prioridade. Em segundo lugar, encontram-se o critério de customização e personalização. A terceira posição é ocupada pelo critério de integração e utilização. Por último, e não menos importante, o critério de suporte.

Após encontradas as prioridades e de modo a continuar o processo AHP, a próxima etapa passa por calcular a Razão de Consistência (RC). Esta mede a consistência das prioridades em relação a grandes amostras de juízos completamente aleatórios (Nicola, 2019). O resultado do RC deve ser inferior a 0,1 de modo a que as prioridades sejam consideradas consistentes e confiáveis. O RC pode ser calculado através da seguinte fórmula:

$$RC = \frac{IC}{IR}$$

, onde IC é o Índice de Consistência e IR o Índice Aleatório. A Tabela 14 foi definida pelo Laboratório Nacional de Oak Ridge, nos EUA, e apresenta os valores de IR em função do número de critérios (Nicola, 2019).

Tabela 14 - Valores IR definidos pelo Laboratório Nacional de Oak Ridge (Nicola, 2019)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Contudo, de forma a obter o RC também é necessário calcular o IC (Nicola, 2019). De modo a realizar esse cálculo, utiliza-se a seguinte fórmula:

$$IC = \frac{\lambda_{max} - n}{n - 1}$$

, onde λ_{max} é o valor próprio e n o número de critérios avaliados (Nicola, 2019).

Assim, λ_{max} é calculado com base na matriz de prioridades inicial e no vetor próprio obtido em cálculos anteriores:

Matriz de prioridades \times Vetor próprio $\cong \lambda_{max} \times$ Vetor próprio

$$\Leftrightarrow \begin{bmatrix} 1 & 0,33 & 0,5 & 0,5 \\ 3 & 1 & 3 & 3 \\ 2 & 0,33 & 1 & 2 \\ 2 & 0,33 & 0,5 & 1 \end{bmatrix} \times \begin{bmatrix} 0,12 \\ 0,48 \\ 0,23 \\ 0,17 \end{bmatrix} \cong \lambda_{max} \times \begin{bmatrix} 0,12 \\ 0,48 \\ 0,23 \\ 0,17 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 0,46 \\ 2,04 \\ 0,96 \\ 0,67 \end{bmatrix} \cong \lambda_{max} \times \begin{bmatrix} 0,12 \\ 0,48 \\ 0,23 \\ 0,17 \end{bmatrix}$$

$$\Leftrightarrow \lambda_{max} \cong 4,04$$

Tendo encontrado o valor de λ_{max} , é possível calcular o IC:

$$IC = \frac{\lambda_{max} - n}{n - 1} = \frac{4,04 - 4}{4 - 1} = \frac{0,04}{3} = 0,01$$

Em último lugar, procede-se ao cálculo do valor de RC, onde o valor de IR, segundo a Tabela 14 é de 0,90 uma vez que o número de critérios estabelecidos são quatro (4):

$$RC = \frac{IC}{IR} = \frac{0,01}{0,90} = 0,01$$

Como referido anteriormente, uma vez que o valor de RC $< 0,1$, isto é, $0,01 < 0,1$, indica que os valores de prioridades são consistentes e confiáveis.

Assim, uma vez que foi verificada a consistência e confiabilidade dos critérios, a próxima fase demanda o cálculo da importância relativa de cada uma das alternativas que compõem a estrutura hierárquica do problema em questão para cada um dos critérios (Nicola, 2019). Para que isto seja possível, deve-se construir a matriz de comparação das alternativas de cada critério, somar cada uma das colunas da matriz, normalizar a mesma e, por último, obter o vetor próprio associado a cada critério (estes cálculos já foram realizados para a matriz de prioridades dos critérios observados desde a Tabela 10 à Tabela 13).

- Critério Suporte:

Tabela 15 – Matriz de prioridades das alternativas para o critério suporte

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
Intuo	2	1	2	$\frac{1}{2}$	$\frac{1}{2}$
PowerSchool	2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{4}$
Adobe Captivate Prime	3	2	2	1	$\frac{1}{2}$
FORINSIA	4	$\frac{12}{3}$	4	2	1

Tabela 16 – Somatório de cada coluna das alternativas do critério suporte

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
Intuo	2	1	2	$\frac{1}{2}$	$\frac{1}{2}$
PowerSchool	2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{4}$
Adobe Captivate Prime	3	2	2	1	$\frac{1}{2}$
FORINSIA	4	2	4	2	1
Soma	12	6	$\frac{19}{2}$	$\frac{13}{3}$	$\frac{5}{2}$

Tabela 17 – Valores normalizados das alternativas do critério suporte

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{19}$	$\frac{1}{13}$	$\frac{1}{10}$
Intuo	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{4}{19}$	$\frac{3}{26}$	$\frac{1}{5}$
PowerSchool	$\frac{1}{6}$	$\frac{1}{12}$	$\frac{2}{19}$	$\frac{3}{26}$	$\frac{1}{10}$
Adobe Captivate Prime	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{4}{19}$	$\frac{3}{13}$	$\frac{1}{5}$
FORINSIA	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{8}{19}$	$\frac{6}{13}$	$\frac{2}{5}$

Tabela 18 – Prioridades das alternativas do critério suporte

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA	Prioridade (vetor próprio)
Talent	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{19}$	$\frac{1}{13}$	$\frac{1}{10}$	0,08
Intuo	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{4}{19}$	$\frac{3}{26}$	$\frac{1}{5}$	0,17
PowerSchool	$\frac{1}{6}$	$\frac{1}{12}$	$\frac{2}{19}$	$\frac{3}{26}$	$\frac{1}{10}$	0,12
Adobe Captivate Prime	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{4}{19}$	$\frac{3}{13}$	$\frac{1}{5}$	0,24
FORINSIA	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{8}{19}$	$\frac{6}{13}$	$\frac{2}{5}$	0,39

Por fim, o vetor próprio obtido para o critério suporte é o seguinte:

$$\begin{bmatrix} 0,08 \\ 0,17 \\ 0,12 \\ 0,24 \\ 0,39 \end{bmatrix}$$

- Critério Migração de Dados:

Tabela 19 – Matriz prioridades das alternativas para o critério migração de dados

	Talent	Intuo	PowerSchool	Adobe Captive Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{7}$	$\frac{1}{3}$	$\frac{1}{8}$
Intuo	2	1	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{7}$
PowerSchool	7	6	1	3	$\frac{1}{2}$
Adobe Captive Prime	3	2	$\frac{1}{3}$	1	$\frac{1}{3}$
FORINSIA	8	7	2	3	1

Tabela 20 – Somatório de cada coluna das alternativas do critério migração de dados

	Talent	Intuo	PowerSchool	Adobe Captive Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{7}$	$\frac{1}{3}$	$\frac{1}{8}$
Intuo	2	1	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{7}$
PowerSchool	7	6	1	3	$\frac{1}{2}$
Adobe Captive Prime	3	2	$\frac{1}{3}$	1	$\frac{1}{3}$
FORINSIA	8	7	2	3	1
Soma	21	$\frac{33}{2}$	$\frac{51}{14}$	$\frac{47}{6}$	$\frac{353}{168}$

Tabela 21 – Valores normalizados das alternativas do critério migração de dados

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	$\frac{1}{21}$	$\frac{1}{33}$	$\frac{2}{21}$	$\frac{6}{329}$	$\frac{79}{1412}$
Intuo	$\frac{2}{21}$	$\frac{2}{33}$	$\frac{7}{153}$	$\frac{3}{47}$	$\frac{158}{2471}$
PowerSchool	$\frac{1}{3}$	$\frac{4}{11}$	$\frac{14}{51}$	$\frac{18}{47}$	$\frac{79}{353}$
Adobe Captivate Prime	$\frac{1}{7}$	$\frac{4}{33}$	$\frac{14}{153}$	$\frac{6}{47}$	$\frac{158}{1059}$
FORINSIA	$\frac{8}{21}$	$\frac{14}{33}$	$\frac{28}{51}$	$\frac{18}{47}$	$\frac{168}{353}$

Tabela 22 – Prioridades das alternativas do critério migração de dados

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA	Prioridade (vetor próprio)
Talent	$\frac{1}{21}$	$\frac{1}{33}$	$\frac{2}{21}$	$\frac{6}{329}$	$\frac{79}{1412}$	0,05
Intuo	$\frac{2}{21}$	$\frac{2}{33}$	$\frac{7}{153}$	$\frac{3}{47}$	$\frac{158}{2471}$	0,07
PowerSchool	$\frac{1}{3}$	$\frac{4}{11}$	$\frac{14}{51}$	$\frac{18}{47}$	$\frac{79}{353}$	0,31
Adobe Captivate Prime	$\frac{1}{7}$	$\frac{4}{33}$	$\frac{14}{153}$	$\frac{6}{47}$	$\frac{158}{1059}$	0,13
FORINSIA	$\frac{8}{21}$	$\frac{14}{33}$	$\frac{28}{51}$	$\frac{18}{47}$	$\frac{168}{353}$	0,44

Por fim, o vetor próprio obtido para o critério migração de dados é o seguinte:

$$\begin{bmatrix} 0,05 \\ 0,07 \\ 0,31 \\ 0,13 \\ 0,44 \end{bmatrix}$$

- Critério Customização e Personalização

Tabela 23 – Matriz de prioridades das alternativas para o critério customização e personalização

	Talent	Intuo	PowerSchool	Adobe Captive Prime	FORINSIA
Talent	1	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{7}$	$\frac{1}{4}$
Intuo	6	1	5	$\frac{1}{4}$	4
PowerSchool	4	$\frac{1}{5}$	1	$\frac{1}{6}$	$\frac{1}{4}$
Adobe Captive Prime	7	4	6	1	5
FORINSIA	4	$\frac{1}{4}$	4	$\frac{1}{5}$	1

Tabela 24 – Somatório de cada coluna das alternativas do critério customização e personalização

	Talent	Intuo	PowerSchool	Adobe Captive Prime	FORINSIA
Talent	1	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{7}$	$\frac{1}{4}$
Intuo	6	1	5	$\frac{1}{4}$	4
PowerSchool	4	$\frac{1}{5}$	1	$\frac{1}{6}$	$\frac{1}{4}$
Adobe Captive Prime	7	4	6	1	5
FORINSIA	4	$\frac{1}{4}$	4	$\frac{1}{5}$	1
Soma	22	$\frac{65}{12}$	$\frac{65}{4}$	$\frac{223}{140}$	$\frac{21}{2}$

Tabela 25 – Valores normalizados das alternativas do critério customização e personalização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	$\frac{1}{22}$	$\frac{2}{65}$	$\frac{1}{65}$	$\frac{140}{1561}$	$\frac{1}{42}$
Intuo	$\frac{3}{11}$	$\frac{12}{65}$	$\frac{4}{13}$	$\frac{35}{223}$	$\frac{8}{21}$
PowerSchool	$\frac{2}{11}$	$\frac{12}{325}$	$\frac{4}{65}$	$\frac{70}{669}$	$\frac{1}{42}$
Adobe Captivate Prime	$\frac{1}{22}$	$\frac{48}{65}$	$\frac{24}{65}$	$\frac{140}{223}$	$\frac{10}{21}$
FORINSIA	$\frac{2}{11}$	$\frac{3}{65}$	$\frac{16}{65}$	$\frac{28}{223}$	$\frac{2}{21}$

Tabela 26 – Prioridades das alternativas do critério customização e personalização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA	Prioridade (vetor próprio)
Talent	$\frac{1}{22}$	$\frac{2}{65}$	$\frac{1}{65}$	$\frac{140}{1561}$	$\frac{1}{42}$	0,04
Intuo	$\frac{3}{11}$	$\frac{12}{65}$	$\frac{4}{13}$	$\frac{35}{223}$	$\frac{8}{21}$	0,26
PowerSchool	$\frac{2}{11}$	$\frac{12}{325}$	$\frac{4}{65}$	$\frac{70}{669}$	$\frac{1}{42}$	0,08
Adobe Captivate Prime	$\frac{1}{22}$	$\frac{48}{65}$	$\frac{24}{65}$	$\frac{140}{223}$	$\frac{10}{21}$	0,49
FORINSIA	$\frac{2}{11}$	$\frac{3}{65}$	$\frac{16}{65}$	$\frac{28}{223}$	$\frac{2}{21}$	0,13

Por fim, o vetor próprio obtido para o critério migração de dados é o seguinte:

$$\begin{bmatrix} 0,04 \\ 0,26 \\ 0,08 \\ 0,49 \\ 0,13 \end{bmatrix}$$

- Critério Integração e Utilização:

Tabela 27 – Matriz de prioridades das alternativas para o critério integração e utilização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{5}$
Intuo	2	1	2	$\frac{1}{2}$	$\frac{1}{3}$
PowerSchool	2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{5}$
Adobe Captivate Prime	4	2	2	1	$\frac{1}{3}$
FORINSIA	5	3	5	3	1

Tabela 28 – Somatório de cada coluna das alternativas do critério integração e utilização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{5}$
Intuo	2	1	2	$\frac{1}{2}$	$\frac{1}{3}$
PowerSchool	2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{5}$
Adobe Captivate Prime	4	2	2	1	$\frac{1}{3}$
FORINSIA	5	3	5	3	1
Soma	14	7	$\frac{21}{2}$	$\frac{21}{4}$	$\frac{31}{15}$

Tabela 29 – Valores normalizados das alternativas do critério integração e utilização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA
Talent	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{3}{31}$
Intuo	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{4}{21}$	$\frac{2}{21}$	$\frac{5}{31}$
PowerSchool	$\frac{1}{7}$	$\frac{1}{14}$	$\frac{2}{21}$	$\frac{2}{21}$	$\frac{3}{31}$
Adobe Captivate Prime	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{5}{31}$
FORINSIA	$\frac{5}{14}$	$\frac{3}{7}$	$\frac{10}{21}$	$\frac{12}{21}$	$\frac{15}{31}$

Tabela 30 – Prioridades das alternativas do critério integração e utilização

	Talent	Intuo	PowerSchool	Adobe Captivate Prime	FORINSIA	Prioridade (vetor próprio)
Talent	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{4}{21}$	$\frac{2}{21}$	$\frac{5}{31}$	0,07
Intuo	$\frac{1}{7}$	$\frac{1}{14}$	$\frac{2}{21}$	$\frac{2}{21}$	$\frac{3}{31}$	0,15
PowerSchool	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{5}{31}$	0,10
Adobe Captivate Prime	$\frac{5}{14}$	$\frac{3}{7}$	$\frac{10}{21}$	$\frac{12}{21}$	$\frac{15}{31}$	0,22
FORINSIA	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{4}{21}$	$\frac{2}{21}$	$\frac{5}{31}$	0,46

Por fim, o vetor próprio obtido para o critério migração de dados é o seguinte:

$$\begin{bmatrix} 0,07 \\ 0,15 \\ 0,10 \\ 0,22 \\ 0,36 \end{bmatrix}$$

Por fim, a última etapa do processo AHP passa pela elaboração de uma matriz comparativa paritária para cada critério, considerando cada uma das alternativas selecionadas (Nicola, 2019).

A Figura 22 a árvore de divisão hierárquica com base nos valores obtidos nos cálculos anteriores, ou seja, valores dos vetores próprios associados (dos critérios e das alternativas).

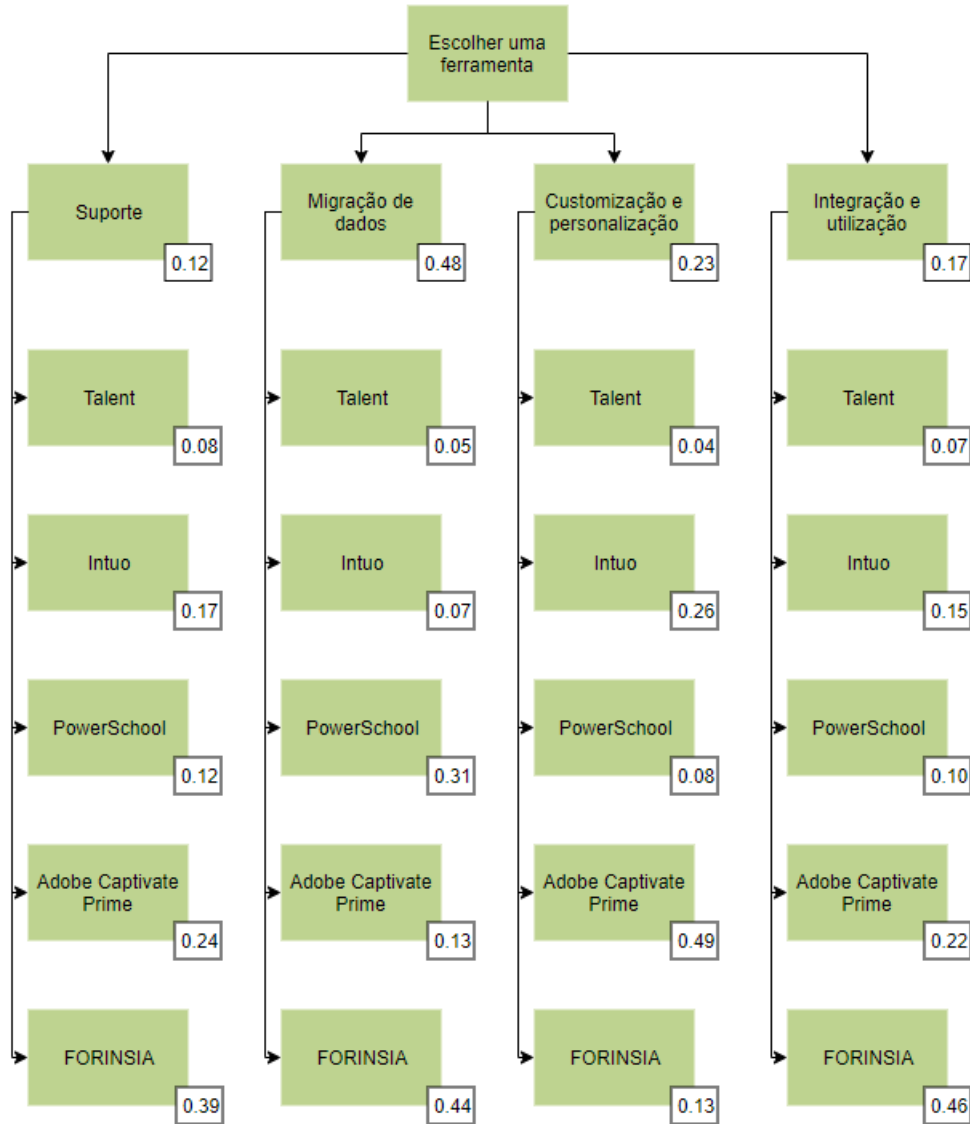


Figura 22 – Árvore de divisão hierárquica com valores associados

Em seguida, calcula-se a prioridade composta para as alternativas, multiplicando os valores obtidos anteriormente e os das prioridades relativas, obtidos no início do método (Nicola, 2019), ou seja:

$$\text{Matriz de Prioridades} \times \text{Peso dos critérios}$$

$$= \begin{bmatrix} 0,08 & 0,05 & 0,04 & 0,07 \\ 0,17 & 0,07 & 0,26 & 0,15 \\ 0,12 & 0,31 & 0,08 & 0,10 \\ 0,24 & 0,13 & 0,49 & 0,22 \\ 0,39 & 0,44 & 0,13 & 0,46 \end{bmatrix} \times \begin{bmatrix} 0,12 \\ 0,48 \\ 0,23 \\ 0,17 \end{bmatrix} = \begin{bmatrix} 0,05 \\ 0,14 \\ 0,20 \\ 0,24 \\ 0,37 \end{bmatrix}$$

Em síntese, através do vetor de prioridades obtido anteriormente, é possível identificar a ferramenta FORINSIA como a melhor escolha no auxílio da gestão da formação. Esta ferramenta obteve o maior valor (0,37), tendo em consideração os critérios definidos anteriormente e das suas respetivas importâncias.

3.1.5 Implementação

A última fase do processo de *Value Analysis* consiste na implementação do produto. Deve ser prestada maior atenção às funcionalidades disponibilizadas pela alternativa “FORINSIA” uma vez que aparece como a mais indicada para a recolha dos requisitos a desenvolver no sistema.

4 Desenvolvimento da Solução

O propósito deste género de secção do presente documento é a complementação do próprio sistema, permitindo a qualquer um dos seus grupos de utilizadores uma compreensão da sua utilização. Assim, são identificados casos de uso e atributos de qualidade, o modelo de domínio, o modelo de dados e exemplos da interface gráfica da plataforma desenvolvida.

4.1 Grupos de Utilizadores e suas Características

Aquando da especificação de requisitos de *software*, foram identificados os grupos de utilizadores abaixo que irão utilizar o produto desenvolvido por meio de diversas funcionalidades. Estes terão diferentes níveis de acesso na plataforma, bem como um conjunto de diferentes funcionalidades disponíveis.

Em seguida, são descritos os diferentes grupos de utilizadores e características associadas:

- *Administrator* – responsável por controlar os utilizadores da plataforma, bem como gerir todas as instituições ALERT;
- *Local Coordinator* – responsável por definir os planos de formação para as instituições ALERT a que pertence, nomeadamente, controlo de *training plans*, certificações e relatórios. Responsável por controlar os *Trainers*;

- *Team Manager* – responsável por gerir os cursos e as atividades que os membros da sua equipa devem frequentar;
- *Training Manager* – responsável por gerir e atribuir formações aos *Trainers*, bem como pela criação de módulos, cursos, grupos e atividades com o intuito de facilitar o modo de aprendizagem da formação presencial e não presencial. Responsável por gerir inscrições, certificados e relatórios sobre as formações efetuadas por dia, mês e ano;
- *Trainer* – responsável por criar módulos, cursos e atividades na plataforma. Encontram-se associados a várias sessões e são responsáveis por regular as inscrições dos *Trainees*, fazendo um *double-check* quando a formação é presencial;
- *Trainee* - responsável pela inscrição em cursos quando a mesma não é automática, pela confirmação da sua presença e realização de atividades.

4.2 Vista de Casos de Uso

Tendo em consideração a secção 2.7, esta secção apresenta a arquitetura do *software* sob um ponto de vista dos seus cenários de utilização. É apresentado o registo dos casos de uso do *software*, e a proposta de implementação de determinados casos de uso com maior impacto na estrutura arquitetural por grupos de utilizadores.

A Figura 23 mostra o diagrama de casos de uso associado ao *Administrator*. Este perfil é o único com acesso a todas as funcionalidades da plataforma desenvolvida.

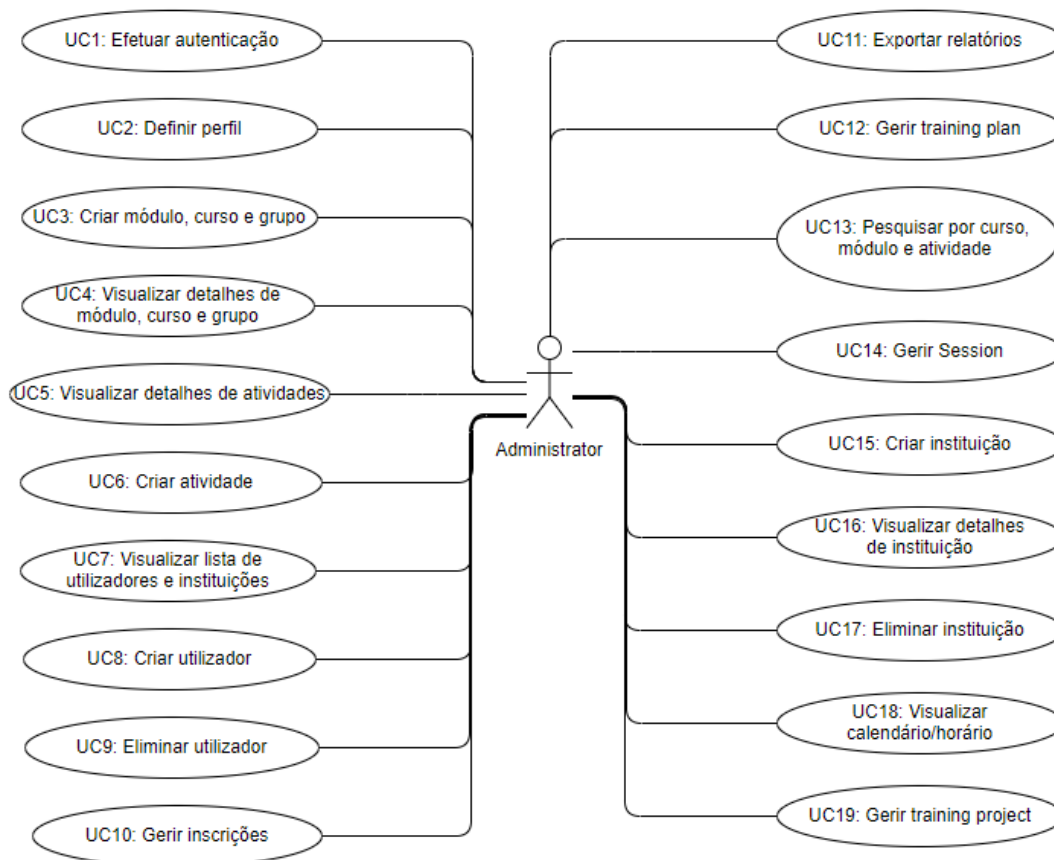


Figura 23 - Diagrama de casos de uso - *Administrator*

Com base na figura anterior (Figura 23), foi elaborada a Tabela 31 que permite identificar os grupos de utilizadores com permissão para os casos de uso expostos.

Tabela 31 - Tabela que apresenta as permissões dos restantes grupos de utilizadores para os casos de uso expostos

Caso de Uso	Local Coordinator	Training Manager	Team Manager	Trainer	Trainee
UC1: Efetuar autenticação	Sim	Sim	Sim	Sim	Sim
UC2: Definir perfil	Sim	Não	Não	Não	Não
UC3: Criar módulo, curso e grupo	Sim	Sim	Não	Sim	Não
UC4: Visualizar detalhes de módulo, curso e grupo	Sim	Sim	Sim	Sim	Sim
UC5: Visualizar detalhes de atividades	Sim	Sim	Sim	Sim	Sim
UC6: Criar atividade	Sim	Sim	Não	Sim	Não

Caso de Uso	Local Coordinator	Training Manager	Team Manager	Trainer	Trainee
UC7: Visualizar lista de utilizadores e instituições	Sim	Não	Não	Não	Não
UC8: Criar utilizador	Sim	Não	Não	Não	Não
UC9: Eliminar utilizador	Não	Não	Não	Não	Não
UC10: Gerir inscrições	Sim	Sim	Sim	Sim	Sim
UC11: Exportar relatórios	Sim	Sim	Sim	Sim	Não
UC12: Gerir <i>training plan</i>	Sim	Sim	Não	Não	Não
UC13: Pesquisar por curso, módulo e atividade	Sim	Sim	Sim	Sim	Sim
UC14: Gerir <i>Session</i>	Sim	Sim	Não	Sim	Não
UC15: Criar instituição	Sim	Não	Não	Não	Não
UC16: Visualizar detalhes de instituição	Sim	Sim	Sim	Sim	Sim
UC17: Eliminar instituição	Não	Não	Não	Não	Não
UC18: Visualizar calendário/horário	Sim	Sim	Sim	Sim	Sim
UC19: Gerir <i>training project</i>	Sim	Sim	Não	Não	Não

Através da tabela anterior, conclui-se que:

- O *Local Coordinator* não tem permissões para eliminar utilizadores e instituições da plataforma, sendo estas funcionalidades da competência de um *Administrator*. Os casos de uso 2, 7, 8, 12, 14, 15 e 17, são exemplos de restrições para um *Local Coordinator*, ou seja, enquanto que um *Administrator* gere todos os utilizadores, instituições, *sessões*, *training projects* e *training plans* da plataforma, o *Local Coordinator* apenas gere das instituições a que pertence;
- O *Training Manager* é idêntico ao *Local Coordinator* com a exceção dos seguintes casos de uso: definir perfis, criar utilizador, criar instituição e visualizar lista de utilizadores e instituições;
- O *Team Manager* está condicionado a requisitos que o ajudem a acompanhar a sua equipa/grupo de trabalho não tendo assim, as funcionalidades principais de um TMS. Pode exportar relatórios personalizados e pesquisar na plataforma. No futuro, pretende-se que este perfil possua mais funcionalidades que o *Trainee*;
- O *Trainer*, contrariamente ao perfil *Team Manager*, pode executar algumas operações CRUD (*Create, Read, Update, Delete*), nomeadamente, criar módulo, curso, grupo e

atividade. Pode exportar relatórios, gerir sessões (apenas as que irá lecionar) e pesquisar na plataforma;

- O *Trainee* pode iniciar sessão na plataforma, visualizar detalhes de cursos, módulos, atividades e dos grupos em que se encontra. Para além disso, pode inscrever-se em sessões e não tem permissão para exportar relatórios. O mais importante deste perfil é a consulta do horário ou visualizar em tabela as sessões que terá de comparecer.

Tendo por base a Figura 23 e a Tabela 31, onde são apresentados todos os casos de uso do sistema e os respetivos atores, define-se em concreto os mesmos:

- (UC1) **Efetuar autenticação:** os utilizadores registados no sistema podem e devem efetuar autenticação através das suas credenciais (*username* e *password*) por todos os utilizadores;
- (UC2) **Definir perfil:** cenário de definição das permissões de um utilizador registado na plataforma pelo *Administrator* e *Local Coordinator*;
- (UC3) **Criar módulo, curso e grupo:** cenário de criação de um módulo, curso e grupo por todos os utilizadores exceto *Team Manager* e *Trainee*;
- (UC4) **Visualizar detalhes de módulo, curso e grupo:** cenário de listagem de módulos, cursos e grupos por todos os utilizadores. Esta listagem deve permitir opções de filtragem por data, estado do curso, entre outros;
- (UC5) **Visualizar detalhes de atividades:** cenário de listagem de atividades por todos os utilizadores. Esta listagem deve permitir opções de filtragem por nome e data;
- (UC6) **Criar atividade:** cenário de criação de uma atividade por todos os utilizadores exceto *Team Manager* e *Trainee*. Uma atividade é um conteúdo ou exercício a ser realizado durante os cursos;
- (UC7) **Visualizar lista de utilizadores e instituições:** cenário de listagem de utilizadores e instituições pelo *Administrator* e *Local Coordinator*;
- (UC8) **Criar utilizador:** cenário de criação de um utilizador na plataforma pelo *Administrator* e *Local Coordinator*;
- (UC9) **Eliminar utilizador:** cenário de eliminação de um utilizador da plataforma pelo *Administrator*;
- (UC10) **Gerir inscrições:** cenário de gestão das inscrições por todos os utilizadores. Se uma sessão for do tipo “presencial” então os utilizadores devem utilizar a aplicação para confirmar a sua presença;

- (UC11) **Exportar relatórios:** cenário que permite exportar relatórios personalizáveis por todos os grupos de utilizadores exceto o de *Trainees*. Como exemplo da exportação de um relatório em formato CSV seria o de um *training plan*;
- (UC12) **Gerir *training plan*:** cenário que permite efetuar operações CRUD sobre um *training plan* pelo *Admininistrator*, *Local Coordinator* e *Training Manager*. Na vista de detalhes deste requisito, deve ser possível consultar as formações que estão a decorrer/iniciar, bem como o histórico de formações já terminadas, entre outros;
- (UC13) **Pesquisar por curso, módulo e atividade:** cenário que permite pesquisar por curso, módulo e atividade por todos os utilizadores;
- (UC14) **Gerir *session*:** cenário que permite efetuar operações CRUD sobre uma sessão de formação pelo *Admininistrator*, *Local Coordinator* e *Training Manager*. As sessões estão associadas a um *training plan* e curso;
- (UC15) **Criar instituição:** cenário que permite criar uma instituição pelo *Administrator* e *Local Coordinator*.;
- (UC16) **Visualizar detalhes de instituição:** cenário de listagem de instituições pelo *Administrator*, *Local Coordinator* e *Training Manager*;
- (UC17) **Eliminar instituição:** cenário que permite a eliminação de uma instituição da plataforma pelo *Administrator*;
- (UC18) **Visualizar calendário/horário:** cenário de listagem de sessões por todos os utilizadores. Este caso de uso deverá apresentar duas opções, vista em tabela ou horário;
- (UC19) **Gerir *training project*:** cenário que permite gerir *training projects*, efetuando operações CRUD pelo *Administrator*, *Local Coordinator* e *Training Manager*.

Face ao que foi descrito e enumerado nos parágrafos anteriores, é importante garantir a autenticidade e integridade dos dados dos utilizadores que navegam na plataforma. Primeiramente, deve existir um mecanismo de autenticação e autorização (processo que ocorre após ser validada a autenticação, isto é, diz respeito aos privilégios que são concedidos a determinado utilizador quando utiliza a plataforma).

Devem ser adicionados mecanismos de encriptação. Este é o processo de transformar uma informação de forma a que alguém indesejado não consiga ler; será necessária uma chave especial (chamada HASH) que permita descodificar essa informação, revertendo-a quando necessário para a sua forma original. Através destes mecanismos de encriptação, será possível proteger os dados dos utilizadores, os conteúdos partilhados, os horários das formações ainda por iniciar ou a decorrer e o histórico de planos de formação (*training plans*) desenvolvidos.

Assim, toda a informação criada ou editada deve ser armazenada numa localização segura e desconhecida ao mundo exterior.

Por fim, sobre o protocolo de comunicação, caso a ALERT pretenda adquirir um certificado do tipo SSL/TLS (*Secure Sockets Layer/Transport Layer Security*) - essencial quando são transmitidas informações pessoais -, pode ser utilizado o protocolo HTTPS (extensão segura do protocolo HTTP) para estabelecer uma configuração segura com o servidor.

4.3 Requisitos Não Funcionais

Os Requisitos Não Funcionais (RNF) são as restrições de qualidade que o sistema deve satisfazer. Esta secção do presente documento especifica alguns dos requisitos não funcionais do sistema, nomeadamente:

4.3.1.1 Usabilidade

Atualmente, interagir com sistemas Web tornou-se uma necessidade comum, o que indica que existe uma adesão crescente das organizações em se adaptarem às novas tecnologias para informatizar os seus processos de formação. Neste sentido, os requisitos de usabilidade são críticos para o sucesso de qualquer sistema, nomeadamente:

- Facilidade na aprendizagem: o colaborador ou cliente ALERT deve aprender a utilizar o sistema em pouco tempo. O sistema deve ser estruturado de maneira a que os utilizadores interajam com ele intuitivamente através das suas funções;
- Eficiência da tarefa: o utilizador da plataforma a desenvolver deve poder terminar uma determinada tarefa num prazo especificado (ou numa quantidade de cliques do *mouse*);
- Facilidade de recordação: o utilizador da plataforma a desenvolver deve poder recordar como se realizam determinadas tarefas, após um tempo especificado de não utilização do sistema;
- Entendimento: o utilizador da plataforma deve entender as mensagens e notificações do sistema e o que o sistema faz;
- A interface deve ser intuitiva, responsiva e apelativa.

4.3.1.2 Confiabilidade

A confiabilidade inclui a habilidade do sistema para continuar a funcionar sob circunstâncias adversas. Por exemplo, o utilizador encontra-se a definir um novo *training plan* ou *training project*, e após inserir todas as informações, ocorre um erro no sistema e as informações

inseridas são perdidas. No contexto deste projeto, o sistema deve ser capaz de tratar estas exceções e recuperar de falhas, sem que haja perda de dados. Backup e restauração da base de dados também se encaixam nesta secção.

4.3.1.3 Desempenho

Pouco importa ter um sistema seguro, interativo e confiável se o mesmo consome muitos recursos do computador e o seu tempo de resposta aumenta aquando a execução dos processamentos. Esta limitação está dependente dos equipamentos utilizados pela ALERT no alojamento do *software*. Para esta plataforma será necessário armazenar grandes quantidades de informação e, por isso, é preciso ter em atenção o espaço disponível nos dispositivos.

Um sistema lento é alvo de crítica dos utilizadores, mesmo que seja funcional. Assim, para melhorar a performance do *software* é necessário utilizar técnicas de programação orientada a objetos e otimização de código (diminuir o número de métodos invocados, e também de variáveis e constantes a serem acedidas em memória, consequentemente melhorando o desempenho do processamento da plataforma).

4.3.1.4 Escalabilidade

O sistema deverá ser desenvolvido e ao longo de todo o processo ter capacidade para crescer, como por exemplo, ter um código e uma arquitetura que é fácil de dar manutenção e de aumentar as suas funcionalidades.

Futuramente, com a possibilidade de aumento do número de utilizadores na plataforma, o *software* deverá estar preparado para se adaptar a este crescimento e manter as suas funcionalidades sem impacto no desempenho das mesmas.

4.3.1.5 Reutilização

O desenvolvimento de um sistema é tipicamente conhecido como um processo caro e lento. Assim, o sistema deve permitir reutilizar código para que o mesmo possa ser utilizado novamente aquando a adição de novas funcionalidades com pequenas, ou nenhuma, modificações. Adicionalmente, deve ser possível reutilizar componentes de *software* existentes, com especificações de interface concisas e documentação atualizada.

4.3.1.6 Restrições de desenho

Apesar do estudo apresentado na secção 2.6, existem restrições no desenho da plataforma, pois deve ser adotado o mesmo *layout* que os restantes produtos ALERT® eLEARNING e ALERT®

eLIBRARY (secção 4.6). Quanto à estrutura do *back-end*, também existem restrições para as ferramentas de desenvolvimento a utilizar, uma vez que, as mesmas devem ir de encontro às utilizadas, atualmente, no processo de desenvolvimento de *software* da ALERT, isto é, no lado do servidor linguagem Java e base de dados Oracle (secção 5).

4.3.1.7 Requisitos de implementação

Do lado do servidor, como referido em 4.3.1.6, deve ser utilizado linguagem Java e base de dados Oracle, seguindo as tecnologias e ferramentas de desenvolvimento de *software* da ALERT. Como esperado, do lado do cliente não existem restrições de implementação, no entanto, é aguardado que sejam aplicados bons princípios da engenharia de *software* que constituem a base dos métodos, tecnologias, metodologias e ferramentas adotadas na prática de desenvolvimento de soluções de *software* (considerar o rigor, a modularidade do sistema, entre outros).

4.4 Modelo de Domínio

Na Figura 24 é apresentando o modelo de domínio com as entidades e respetivas relações para o desenvolvimento do projeto. Por uma questão de simplicidade e melhor compreensão do diagrama alguns dos atributos foram omitidos.

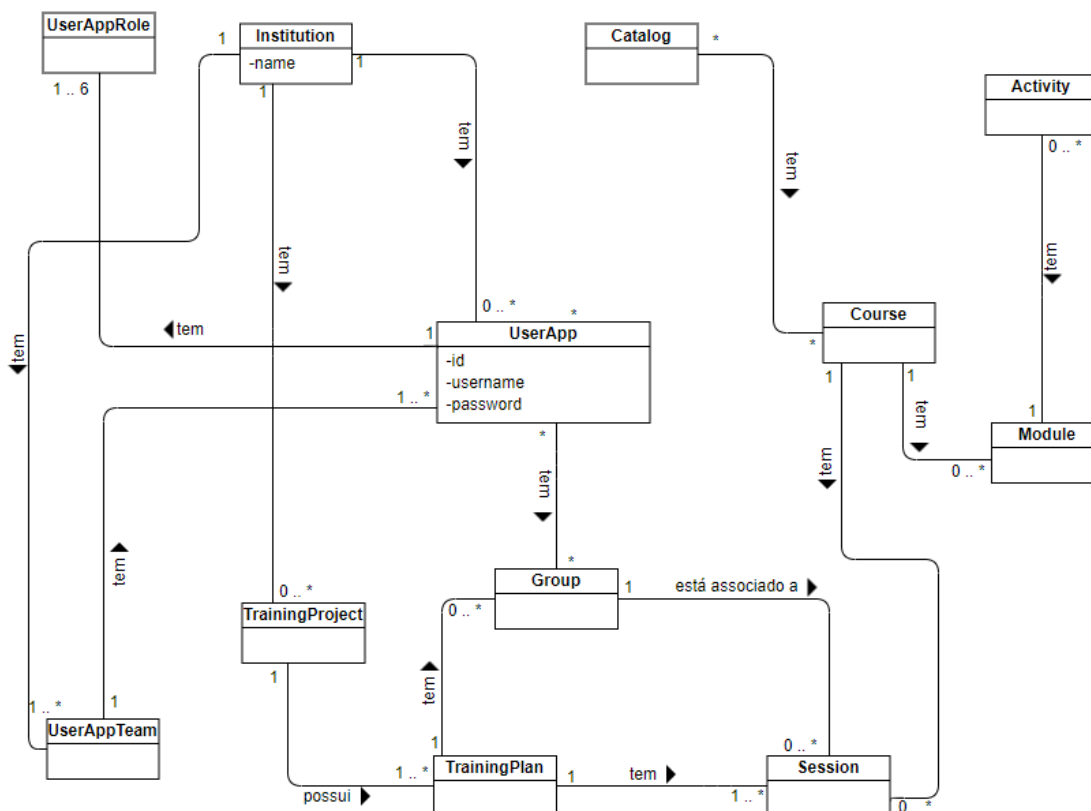


Figura 24 - Modelo de Domínio do sistema

De forma a compreender o modelo desenhado na figura acima (Figura 24), foi elaborada a Tabela 32 que apresenta uma breve descrição de cada entidade.

Tabela 32 - Descrição das entidades do modelo de domínio

Entidade	Descrição
UserApp	Entidade que representa o utilizador do sistema
Institution	Entidade que representa uma instituição ALERT
UserAppRole	Entidade que representa o perfil de utilizador do sistema
UserAppTeam	Entidade que representa a equipa do utilizador
Group	Entidade que representa o grupo do utilizador para uma ou mais sessões
TrainingProject	Entidade que representa o TrainingProject associado a uma instituição

Entidade	Descrição
TrainingPlan	Entidade que representa o TrainingPlan associado a um TrainingProject
Course	Entidade que representa um curso
Module	Entidade que representa o módulo de um curso
Activity	Entidade que representa uma atividade (ação de formação)
Session	Entidade que representa uma sessão para um TrainingPlan
Catalog	Entidade que representa um catálogo

Relativamente à Tabela 32, o mais importante é compreender que uma instituição possui muitos *Training Projects* e utilizadores. Um *Training Project* pode conter vários *Training Plans* que, por sua vez, tem vários utilizadores (divididos por grupos) inscritos em uma ou mais sessões (data e hora de realização das formações para um ou mais cursos). Através da lista de sessões associadas a um *Training Plan*, é possível construir um calendário/horário personalizado. Adicionalmente, um curso tem associado a si módulos de aprendizagem, que, logicamente, contém atividades para realizar no decorrer ou após término da sessão. Por fim, um curso pode pertencer a um ou mais catálogos.

4.5 Vista de Dados

Esta secção apresenta a arquitetura de *software* sob um ponto de vista de persistência de dados, com um alto nível de abstração. A persistência dos dados segue uma abordagem relacional. Na Figura 25 são definidas as entidades e respetivas relações.

Entidade	Descrição
TRAINING_PLAN	Tabela responsável pela persistência de dados dos <i>Training Plans</i> dos <i>Training Project</i> do sistema
SESSION_MEETING	Tabela responsável pela persistência de dados das sessões dos <i>Training Plans</i> do sistema
TEAM_GROUP	Tabela responsável pela persistência de dados dos grupos dos <i>Training Plans</i> do sistema
TEAM_GROUP_USER_APP	Tabela responsável pela persistência de dados de associação entre <i>Team_Group</i> e <i>User_App</i> do sistema
COURSE	Tabela responsável pela persistência de dados dos cursos do sistema
COURSE_MODULE	Tabela responsável pela persistência de dados de associação entre <i>Course</i> e <i>Module</i> do sistema
MODULE	Tabela responsável pela persistência de dados dos módulos dos cursos do sistema
MODULE_ACTIVITY	Tabela responsável pela persistência de dados de associação entre <i>Module</i> e <i>Activity</i> do sistema
ACTIVITY	Tabela responsável pela persistência de dados das atividades dos módulos do sistema
CATALOG_TRAINING	Tabela responsável pela persistência de dados dos catálogos do sistema
CATALOG_COURSE	Tabela responsável pela persistência de dados de associação entre <i>Catalog</i> e <i>Course</i> do sistema
TEAM_DEPARTMENT	Tabela responsável pela persistência de dados das equipas dos utilizadores do sistema

Como ilustrado na Figura 25, a sua estrutura é semelhante à apresentada na secção 4.4, excetuando os nomes de algumas tabelas, uma vez que, a própria implementação do JPA se encarrega de criar as tabelas automaticamente na base de dados relacional. Isso significa que as tabelas intermédias também são geradas (como ilustrado e descrito na Tabela 33).

Assim, uma instituição (tabela “INSTITUTION”) é criada por um utilizador (tabela “USER_APP”). Como os dois lados devem ter a possibilidade de referenciar o outro, é criada uma tabela intermédia que contém as chaves estrangeiras. Contudo, é necessário definir o *owner side*, que, para este caso, foi a tabela da instituição através da anotação “@JoinTable” e as chaves estrangeiras com as anotações “@JoinColumn” (atributo que se conecta ao *owner side* e *child side* do relacionamento).

Com uma estrutura semelhante, temos:

- Um *Training Plan* possui vários grupos (tabela “TEAM_GROUP”) com utilizadores. Como esta é uma relação de muitos para muitos, é criada uma tabela intermédia com as chaves primárias de ambos os lados. Para este caso em particular, a tabela “TEAM_GROUP” representa o *owner side*. Assim, quando se adiciona uma sessão à plataforma, é possível associar de imediato um grupo do *training plan* escolhido, contrariamente, a escolher utilizador por utilizador;
- Um curso (tabela “COURSE”) possui um ou mais módulos (tabela “MODULE”) o que significa que existe uma tabela intermédia. Neste caso, a tabela cursos representa o *owner side*;
- Um módulo tem várias atividades (tabela “ACTIVITY”) que devem ser realizadas no final ou depois do término de uma sessão. Assim, é gerada uma tabela intermédia onde a tabela “MODULE” representa o *owner side*;
- A tabela “CATALOG_TRAINING” regista o nome, o acrónimo e o dia e hora a que o mesmo foi criado. Esta representa o *owner side* da relação com a tabela “COURSE”. Assim, é possível, por exemplo, encapsular os cursos mais importantes de um determinado ano e, mais tarde, consultar todos os seus detalhes de forma rápida.

Um utilizador pertence a uma equipa (tabela “TEAM_DEPARTMENT”). Esta tabela é necessária porque um *Team Manager* deve poder acompanhar e monitorizar os seus membros de equipa e, se assim entender, efetuar a inscrição dos mesmos nas sessões dos cursos.

A tabela “TRAINING_PROJECT” é uma tabela que regista os projetos de formação para as instituições. Por consequência, esta tabela possui a chave primária da tabela “INSTITUTION”, uma vez que, estamos perante uma relação de um para muitos. Com uma estrutura semelhante, temos:

- A tabela “TRAINING_PLAN” que regista os planos de formação (*training plans*) para os *training projects* das instituições. Como exemplo, um *training project* do tipo “INTERNAL TRAINING” contém o plano de formação do ano de 2020 e o plano de formação do ano 2019. Para este caso em particular, a tabela “TRAINING_PLAN” possui a chave primária da tabela “TRAINING_PROJECT”;

- A tabela “SESSION_MEETING” está ligada à tabela “TRAINING_PLAN”. Para este caso em específico, a primeira tabela guarda a chave primária das tabelas: “TRAINING_PLAN”, “COURSE” e “TEAM_GROUP”, uma vez que, permite gerir a verdadeira essência de um TMS ajudando a controlar os utilizadores que estão previstos, foram inscritos e apareceram. Com esta tabela é ainda possível registar o formador, associar uma sessão a um curso e controlar as inscrições.

4.6 Interface Gráfica

A análise de requisitos é uma fase crítica de um projeto de software, afinal, é neste momento que são definidas as principais características do sistema e negócio que um projeto deverá atender (Alff, 2018). No contexto deste projeto, foi utilizada a ferramenta MockFlow (2020) para desenhar protótipos de interface do sistema como referência base para o desenvolvimento do *layout* da aplicação. Dado que a construção desta nova plataforma na ALERT não apresenta projetos anteriores, ou seja, não existem bases e ideias definidas para a interface do produto, foram seguidas algumas apreciações do estudo efetuado na secção 2 (Estado da Arte), nomeadamente, de sistemas de gestão de formação e sistemas de gestão de aprendizagem. Sobre a existência e posicionamento de determinados componentes, foi reaproveitado o *layout* do produto ALERT® eLibrary, nomeadamente, o cabeçalho, rodapé e posicionamento da barra de navegação.

Relativamente ao cabeçalho, que se destaca pela sua cor cinzenta, foi adotada uma foto circular, uma linha horizontal de cor verde no topo da página e o logo do novo produto ALERT® TMS. Quanto às cores, destaca-se a utilização predominante da cor verde em botões, hiperligações e textos, uma vez que, está diretamente relacionada com a área de *training* na ALERT.

Nas secções seguintes é ilustrado algumas das principais páginas da plataforma, bem como a explicação detalhada do planeamento efetuado para a interface gráfica.

4.6.1 Cabeçalho

O cabeçalho da plataforma (Figura 26), como já explicado, deve conter o logo desenhado pela ALERT para o sistema de gestão de formação no canto superior esquerdo, a foto de perfil do utilizador no canto superior direito e uma linha horizontal sólida de cor verde. Esta página é comum a todas as existentes na plataforma com exceção da página onde é efetuada a autenticação.

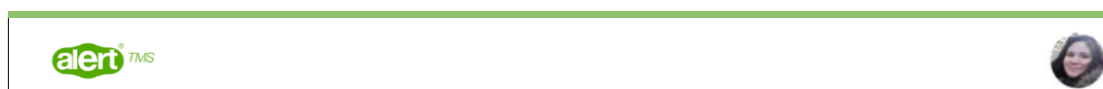


Figura 26 - Cabeçalho para a nova plataforma

4.6.2 Rodapé

O rodapé (Figura 27), comum a todas as páginas, apresenta uma referência para os direitos de autor, bem como referências para as redes sociais da ALERT. Este rodapé é idêntico aos produtos ALERT® eLEARNING.



Figura 27 - Rodapé para a nova plataforma

4.6.3 Barra de Navegação

A Figura 28 mostra como será representada a barra de navegação na plataforma, que deve estar presente em todas as páginas. Esta deverá apresentar uma hiperligação para *Dashboard* (página principal) e as seguintes *dropdowns* com o texto: *Training Project*, *Catalog* e *Add*. Para além disso, são apresentadas as seguintes componentes: caixa de pesquisa e menu com hiperligações suportadas pelo ícone na barra de navegação, à direita.

Relativamente ao comportamento das *dropdowns* com o texto *Training Project* e *Add*, serão mostradas unicamente ao *Administrator*, *Local Coordinator* e *Training Manager*. As restantes também podem ser visualizadas pelos outros grupos de utilizadores.

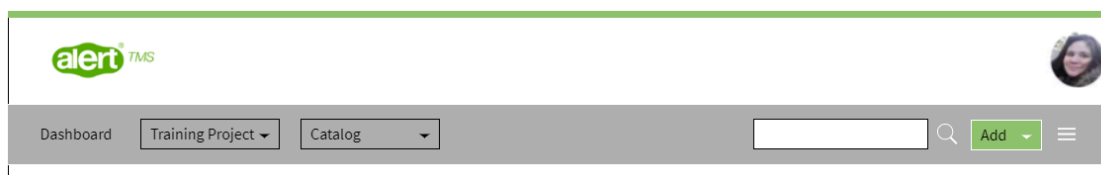


Figura 28 - Barra de navegação para a nova plataforma

A Figura 29 mostra como será representado o comportamento suportado pelo ícone na barra de navegação, à direita. É expectável que esta apresente as seguintes hiperligações de modo a responder às funcionalidades pretendidas para o novo produto ALERT® TMS: o calendário, os grupos e cursos, as instituições, os utilizadores e o *Logout*. As hiperligações das instituições e dos utilizadores apenas serão exibidas ao *Administrator*, *Local Coordinator* e *Training Manager*.

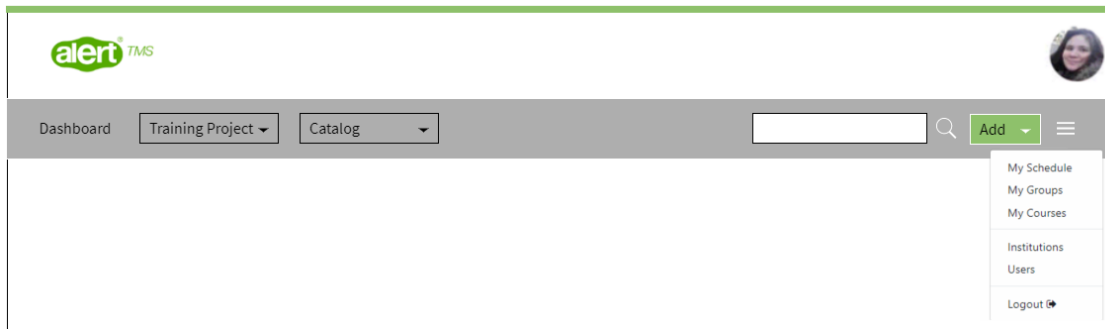


Figura 29 - Comportamento suportado pelo ícone na barra de navegação, à direita para a nova plataforma

A Figura 30 mostra como é esperado o comportamento suportado pela *dropdown* com o texto *Add*. Aquilo que efetivamente se pretende, é um mecanismo rápido para adicionar um *training project*, *training plan*, *group*, *course*, *module*, uma *activity* e sessão a qualquer momento.

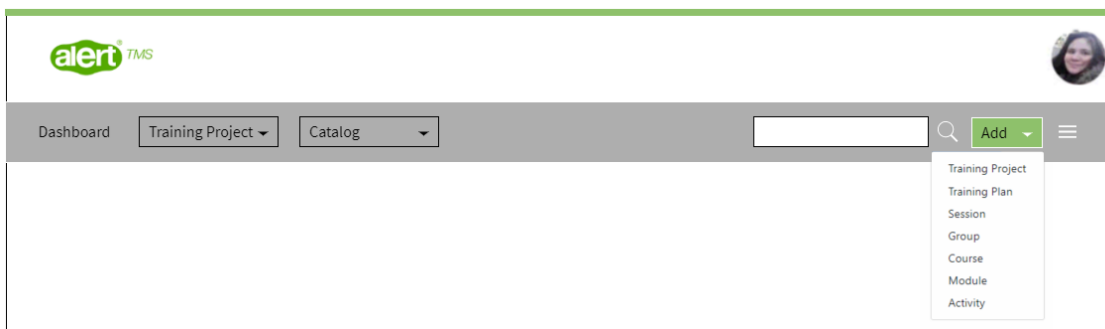


Figura 30 - Comportamento suportado pela *dropdown* com o texto *Add* para a nova plataforma

Cada hiperligação exibe uma página de registo genérica para cada cenário, como por exemplo, criar uma sessão e associar diretamente a um *training plan*. Este comportamento permite poupar tempo na procura da página do *training plan* a que se pretende adicionar a nova sessão.

4.6.4 Vista de Detalhes de uma Instituição

Na Figura 31 é apresentada a vista planeada para a página de detalhes de uma instituição.

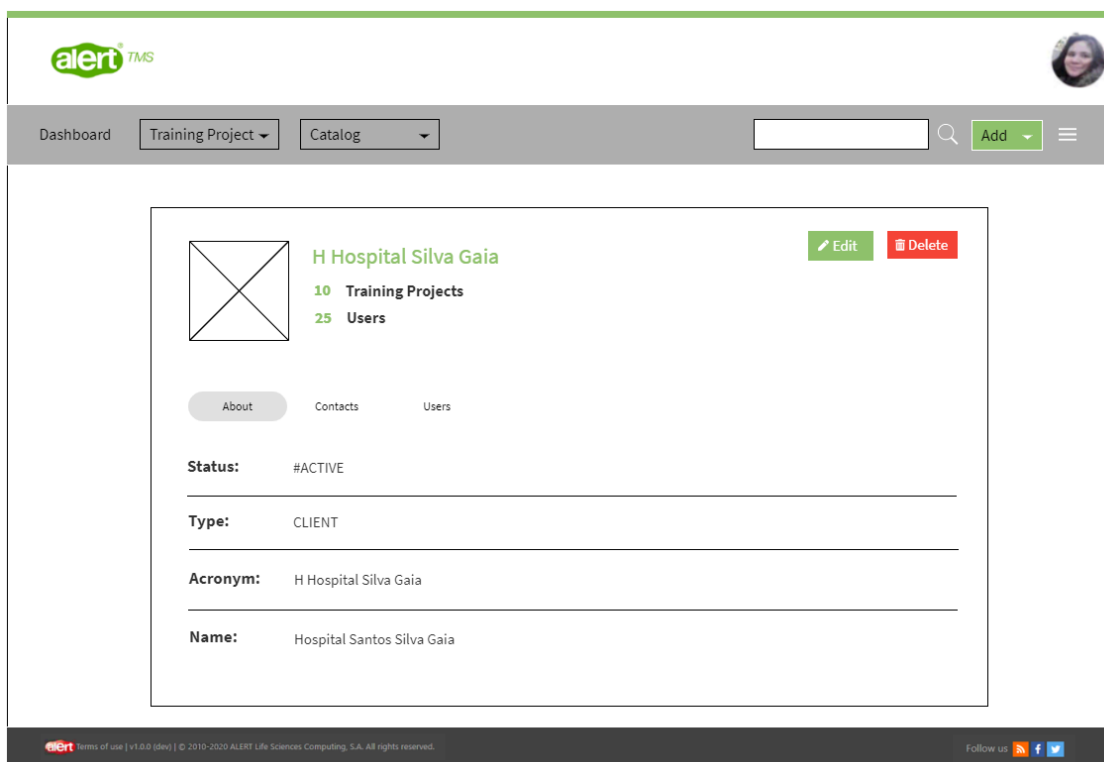


Figura 31 – Vista de detalhes de uma instituição para a nova plataforma

A vista principal de uma instituição pode ser visualizada pelos grupos de utilizador *Administrator* e *Local Coordinator*. Relativamente ao segundo grupo, que, como já referido em secções anteriores, apenas pode consultar as que pertence/gere. Para além disso, esta página deve expor dados estatísticos sobre o número de *training projects* e *users* associados à instituição. Por fim, deve ser constituída por uma *nav* com os itens *About*, *Contacts* (morada, correio eletrónico e número de telefone) e *Users* (tabela com todos os utilizadores da instituição).

4.6.5 Vista Principal de Cursos

Na Figura 32 ilustra o comportamento do catálogo de cursos para todos os grupos de utilizadores. A um curso, está associado um estado, como por exemplo, DRAFT, RETIRED, PUBLISHED ou DELETED.

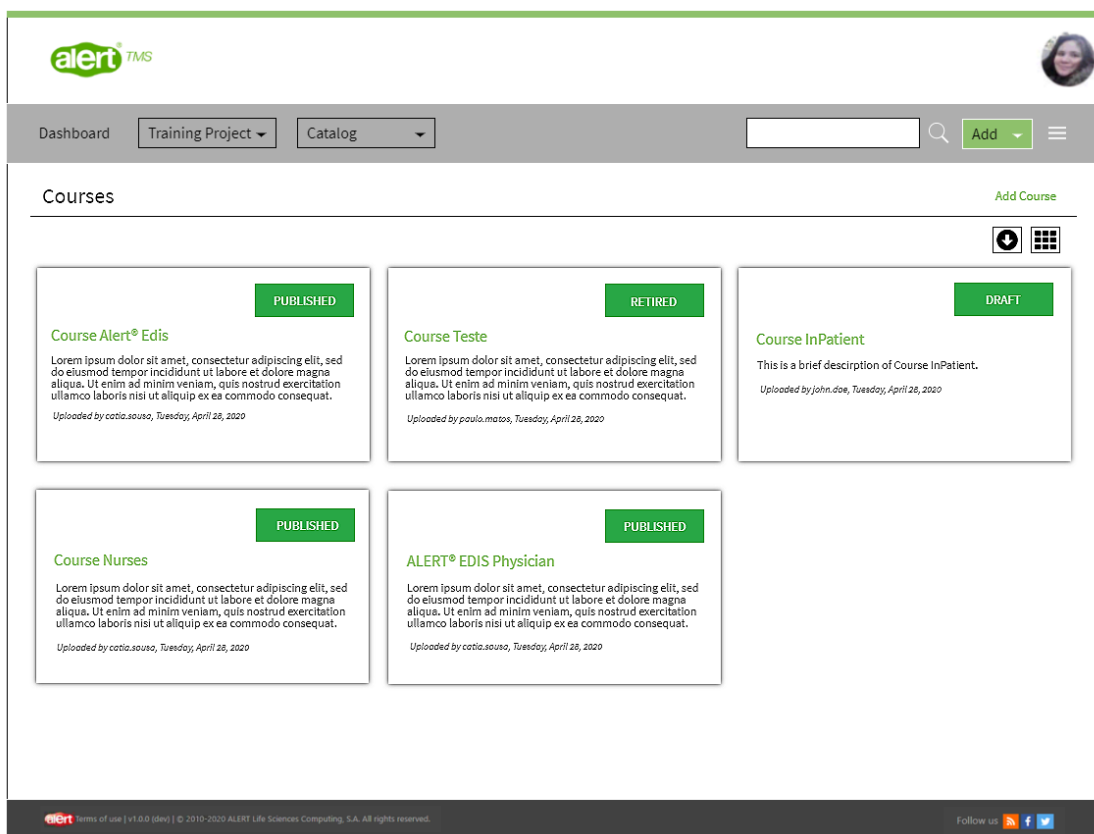


Figura 32 - Vista principal de Cursos

Os cursos que apresentem um estado diferente de PUBLISHED, são unicamente mostrados aos seguintes grupos de utilizadores: *Administrator*, *Local Coordinator*, *Trainer* e *Training Manager*. Similarmente, os botões com os ícones seta para baixo, grelha e *Add Course* apenas serão exibidos a esses grupos. O nome do curso deve ser uma hiperligação para a página de detalhes do mesmo.

4.6.6 Vista de Detalhes de um Curso

Na Figura 33 é apresentada a vista planeada para a página de detalhes de um curso. Esta página pode ser consultada por todos os grupos de utilizadores referidos na secção 4.1.


Course Alert® Edis Add Course

Status: PUBLISHED

Created At: Tuesday, April 28, 2020

Brief Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Overview: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trainer: 
Cátia Sousa

Session Modules

Session Date	Start Session	End Session	Local
Wednesday July 6	10:00	11:00	Room 1
Wednesday July 6	15:00	16:00	Room 1
Wednesday July 7	10:00	11:00	Room 1


Alert Terms of use | v1.0.0 (dev) | © 2010-2020 ALERT Life Sciences Computing, S.A. All rights reserved. Follow us: 

Figura 33 – Vista de detalhes de um curso para a nova plataforma

Os botões com o texto *Add Course* e *Retire* devem ser mostrados para os grupos de utilizador *Administrator*, *Local Coordinator*, *Training Manager* e *Trainer*. Além das propriedades apresentadas com texto a negrito, pode apresentar uma *nav* com os itens *Session* (mostrar as datas e horas das sessões que irão decorrer no futuro) e *Modules* (os que serão lecionados durante a formação).

4.6.7 Criação de um *Training Plan*

Na Figura 34 é apresentada a vista planeada para a página de criação de um *training plan*. Esta página pode ser consultada pelos grupos de utilizadores *Administrator*, *Local Coordinator* e *Training Manager*.

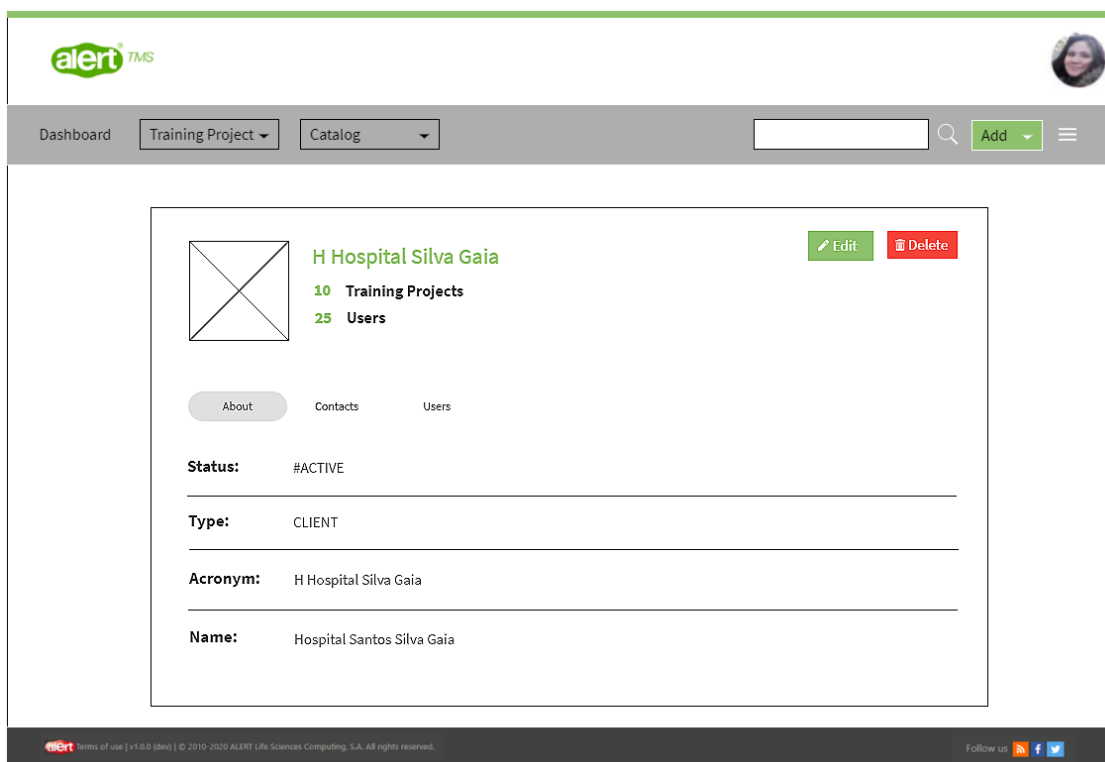


Figura 34 - Criação de um Training Plan

O objetivo desta vista é iniciar o processo de criação de um *training plan* que deverá ser associado, sempre, a um *training project* e uma instituição. Não menos obstante, caso o utilizador escolha uma instituição do tipo INTERNAL e não existir nenhuma na plataforma, o mesmo deverá ser redirecionado para a página de criação da mesma. Posteriormente, e após término do processo, deve ser redirecionado para a página de registo de um training plan e concluir o processo. Contudo, o mesmo irá acontecer com a *dropdown* do *training project* uma vez que não existirá nenhum na base de dados.

4.6.8 Vista de Detalhes de um *Training Plan*

Na Figura 35 é apresentada a vista planeada para a página de detalhes de um *Training Plan*. Esta página pode ser consultada pelos grupos de utilizadores *Administrator*, *Local Coordinator* e *Training Manager*.

The screenshot shows the 'Training Plan - Formação 2020' page in the alert TMS system. At the top, there is a navigation bar with 'Dashboard', 'Training Project', and 'Catalog' dropdown menus, a search bar, and an 'Add' button. The main content area features the title 'Training Plan - Formação 2020' and an 'Edit Training Plan' link. Below the title, there is a box for 'Training Project of this Training Plan' containing 'Training Project Z'. To the right, there are three buttons: 'Add Session', 'Add Group', and 'View Schedule'. Below these buttons, there is a user profile section with a photo and the text 'Uploaded by catia.sousa, Tuesday, April 28, 2020'. At the bottom, there is a table with the following data:

Group Name	Number of Users	
EU.1	6	
EU.5	10	
EU.4	0	

The footer contains the alert logo, terms of use, and social media links.

Figura 35 – Vista de detalhes de um training plan para a nova plataforma

Esta página deve apresentar um *layout* idêntico ao ilustrado na Figura 33. É importante manter esta coerência na plataforma pela questão da usabilidade (requisito não funcional detalhado na seção 4.3.1.1. Os botões com o texto *Add Session* e *Add Group* redirecionam o utilizador para páginas de criação da respetiva operação para o *training plan* em que se encontra. O botão com o texto *View Schedule* permite visualizar o horário com todas as sessões por concluir ou concluídas do *training plan* consultado.

Por fim, é expectável que seja exibida a foto de perfil do utilizador que criou o *training plan*, bem como o dia em que efetuou essa operação. Este dado, que apesar de não ser estatístico, permite controlar o fluxo do *training plan* e solicitar o utilizador caso seja necessário colocar alguma dúvida.

4.6.9 Criação de uma Sessão

Na Figura 36 é apresentada a vista de criação de uma Sessão. Esta, como já referido em parágrafos anteriores, pode ser pedida pelos grupos de utilizadores *Administrator*, *Local Coordinator* e *Training Manager*.

The screenshot shows the 'Create new Session' form in the alert TMS interface. The form is titled 'Create new Session' and is located in the main content area. The form fields are as follows:

- Institution Type*:** CLIENT
- Institution (Client)*:** Hospital Santos Silva Gaia
- Training Project*:** Training Project Y
- Training Plan Name*:** Training Plan Teste
- Course*:** Course Teste
- Name*:** Session Teste
- Group*:** EU-5
- Trainees Planned*:** 10
- Session Status*:** IN_PROGRESS
- Add Session Parts*:**
 - Start Time*:** [Empty text input]
 - End Time*:** [Empty text input]
 - Add Session Part:** [Green button]
- Trainer:** IN_PROGRESS
- Resources:** Computers
- Local:** Room 1
- Details:** [Empty text area]

At the bottom right of the form, there are two buttons: 'Submit' (green) and 'Cancel' (red). The 'Submit' button is currently disabled.

Figura 36 – Criação de uma sessão para a nova plataforma

De modo a ser possível criar uma sessão, o botão com o texto *Submit* deve ser desbloqueado quando as seguintes informações forem preenchidas: *institution type*, *institution (client)*, *training project*, *training plan name*, *course*, *name*, *group*, *trainees planned*, *sessions status*, *sessions parts* (deve existir pelo menos uma). Os restantes campos não são obrigatórios ficando ao critério do utilizador preencher no momento, ou mais tarde.

5 Arquitetura

Esta secção apresenta a arquitetura de *software* sob um ponto de vista lógico com alto nível de abstração. Assim, são apresentados os diagramas de componentes, objetivos e restrições arquiteturais, tecnologias e processos de desenvolvimento, a vista lógica e de implantação.

5.1 Diagrama de Componentes

A Figura 37 mostra ao pormenor a estrutura das componentes e respetivas relações. Com base nela, é possível interpretar o funcionamento interno do *software*.

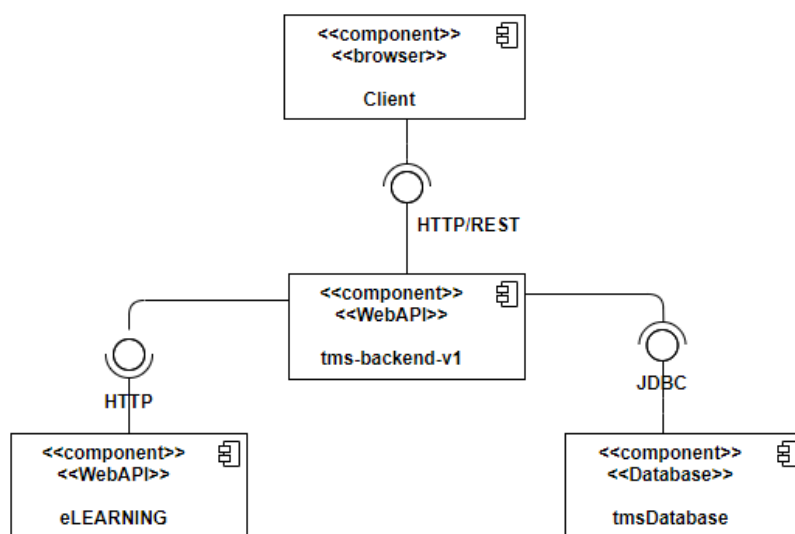


Figura 37 – Diagrama de componentes

Com base na figura anterior (Figura 37), em seguida são descritas as principais componentes:

- eLEARNING – componente representativa da API do produto ALERT® eLEARNING;
- tms-backend-v1 – componente representativa da API do sistema a desenvolver. É onde se encontra toda a lógica de negócio. Esta componente comunica com tmsDatabase, consome serviços de eLEARNING e disponibiliza serviços a *Client*;
- tmsDatabase – componente representativa da base de dados da aplicação Web;
- Client - componente representativa da aplicação Web que consome serviços de tms-backend-v1.

Nesta arquitetura é adotado o modelo Cliente-Servidor. Do lado do servidor, temos a componente “tms-backend-v1”, uma RESTful API que utiliza Spring Boot 2.2 para disponibilizar serviços Web a outras componentes. É nesta componente que se encontra a lógica de negócio do sistema. Do lado do cliente, temos a componente “*Client*” que representa o dispositivo cliente responsável por exibir a interface gráfica através de um *browser*, que consome serviços Web utilizando o protocolo de comunicação HTTP.

A componente “tms-backend-v1” utiliza a *framework* Hibernate e implementa JPA (*Java Persistence API*). Por meio do JPA, é possível mapear, armazenar, atualizar e recuperar dados da base de dados para objetos Java e vice-versa. Assim, JPA permite trabalhar diretamente com objetos e não com instruções SQL facilitando o trabalho do programador. Como referido na secção 4.3.1.6, a componente “tmsDatabase” representa o sistema de gestão de base de dados da Oracle.

Por fim, a API “tms-backend-v1” (plataforma de gestão da formação) necessita de consumir os serviços Web disponibilizados pela componente “eLEARNING” (SGA) através do protocolo de comunicação HTTP.

5.2 Objetivos e Restrições Arquiteturais

Para a implementação da plataforma não foram colocados quaisquer tipos de restrições arquiteturais, nem tecnológicas. Foi única e exclusivamente indicado que o principal objetivo incide numa aplicação que deve executar num ambiente *web*. Posto isto, os utilizadores desta plataforma necessitarão de um *web browser* instalado no seu dispositivo como, por exemplo, o Google Chrome, Mozilla Firefox, Safari, Waterfox, entre outros. Para que seja possível aceder à plataforma, é necessária ligação à Internet.

5.3 Tecnologias e Processos de Desenvolvimento

Esta secção do presente documento permite compreender as tecnologias selecionadas e justificar o porquê das mesmas.

5.3.1 Tecnologias Selecionadas

Relativamente ao que já foi referido na secção 5.1, é importante reforçar que para o desenvolvimento da solução pretendida a opção recaiu sobre uma plataforma web onde:

- O *front-end* é desenvolvido com recurso à *framework* Angular (versão 8), seguindo o padrão MVVM (*Model*, *View* e *ViewModel*). Este é um padrão que aumenta a legibilidade e manutenção do código, dividindo-o nas secções mencionadas anteriormente;
- Posteriormente, o *back-end* é constituído por uma REST API com recurso a Spring Boot 2.2. Este seguirá o padrão de separação em camadas, nomeadamente, a camada de apresentação que receberá pedidos REST (primeiro na camada *Security* e, após confirmada a autorização, reencaminhará o pedido para a camada *Controller*), a camada *Service* que receberá pedidos do *Controller* e, por último, a camada de persistência (*Repository*) que irá fazer a gestão dos dados do sistema. Estes serão persistidos seguindo um modelo relacional com recurso a uma base de dados Oracle.

5.3.2 Justificação

Como referido na secção 5.2, apesar de não existirem quaisquer tipo de restrições arquiteturais para o desenvolvimento deste projeto, era importante ter em consideração as tecnologias utilizadas pelo ALERT® eLEARNING (principalmente, para o *back-end*) e pelo ALERT® eLibrary. Adotar as mesmas tecnologias foi uma decisão determinante devido aos objetivos futuros, que, como ilustrado no diagrama de componentes da secção 5.1, pretende-se integrar o ALERT® eLEARNING com o novo produto ALERT® TMS. Até há data de entrega deste documento, essa ligação não foi executada com sucesso uma vez que o ALERT® eLEARNING apresenta um tipo de tecnologia e linguagens antigas, dificultando assim a integração.

Relativamente à preferência por um modelo cliente-servidor e não micro serviços, a escolha voltou a recair por uma arquitetura igual ao produto ALERT® eLibrary.

5.4 Vista Lógica

Seguindo boas práticas de engenharia de *software* e com base no estudo da secção 2, a Figura 38 apresenta a vista lógica da plataforma identificada na secção 5.1 com exclusão da ligação com o ALERT® eLEARNING.

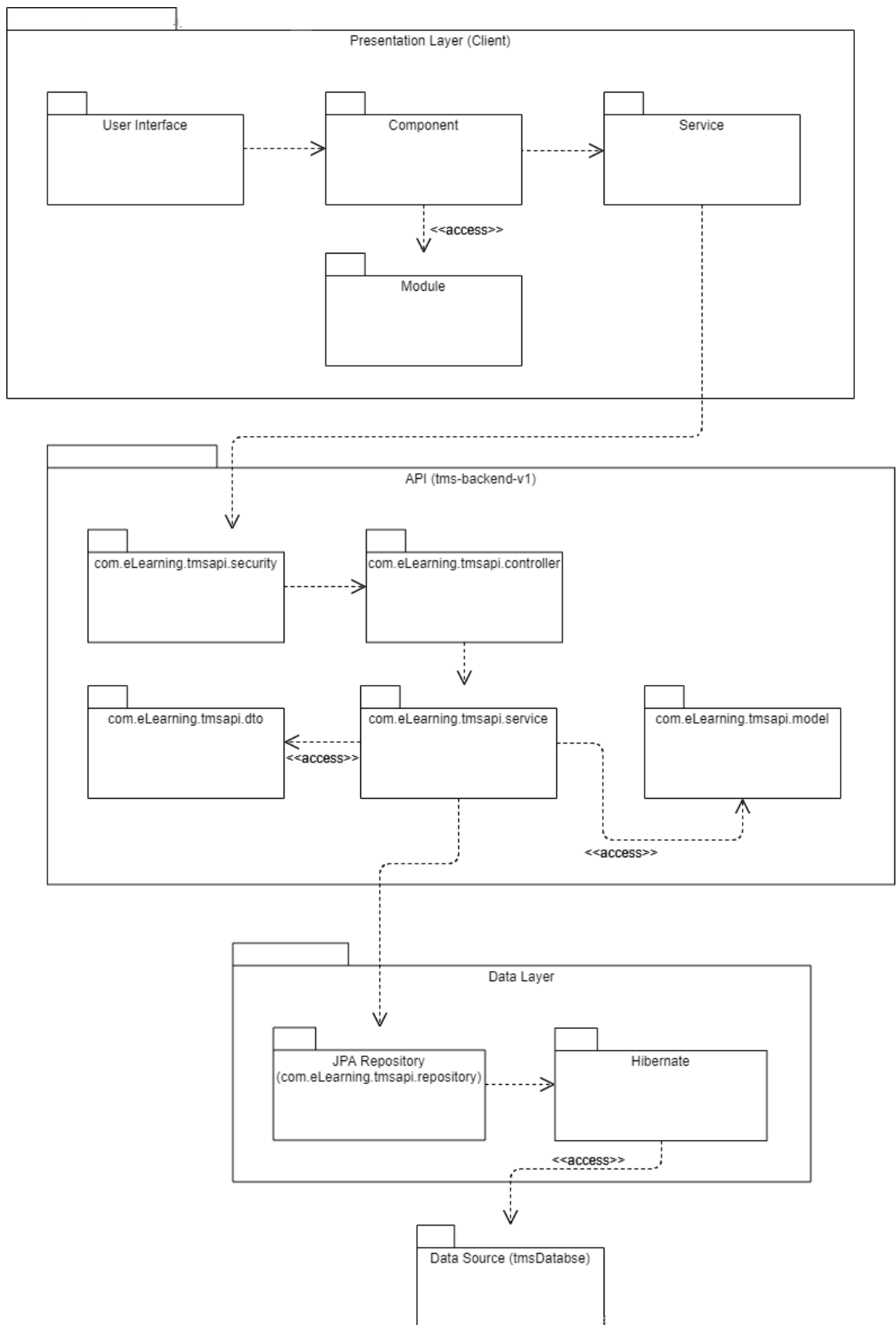


Figura 38 – Vista Lógica da plataforma

Com base na figura anterior (Figura 38), é explícito que a plataforma se encontra estruturada em quatro camadas, nomeadamente, Presentation Layer, API, Data Layer e Data Source.

Em seguida, são descritos os principais pacotes sob um ponto de vista arquitetural:

- Presentation Layer (camada de apresentação) – é responsável pela apresentação das páginas para o cliente e exibição dos dados. Esta camada é, portanto, constituída pelas páginas HTML(5) que recebem dados dos respetivos componentes que as suportam e os apresentam ao utilizador. Estes componentes são, também, responsáveis por atribuir os valores recebidos do servidor/API à página que será exibida ao utilizador e, em algumas situações, contém alguma lógica de negócio que deverá ser validada (também) no lado do cliente;
- API (camada de lógica de negócio) – contém todos os componentes responsáveis pela lógica de negócio. Esta camada é constituída por Security que recebe os pedidos externos (da camada Presentation) e valida a autorização do utilizador que efetuou o pedido. Caso o utilizador tenha autorização, o pedido é então enviado para Controller que o comunica ao respetivo serviço (Service). Service é responsável por encapsular a lógica de negócio da aplicação, controlando transações e coordenando respostas na implementação das suas operações. Para além disso, Service comunica essas transações à camada Data Layer. Quando este processo termina, a camada Service passa a informação para o respetivo Controller que devolve uma resposta à camada de apresentação. Relativamente aos DTOs, estes são objetos de transferência de dados onde o seu uso se restringe à ocultação dos detalhes de um objeto da restante aplicação, evitando assim que terceiros consigam modificar a aplicação;
- Data Layer (camada de dados) – é constituída por componentes de acesso à base de dados. Os repositórios presentes nesta camada são chamados por parte dos serviços (Service) e são responsáveis por gerir os dados presentes na camada de persistência de dados (Data Source).

Em síntese, com a implementação desta estrutura, é possível distinguir e distribuir as responsabilidades de forma correta, sendo mais fácil de entender, programar e testar. Por exemplo, se for necessário adicionar um novo caso de uso ao sistema ou estender as regras de negócio, a sua implementação será mais difícil se o processo ou a lógica de negócio se encontrarem “espalhados” pelo código sem qualquer tipo de estrutura.

5.5 Vista de Implantação

Esta secção apresenta a arquitetura do *software* sob um ponto de vista mais físico, com um alto nível de abstração. A Figura 39 mostra os nós da infraestrutura, as suas componentes e respetivas relações (protocolos de comunicação utilizados).

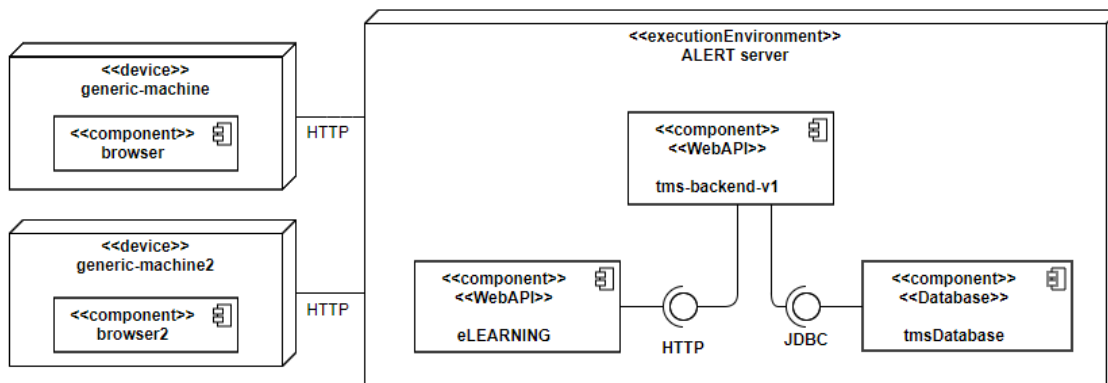


Figura 39 – Diagrama de implantação

Com base na figura anterior (Figura 39), verificam-se os principais nós da infraestrutura e respetiva alocação de componentes:

- ALERT server – nó da infraestrutura que é responsável pelo encapsulamento das componentes “tms-backend-v1”, “eLEARNING” e “tmsDatabase”. Este nó de infraestrutura condiciona a rede no acesso às suas componentes;
- generic-machine – nó da infraestrutura que é responsável pelo encapsulamento da componente *browser*. Esta componente consome os serviços de “tms-backend-v1” através do protocolo de comunicação HTTP;
- generic-machine2 – nó de infraestrutura que é responsável pelo encapsulamento da componente *browser2*. Esta componente consome os serviços de “tms-backend-v1” através do protocolo de comunicação HTTP.

6 Realização de Casos de Uso de maior Relevância

Neste capítulo, é descrito a solução desenvolvida. Assim, são apresentados os desenhos, bem como a implementação dos casos de uso do sistema apresentados na secção 4.2, considerando a estrutura e/ou componentes definidas na secção 5. Os casos de uso estão distribuídos por ator.

6.1 Autenticação e Autorização

A autenticação é o processo de verificação de uma identidade. Por sua vez, a autorização é o processo que ocorre após ser validada a autenticação e diz respeito aos privilégios que são concedidos a determinado utilizador de uma aplicação. Assim, na plataforma desenvolvida são comparadas as credenciais inseridas pelo utilizador com as credenciais que já existem no *Active Directory (AD)* da ALERT através do *Lightweight Directory Access Protocol (LDAP)*.

Sinteticamente, LDAP é um protocolo de *software* que permite a qualquer pessoa localizar dados sobre organizações, utilizadores ou outros tipos de recursos (ficheiros e dispositivos na rede – seja numa Internet pública ou corporativa). Para este projeto em particular, o LDAP é utilizado para confirmar as credenciais inseridas pelo utilizador quando pretende iniciar sessão na plataforma uma vez que o este diretório oferece um local central para autenticação – o que significa que ele armazena nomes e senhas de utilizadores. Contudo, qual é a vantagem em prover um ponto central para autenticação para a plataforma desenvolvida neste projeto?

A autenticação por LDAP retira responsabilidade ao sistema desenvolvido porque não necessita de atualizar – constantemente, os dados, sendo apenas requerido o registo do utilizador na base de dados com o mesmo *username* que o utilizado em autenticação por LDAP. Para além

disso, a utilização deste protocolo permite obter uma *thumbnail* para utilizar como foto de perfil do utilizador com sessão iniciada.

No desenvolvimento deste projeto, foram utilizadas bibliotecas disponibilizadas pelo Spring Security para realizar o processo de autenticação, não sendo necessário criar um *endpoint* para login. Assim, quando o pedido de autenticação retorna um código HTTP com status igual a 200 (OK), a resposta conterá um JSON Web Token (JWT), que é guardado localmente. Sempre que o utilizador desejar aceder a uma roa ou recurso protegido, deve ser enviado o JWT no cabeçalho do pedido (*Header*) utilizando o esquema *Bearer*.

Sempre que o servidor *back-end* receber um pedido com um JWT, o primeiro passo é validar o *token*. Isto é um processo que se divide em etapas distintas e, se alguma falhar, o pedido deverá retornar, por exemplo, um código HTTP com status igual a 403 (não autorizado). A Figura 40 evidencia esse conjunto de validações.

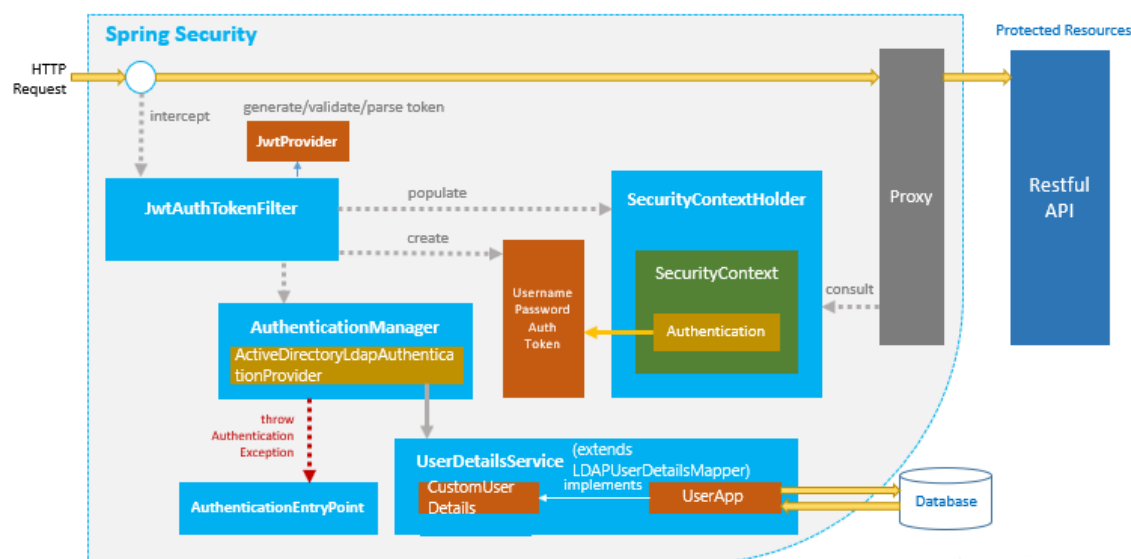


Figura 40 – Spring Security - Arquitetura no processo de Autenticação e Autorização JWT. Adaptado de (grokonez, 2020)

Quando a API (“tms-backend-v1”) recebe o pedido HTTP através de um browser ou *Web Service*, irá passar por uma cadeia de filtros para fins de autenticação e, posteriormente, autorização. A classe “*JwtAuthTokenFilter*” extrai as principais credencias do *token* (*username*, *password* e *role*), conforme a necessidade do pedido (autenticação ou autorização).

Particularmente para o caso de autenticação, sempre que um utilizador pretender iniciar sessão na plataforma, é criado um objeto do tipo “*UsernamePasswordAuthenticationToken*” que tem como objetivo obter as credenciais inseridas e combinar numa instância do “*AuthenticationManager*”. O Código 1 mostra um dos métodos da classe “*JWTAuthenticationFilter*”, do pacote *Security*, responsável pelo processo de autenticação.

```
@Override
public Authentication attemptAuthentication(HttpServletRequest req,
```

```

        HttpServletResponse res) throws AuthenticationException {
    try {
        UserApp creds = new ObjectMapper()
            .readValue(req.getInputStream(), UserApp.class);

        return authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(
                creds.getUsername(),
                creds.getPassword(),
                new ArrayList<>())
        );
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

```

Código 1 – Método de validações e recolha de dados do *token* de acesso

O “AuthenticationManager” utiliza o “ActiveDirectoryLdapAuthenticationProvider” para concretizar a ligação ao LDAP da ALERT e, conseqüentemente, validar a instância do tipo “UsernamePasswordAuthentication” retornada no método apresentado. Esta instância permite autenticar o utilizador ao comparar as credenciais submetidas. Caso a instância não seja considerada como válida, é retornada uma exceção do tipo “AuthenticationException”. Contudo, se a instância for válida, é utilizado um dos métodos da classe “UserDetailsService” para aceder aos dados do utilizador obtidos do LDAP.

Em seguida, de modo a dar continuidade ao processo de autenticação, é instanciado um objeto do tipo “CustomUserDetails”, bem como um *array* do tipo “byte[]”. A função deste segundo objeto é o armazenamento da *thumbnail* do utilizador que se autenticou por intermédio da interface “LdapUserDetails”. Isto acontece porque a interface não foi desenvolvida com esse propósito, sendo necessário o acréscimo deste passo.

Ainda sobre a classe “CustomUserDetails”, esta foi modificada para guardar os dados recebidos do LDAP e a *thumbnail* do utilizador. Não menos obstatante, é ainda necessário obter a *role* do utilizador que se encontra em fase de autenticação. Para isso, é realizado um pedido à base de dados com o *username* em questão, que deve retornar uma instância do tipo “UserApp”. Seguidamente, é efetuado um ciclo *for* para separar as *roles* num *array* (no início do projeto, foi concluído que um utilizador tinha mais do que uma *role*, daí este ciclo para separar as mesmas num *array* do tipo *String* para armazenar no *token*, no entanto, mais tarde, o mesmo foi atualizado para que um utilizador tenha apenas uma *role*). Este ciclo manteve-se uma vez que não influenciava a criação da *role* para o *token*.

Por fim, de modo a retornar um objeto do tipo “CustomUserDetails”, que já contém os campos *username* e *password*, foram editados os seguintes campos: “Authorities” devido à *role*, o tempo de expiração e a *thumbnail*. O Código 2 explica os passos anteriormente descritos.

```

@Override
public CustomUserDetails mapUserFromContext(DirContextOperations ctx,
String username, Collection<? extends GrantedAuthority> authorities) {

```

```

    LdapUserDetailsImpl details = (LdapUserDetailsImpl)
super.mapUserFromContext(ctx, username, authorities);

    CustomUserDetails customUserDetails = new CustomUserDetails();
    byte[] photo = null;

    try {
        if(ctx.getAttributes().get("thumbnailPhoto") != null) {
            photo =
                (byte[])ctx.getAttributes().get("thumbnailPhoto").get()
        }
    } catch (NamingException e) {
        e.printStackTrace();
    }

    UserApp applicationUser =
this.userAppRepository.findByUsernameIgnoreCase(username);
    if (applicationUser == null) {
        throw new UsernameNotFoundException(username);
    }
    customUserDetails.setDn(details.getDn());
    customUserDetails.setPassword(details.getPassword());
    customUserDetails.setUsername(details.getUsername());

    // get roles
    String[] rolesAux = new String[applicationUser.getRoles().size()];
    for(int i = 0; i < applicationUser.getRoles().size(); i++) {
        rolesAux[i] = applicationUser.getRoles().get(i).getName();
    }

    //AuthorityUtils.createAuthorityList(applicationUser.getRoles())
    customUserDetails.setAuthorities(AuthorityUtils.createAuthorityList(rolesAux));
    customUserDetails.setGraceLoginsRemaining(details.getGraceLoginsRemaining());
    customUserDetails.setTimeBeforeExpiration(details.getTimeBeforeExpiration());
    customUserDetails.setThumbnailPhoto(photo);

    return customUserDetails;
}

```

Código 2 - Criação do CustomUserDetails com dados obtidos do LDAP e da base de dados

Neste momento, é iniciada a etapa onde se cria o *token* com os dados retornados no método apresentado acima. Primeiro, é criado um objeto do tipo “GrantedAuthority” que itera sobre o *array* de *String*. Em seguida, é executada uma condição que permite compreender se a autenticação é efetuada por LDAP ou base de dados (a partir do objeto retornado no Código 2). Se uma autenticação for efetuada por base de dados é porque o *username* não é o mesmo que o obtido por LDAP.

Independentemente do caminho que o programa seguir na condição, será sempre criado um *token* onde um dos seus campos identifica se a autenticação é ou não por LDAP. O *token* é retornado no cabeçalho da resposta HTTP em conjunto com a *thumbnail*. Apesar de não ser identificado como um passo significativo, foi decidido efetuar a codificação da *thumbnail* no

lado do servidor e não no cliente uma vez que a API deve ser genérica. Esta codificação é em Base64.

O Código 3 apresenta as operações descritas nos dois parágrafos anterior.

```
@Override
protected void successfulAuthentication(HttpServletRequest req,

    HttpServletResponse res, FilterChain chain, Authentication auth) throws
IOException, ServletException {

    // saber role do user
    String [] arrayRoles = new String [auth.getAuthorities().size()];
    Iterator<GrantedAuthority> iteratorRoles = (Iterator<GrantedAuthority>)
auth.getAuthorities().iterator();
    int index = 0;
    while(iteratorRoles.hasNext()) {
        arrayRoles[index] = iteratorRoles.next().getAuthority().toString();
        index++;
    }

    if(auth.getPrincipal() instanceof CustomUserDetails) { // autenticação por
LDAP

        // create user
        CustomUserDetails userDetails = (CustomUserDetails)
auth.getPrincipal();

        String token = JWT.create()
            .withClaim("ldap", true)
            .withSubject(userDetails.getUsername())
            .withArrayClaim("authorities", arrayRoles)
            .withExpiresAt(new Date(System.currentTimeMillis() +
EXPIRATION_TIME)).sign(HMAC512(SECRET.getBytes()));
        res.addHeader(HEADER_STRING, TOKEN_PREFIX + token);

        // photo
        if(userDetails.getThumbnailPhoto() != null) {
            PrintWriter writer = res.getWriter();
            writer.append(DatatypeConverter.printBase64Binary(userDetail
s.getThumbnailPhoto()));
            writer.close();
        }

    } else { // autenticação por DB

        // criar token
        String token = JWT.create()
            .withClaim("ldap", false)
            .withSubject(((User)
auth.getPrincipal()).getUsername())
            .withArrayClaim("authorities", arrayRoles)
            .withExpiresAt(new Date(System.currentTimeMillis()
+ EXPIRATION_TIME)).sign(HMAC512(SECRET.getBytes()));
        res.addHeader(HEADER_STRING, TOKEN_PREFIX + token);
    }
}
```

Código 3 - Criação e adição do *token* e da *thumbnail* ao cabeçalho da resposta

Como implementamos o filtro responsável pela autenticação de utilizadores, agora é necessário implementar o filtro responsável pela autorização do utilizador. Esse filtro é criado como uma nova classe, chamada “JWTAuthorizationFilter” no pacote Security.

A classe “JWTAuthorizationFilter” estende o “BasicAuthenticationFilter” de modo a que o Spring possibilite uma implementação personalizada do programador com os dados que o mesmo precisa trabalhar (referidos anteriormente). A parte mais importante do novo filtro implementado é o método privado apresentado no Código 4 (“getAuthentication”). Esse método lê o JWT presente no cabeçalho de autorização e, em seguida, utiliza o JWT para validar o *token*. Se a execução não lançar exceções, é retornada uma instância de “UsernamePasswordAuthenticationToken”, que permite prosseguir com o pedido.

```
private UsernamePasswordAuthenticationToken
getAuthentication(HttpServletRequest request) {
    String token = request.getHeader(HEADER_STRING);

    if(token != null) {

        // parse the token
        String user =
        JWT.require(Algorithm.HMAC512(SECRET.getBytes())) //
        Algorithm.HMAC512
            .build()
            .verify(token.replace(TOKEN_PREFIX, ""))
            .getSubject();

        List<String> claims =
        JWT.require(Algorithm.HMAC512(SECRET.getBytes()))
            .build()
            .verify(token.replace(TOKEN_PREFIX, ""))
            .getClaim("authorities").asList(String.class);

        if(user != null) {

            return new UsernamePasswordAuthenticationToken(user,
            null,
            claims.stream().map(SimpleGrantedAuthority::new).collect(Collectors
            .toList()));
        }
        return null;
    }
    return null;
}
```

Código 4 – Método de validação e recolha de dados do *token* de acesso

A instância retornada pelo método acima vai ser utilizada para criar e armazenar o objeto “Authentication” no objeto “SecurityContextHolder”. Spring Security utiliza o “SecurityContextHolder” para guardar os detalhes do contexto de segurança da plataforma em relação ao utilizador que realizou o pedido. Assim, o “SecurityContextHolder” é utilizado nos filtros de autorização para validar se o utilizador tem permissão para aceder ao *endpoint* requerido. O Código 5 apresenta o método “doFilterInternal” responsável pela definição do “SecurityContextHolder”. Este método recebe o objeto gerado Código 4.

```

@Override
Protected void doFilterInternal(HttpServletRequest req,
    HttpServletResponse res, FilterChain chain) throws IOException,
    ServletException {

    String header = req.getHeader(HEADER_STRING);

    if(header == null || !header.startsWith(TOKEN_PREFIX)) {

        // throw new JwtTokenMissingException("No JWT token found in
        request headers");
        chain.doFilter(req, res);
        Syetm.out.println("No JWT token found in request headers");
        return;
    }

    UsernamePasswordAuthenticationToken authentication =
    getAuthentication(req);

    SecurityContextHolder.getContext().setAuthentication(authentication
    );
    chain.doFilter(req, res);
}

```

Código 5 - Processo de autorização

Agora que temos os dois filtros de segurança criados corretamente, precisamos configurar os mesmos na cadeia de filtros do Spring Security. Para isso, foi necessário criar uma classe chamada “WebSecurity” no pacote Security.

Essa classe é anotada com o “@EnableWebSecurity” e estende o “WebSecurityConfigurerAdapter” para aproveitar a configuração da Spring Security. Isto permite que o programador ajuste a estrutura de modo a responder às suas necessidades, definindo três métodos. Contudo, apenas será mostrado o método apresentado no Código 6.

```

@Override
Protected void configure(HttpSecurity http) throws Exception {

    http.cors().and.csrf().disable().authorizeRequests()
        .antMatchers(HttpMethod.OPTIONS, “/**”).permitAll()
        // USERS
        .antMatchers(HttpMethod.GET,
    SecurityEndpoints.API_GET_ALL_USERS).hasAnyAuthority(“ROLE_ADMIN”,
    “ROLE_LOCAL_COORDINATOR”, “ROLE_TRAINING_MANAGER”)
        .antMatchers(HttpMethod.GET,
    SecurityEndpoints.API_GET_USERS_SORTED).permitAll()
        .and()
        .addFilter(new
    JWTAuthenticationFilter(authenticationManager(), propertiesValues))
        .addFilter(new
    JWTAuthorizationFilter(authenticationManager()))
        // this disables session creation on Spring Security

        .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.ST
    ATELESS);
}

```

```
}
```

Código 6 - Configurações de segurança de acesso a *endpoints* da API

Este método permite distinguir que recursos devem ser públicos e protegidos. Através dele, é possível configurar o suporte ao CORS (*Cross-Origin Resource Sharing*) por meio do “`http.cors()`” e é adicionado um filtro de segurança personalizado na cadeia de filtros do Spring Security. Acima, são mostrados dois *endpoints* da API:

- `API_GET_ALL_USERS` – um método HTTP do tipo GET que só pode ser acessado pelos 3 grupos de utilizadores: *Administrator*, *Local Coordinator* e *Training Manager*;
- `API_GET_USERS_SORTED` – um método HTTP do tipo GET que pode ser acessado por todos os grupos de utilizadores.

6.2 Visualizar Lista de Utilizadores da Plataforma

Como ilustrado na secção 4.2, um dos principais casos de uso deste projeto é a gestão de utilizadores da plataforma, que pode ser realizada por *Administrators* e *Local Coordinators*. Na Figura 41 é apresentado o fluxo de interações entre as diferentes componentes lógicas representadas na secção 5.4.

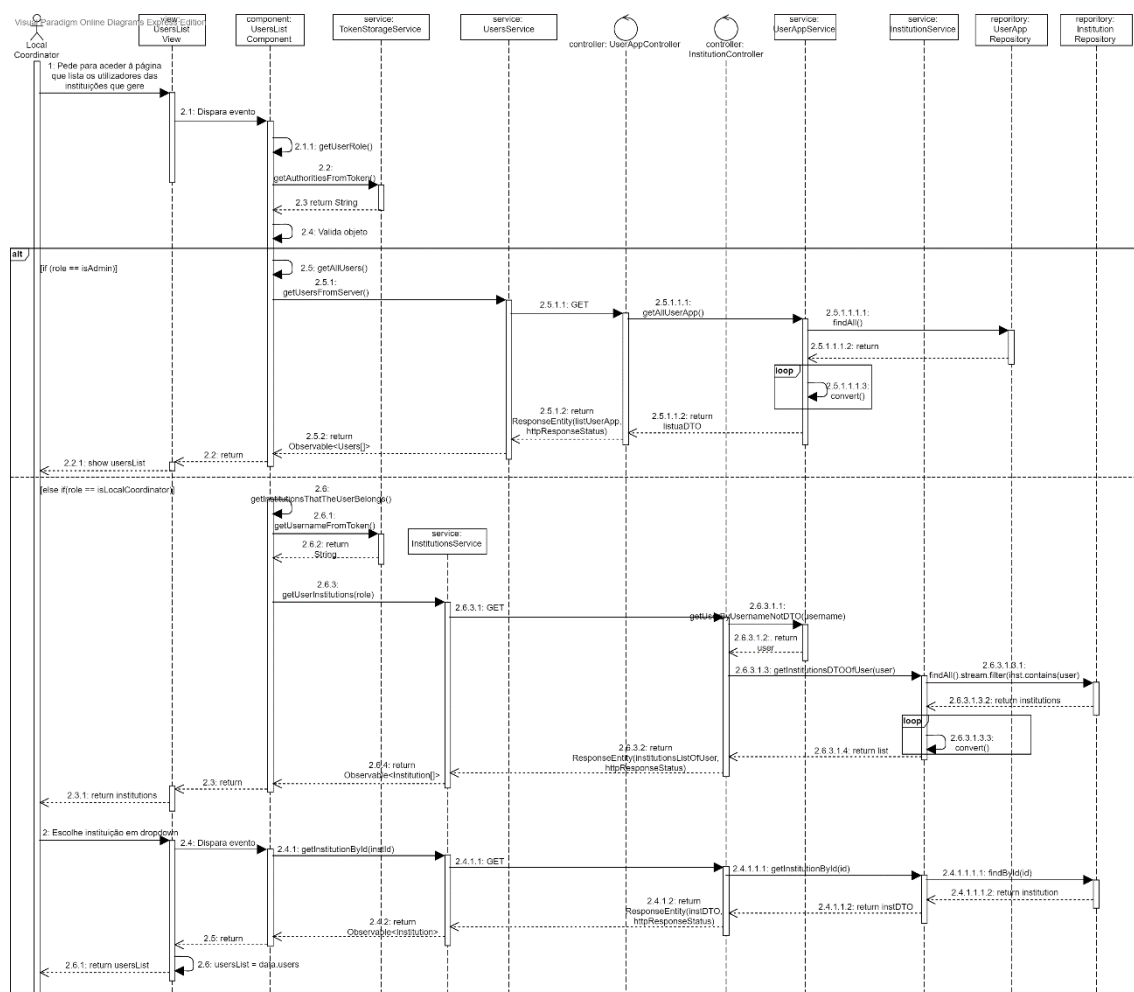


Figura 41 - Diagrama de sequência com o fluxo de execução do UC7

Iniciando o processo na camada de apresentação, quando o utilizador solicita a página que permite visualizar todos os utilizadores da plataforma, é espoletado um evento na componente de lista de utilizadores (“UsersListComponent”) da camada de apresentação. Em seguida, de modo a identificar o tipo de informação e funcionalidades que devem ser exibidos para o utilizador autenticado, a componente invoca o método “getUserRole()”, que, consequentemente, chama o serviço “TokenStorageService” para retornar a *role*.

O “TokenStorageService” é um serviço genérico que trata toda a lógica de negócio relativamente ao que se encontra guardado na *Local Storage* do *browser*, nomeadamente, as credenciais recebidas do *token* após autenticação.

Assim, para este componente podem acontecer os seguintes cenários:

- Cenário 1 - a permissão do utilizador autenticado é “ROLE_ADMIN”;
- Cenário 2 - a permissão do utilizador autenticado é “ROLE_LOCAL_COORDINATOR”.

A Figura 41 ilustra os fluxos de interação para ambos os cenários, no entanto, a explicação para este caso de uso incidirá naquilo que acontece no cenário 2. O Código 7 mostra o método “getUserRole()” que tem como objetivo escolher um dos cenários após identificação da *role*.

```
// GET USER ROLE
getUserRole(): void {
    let role = this.tokenStorageService.getAuthoritiesFromToken();

    // controls flags for UI components to display
    this.isAdmin = role.includes(Role.Admin);
    this.isLocalCoordinator = role.includes(Role.Local_Coordinator);

    if(this.isAdmin == true) {
        this.getAllUsers();
    } else if(this.isLocalCoordinator == true) {
        this.getInstitutionsThatTheUserBelongs();
    }
}
```

Código 7 – Implementação do método que identifica a *role* do utilizador e encaminha para um dos dois cenários possíveis

Após avançar para o método “getInstitutionsThatTheUserBelongs()”, é efetuado um pedido HTTP do tipo GET à API. Este é direcionado para o controlador de instituições (“InstitutionController”), onde são realizadas algumas validações e, caso nenhuma exceção tenha sido espoletada, encaminha o pedido para o respetivo serviço onde se encontra o método apresentado no Código 8.

```
@Override
public List<InstitutionDTO> getInstitutionsDTOfUser(UserApp user) {
    // get institutions
    List<Institution> institutions =
    this.institutionRepository.findAll().stream().filter(inst ->
    inst.getUsers().contains(user)).collect(Collectors.toList());
    if(institutions == null)
        return null;

    List<InstitutionDTO> finalInstitutions = new ArrayList<>();
    for(Institution inst : institutions) {
        finalInstitutions.add(inst.convertToDTO(inst));
    }
    return finalInstitutions;
}
```

Código 8 - Implementação do método para obter as instituições que pertencem a um utilizador com permissão *Local Coordinator*

Este método chama o repositório de instituições responsável por retornar uma lista com todas as instituições armazenadas na base de dados. Quando isso acontece, ou seja, quando a *query* for concluída, é executada uma filtragem imediata no objeto retornado. Isso acontece porque, como referido em outras secções, o *Local Coordinator* só pode visualizar/gerir os utilizadores das instituições a que pertence.

Neste momento a API retorna o código “HTTP OK” que indica que a requisição foi concluída e, conseqüentemente, envia a lista de instituições na resposta. O serviço de instituições da

camada de apresentação recebe e reencaminha o pedido para o respetivo componente. Aqui, se o objeto retornado não estiver vazio, são mostrados os utilizadores da primeira instituição do *array* na página solicitada. Contudo, se o objetivo do utilizador que solicitou a página é visualizar os de outra instituição, o mesmo deverá selecionar a *dropdown* que lista as instituições a que o mesmo pertence. Se isto acontecer, é espoletado um novo evento da view para o componente e do componente para o serviço de instituições da camada de apresentação. Neste momento, é efetuado um novo pedido à API que tem como objetivo retornar apenas os utilizadores da instituição selecionada. O Código 9 mostra o método do serviço de instituições da API responsável por tratar desses dados.

```
@Override
public InstitutionDTO getInstitutionByIdDTO(long id) {

    Institution institution =
this.institutionRepository.findById(id).orElse(null);
    if(institution == null)
        return null;
    // convert to DTO
    return institution.convertToDTO(institution);
}
```

Código 9 - Implementação do método que permite obter uma instituição pelo seu id

Por fim, a lista de utilizadores enviada na resposta HTTP é encaminhada para o componente de instituições presente na camada de apresentação que trata os dados e retorna à *view*. Esta é ilustrada na Figura 42.

Username	First Name	E-mail	Role
outro-something-trainer	Outro Trainer	catia.sousa@alert-online.com	ROLE_TRAINER
Teste	LC	catia.sousa@alert-online.com	ROLE_LOCAL_COORDINATOR
JoaoFiaes	joao	joaofiaes@catia.sousa@alert-online.com	ROLE_TRAINEE

Figura 42 – Página com código HTML para o UC7

6.3 Criar Instituição

Para que o ALERT® TMS funcione corretamente, todos os dados relativos à formação (por exemplo, *training projects*, *training plans*, sessões) devem ser sempre associados a uma instituição. Na Figura 43 é apresentado o fluxo de interações entre as diferentes componentes lógicas representadas na secção 5.4.

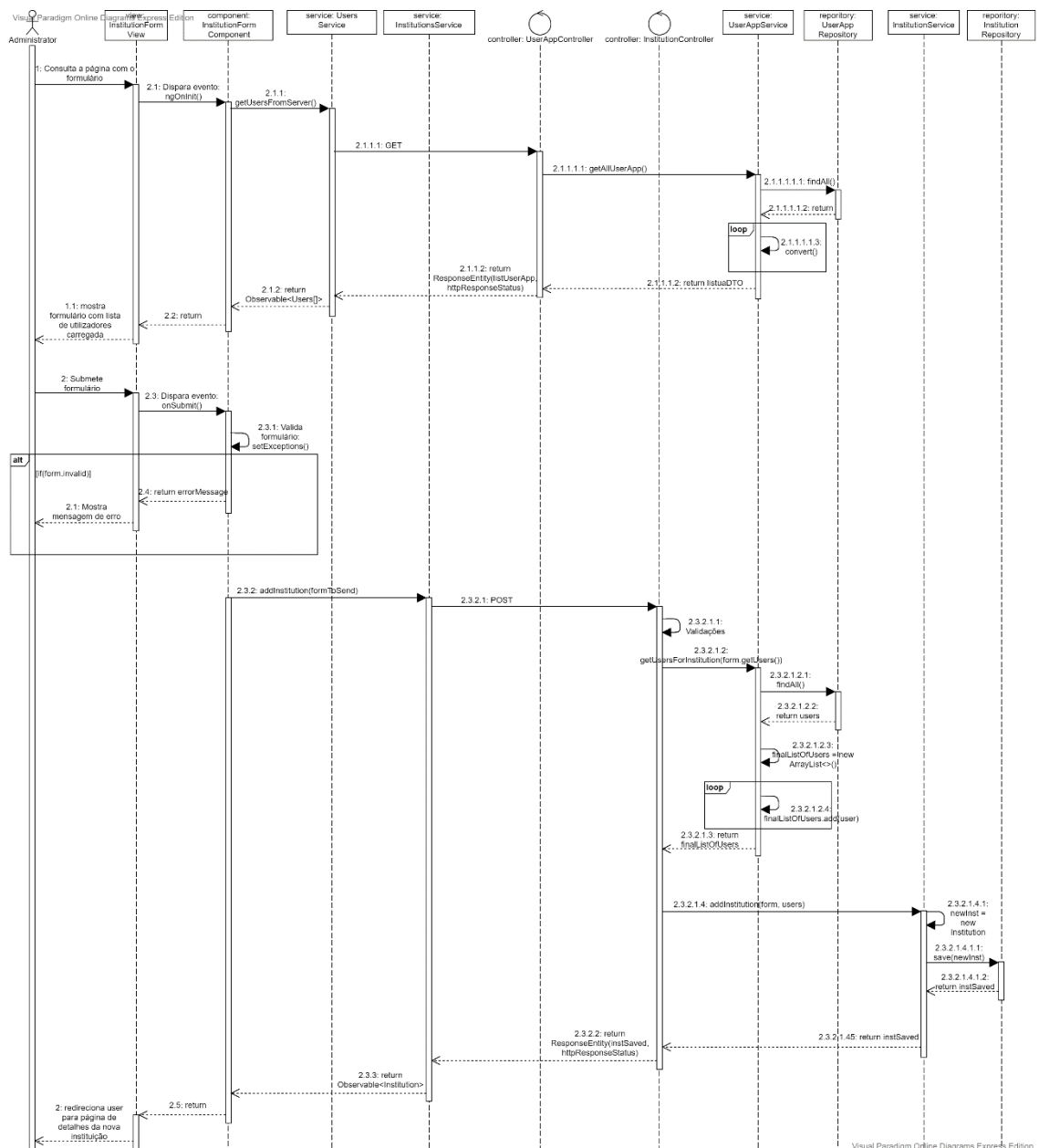


Figura 43 – Diagrama de sequência com o fluxo de execução do UC16

Assim que o utilizador pede acesso à vista que permite adicionar uma instituição à aplicação, é disparado um evento que chama o componente da camada de apresentação responsável por suportar a exibição da página. O evento espoletado não pode mostrar o formulário sem

primeiro realizar um pedido HTTP do tipo GET à API. Este pedido tem como objetivo retornar uma lista que contenha todos os utilizadores da plataforma. Na realidade, quando um *Administrator* ou *Local Coordinator* adiciona uma instituição à aplicação, apesar da não obrigatoriedade, o mesmo pode associar utilizadores. Para isso, o pedido cai sobre o controlador dos utilizadores (“UserAppController”), que, posteriormente, o reencaminha para o serviço responsável (caso nenhuma exceção tenha sido espoletada). O Código 10 apresenta a implementação do método desse serviço que tem como objetivo obter uma lista de todos os utilizadores guardados na base de dados através da execução da *query* que permite seleccionar todas as linhas da tabela “UserApp”.

```
@Override
public List<UserAppDTO> getAllUserApp() {

    // buscar todos os UserApp à database com a ajuda do JPA
    List<UserApp> lista = this.userAppRepository.findAll();

    if(lista == null)
        return null;
    // retornar em DTO
    List<UserAppDTO> listuaDTO = new ArrayList<>();
    for(UserApp u : lista)
        listuaDTO.add(u.convert(u));

    return listuaDTO;
}
```

Código 10 – Implementação do método que permite obter os utilizadores da plataforma

Antes de retornar o objeto ao respetivo controlador, os objetos da lista devem ser convertidos e, conseqüentemente, adicionados a uma segunda lista do tipo “UserAppDTO”. O processo termina quando a resposta chega ao serviço das instituições, que trata a mesma e retorna ao componente de criação da instituição. Em seguida, os dados são carregados e, finalmente, é possível mostrar o formulário.

Após preenchimento dos campos obrigatórios do formulário, quando o utilizador carrega no botão de submissão, a *view* dispara um evento sobre o respetivo componente. O método apresentado no Código 11 é responsável por validar o formulário (verificar os campos obrigatórios), preencher a instância do formulário que deve ser enviado no pedido HTTP e, posteriormente, chamar o serviço de instituições da camada de apresentação.

```
onSubmit(): void {
    this.submitted = true;

    // stop here if form is invalid
    if (this.registerForm.get('name').value == "" ||
        this.registerForm.get('institution_type').value == ""
        || this.registerForm.get('email').value == "") {
        return;
    }

    this.setExceptions();

    // call service
    this.instituitonsService.addInstitution(this.formToSend).subscribe(
```

```

        (data) => {
            alert('You added a new institution to your application
            succesfully! Now, you will be returned to your institutions list
            page.');
```

```

            this.router.navigate(['institutions']);
        }, error => {
            // set data
            this.registerForm.get('createdAt').setValue(null);
            this.formToSend.get('createdAt').setValue(null);
            this.formToSend.get('users').setValue([]);
            this.usersFromFormToSend = [];
            this.errorMessageBool = true;
            this.errorMessageFromServer = error.error;
            return;
        }
    });
}

```

Código 11 – Implementação do método que permite validar o formulário e encaminhar o pedido para o respetivo serviço na camada de apresentação

Neste momento, o pedido HTTP do tipo POST foi efetuado à API. Um dos primeiros pedidos do controlador (“InstitutionController”), é obter uma lista de utilizadores do tipo “UserApp” que devem ser adicionados à nova instituição por intermédio do serviço e repositório de utilizadores. Se não ocorrer nenhum erro, o pedido segue o fluxo normal, onde volta ao controlador das instituições. Aqui, é chamado o serviço com o método responsável por guardar a nova instituição. Assim, é instanciado um novo objeto de instituições com o nome, o tipo de instituição (“INTERNAL”, “CLIENT” ou “PARTNERS”), correio eletrónico e um estado (valor igual a “ACTIVE”).

Por fim, a API retorna o código “HTTP CREATED” que indica que a requisição foi bem-sucedida, bem como o objeto da nova instituição. Posteriormente, na camada de apresentação, o utilizador recebe uma mensagem onde é informado do sucesso da operação. A Figura 44 ilustra a página disponível na plataforma desenvolvida, responsável por criar uma instituição.

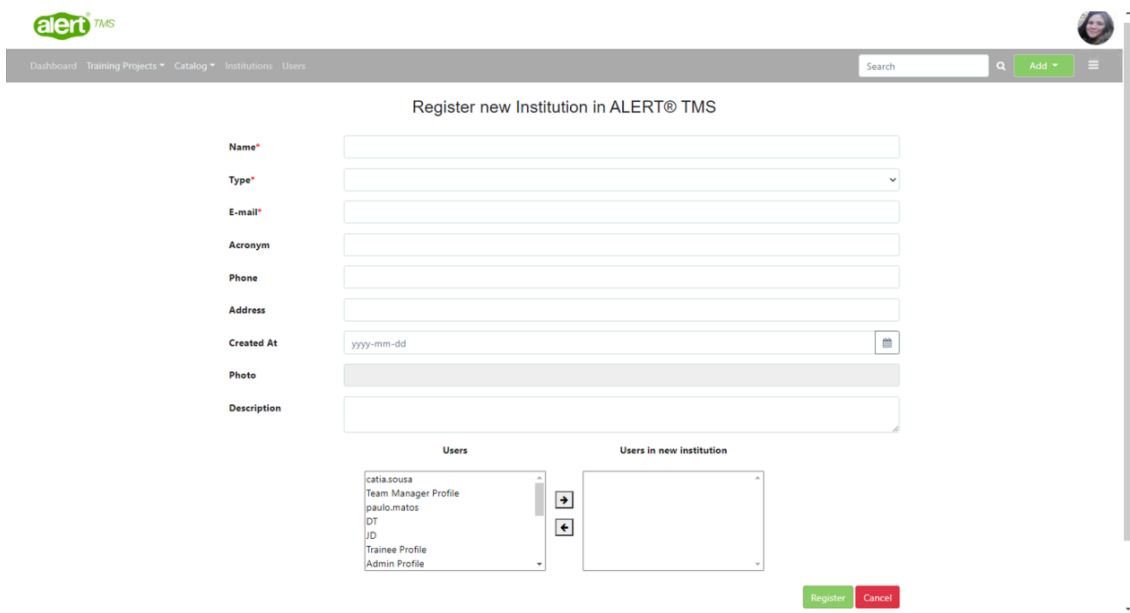


Figura 44 – Página com código HTML para o UC16

6.4 Criar Curso

Antes de detalhar a interação entre as componentes lógicas da plataforma no que respeita à criação de um curso, primeiro é necessário compreender os estados que o mesmo pode assumir, tendo por base o estudo realizado nas secções 2.5.4 e 2.6 sobre o Adobe Captive Prime. Na Figura 45 são exibidas as interações entre os diferentes estados para um curso.

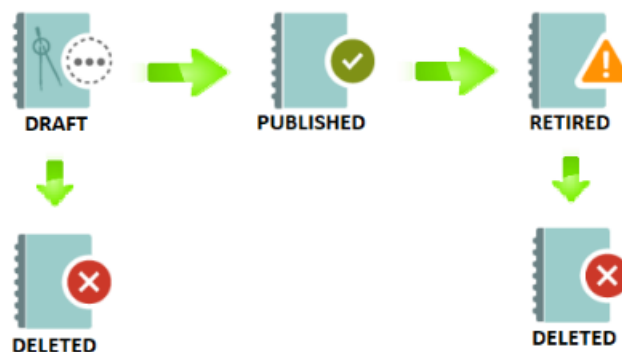


Figura 45 - Ciclo de vida típico de um curso. Adaptado de (Courses, 2020)

Assim, um ciclo de vida típico de um curso é o seguinte:

- DRAFT – quando um utilizador autenticado, com permissão, conclui a criação de um curso e o guarda. Neste caso em particular, o curso ainda não se encontra disponível para *Trainees* e *Team Managers*;

- PUBLISHED – quando um utilizador autenticado, com permissão, conclui a criação de um curso e publica. Neste caso em particular, o curso ficará, no imediato, disponível para *Trainees* e *Team Managers*;
- RETIRED – após publicação de um curso, se o utilizador assim entender, pode mudar o estado de PUBLISHED para RETIRED, caso não pretenda que o mesmo apareça no catálogo de cursos;
- DELETED – sempre que um curso apresentar o estado DRAFT ou RETIRED, é quando o mesmo pode ser completamente removido da aplicação ALERT® TMS.

Após compreensão do ciclo de vida de um curso, na Figura 46 são apresentados os fluxos de interações entre as diferentes componentes lógicas representadas na secção 5.4, de modo a que seja possível criar um curso.

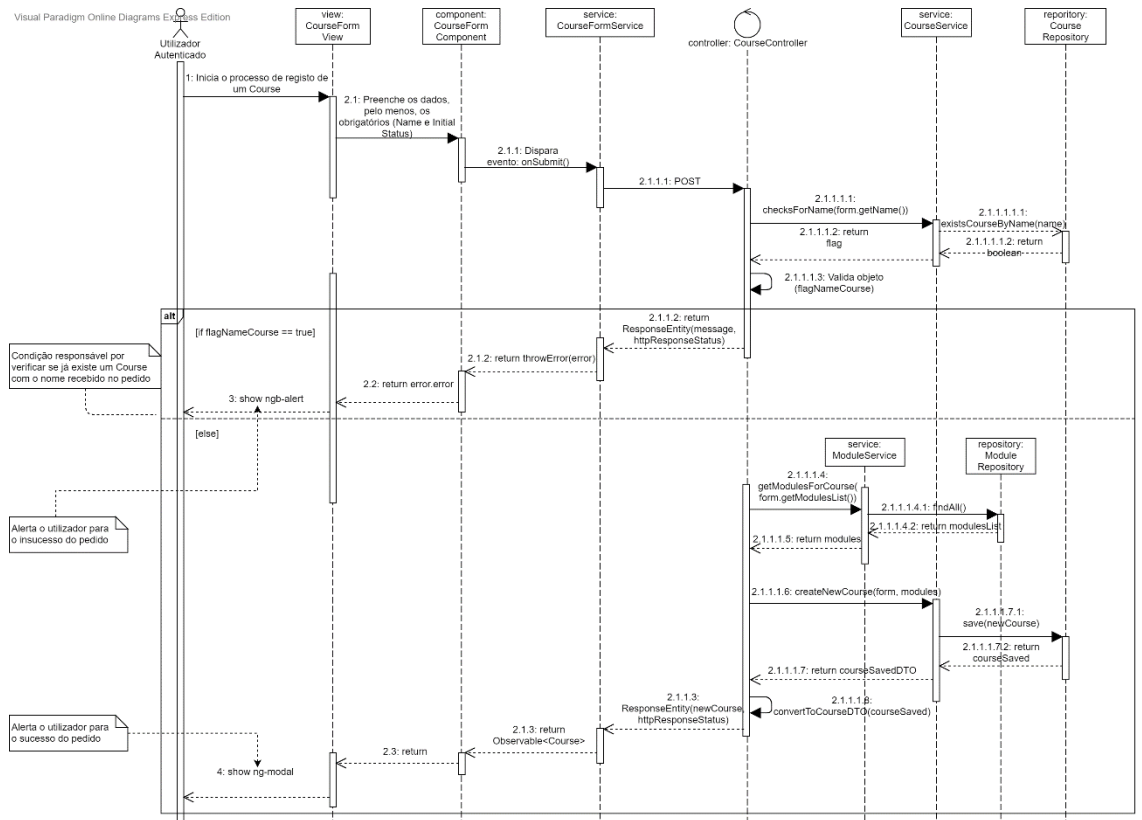


Figura 46 - Diagrama de sequência com o fluxo de execução para o UC3

Um curso pode ser adicionado à aplicação após preenchimento obrigatório dos campos que se encontram na página HTML solicitada pelo utilizador autenticado. Após submissão do formulário, é espoletado o envio de um pedido HTTP do tipo POST à API. Um dos primeiros pedidos do controlador (ControllerCourse), é obter uma lista de cursos através de um pedido à base de dados (camada Data Source) intermediado pelo respetivo serviço (CourseService) e repositório (CourseRepository, presente na camada Data Layer). O Código 12 apresenta o método do CourseService responsável por este pedido.

```

// CHECKS IF NAME EXISTS IN DB
@Override
public Boolean checksForName(String name) {

    Boolean flag = this.courseRepository.existsCourseByName(name);
    return flag;
}

```

Código 12 - Implementação do método responsável por verificar se já existe um curso com o mesmo nome

Após verificação do valor retornado do serviço de cursos, o controlador verifica se o mesmo é igual a “false”. Em caso verdadeiro, o pedido segue o fluxo normal, onde agora é necessário chamar o serviço de módulos para obter todos os dados dos módulos que devem ser adicionados a este curso, uma vez que, até ao momento, a lista de módulos no pedido era do tipo “ModuleDTO”. Após término deste passo, o controlador de cursos volta a chamar o serviço com o método responsável por guardar o novo curso. Assim, é instanciado um novo objeto de cursos com o nome, o estado inicial (“DRAFT” ou “PUBLISHED”), uma breve descrição, uma descrição detalhada e os módulos.

Por fim, a API retorna o código “HTTP CREATED” que indica que a requisição foi bem-sucedida e o novo curso criado, sendo o utilizador informado do sucesso da operação. O Código 13 apresenta o método do controlador de cursos que recebe o pedido e envia a resposta para o cliente (camada de apresentação).

```

// CREATE NEW COURSE
@PostMapping("/create")
public ResponseEntity<?> createCourse(@RequestBody CourseDTO form) {
    try {
        // empty form
        if(form == null) {
            return new ResponseEntity<>("Your form is invalid. Please
            ensure that you " + "provide the main fields (name, initial
            status and enrollment type).", HttpStatus.CONFLICT);
        }
        // check if name already exists
        Boolean flagNameCourse =
this.courseService.checksForName(form.getName());
        if(flagNameCourse == true)
            return new ResponseEntity<>("The name of the course already
            exists. Please, choose " + "another.", HttpStatus.CONFLICT);
        // incorrect initial status
        if(form.getStatus().name() != "PUBLISHED") {
            if(form.getStatus().name() != "DRAFT") {
                return new ResponseEntity<>("Incorrect initial
status.", HttpStatus.CONFLICT);
            }
        }

        // create new course
        List<Module> modules = new ArrayList<>();
        CourseDTO newCourse = new CourseDTO();
        if(form.getModulesList() != null) {
            // GET proper modules attributes

```

```

        modules =
this.moduleService.getModulesForCourse(form.getModulesList());
        // add course
        newCourse = this.courseService.createNewCourse(form,
modules);
    } else {
        // add module with empty activities list
        newCourse = this.courseService.createNewCourse(form,
modules);
    }
    return new ResponseEntity<>(newCourse, HttpStatus.CREATED);
} catch(RuntimeException e) {
    return new ResponseEntity<>("Failed to create new Course.",
HttpStatus.BAD_REQUEST);
}
}
}

```

Código 13 - Método do controlador de cursos que recebe o pedido de adição de curso

Na Figura 47 é ilustrada a página HTML com o formulário que espoleta os fluxos de interações descritos nesta secção.

Figura 47 – Página com código HTML para o UC3

6.5 Visualizar Cursos

Nesta secção é apresentado o processo de visualização dos cursos em duas perspetivas: o utilizador autenticado é *Trainee* ou *Team Manager*, ou pertence a um dos outros quatro grupos de utilizadores. Na Figura 48 são apresentados os fluxos de interações entre as diferentes componentes lógicas representadas na secção 5.4.

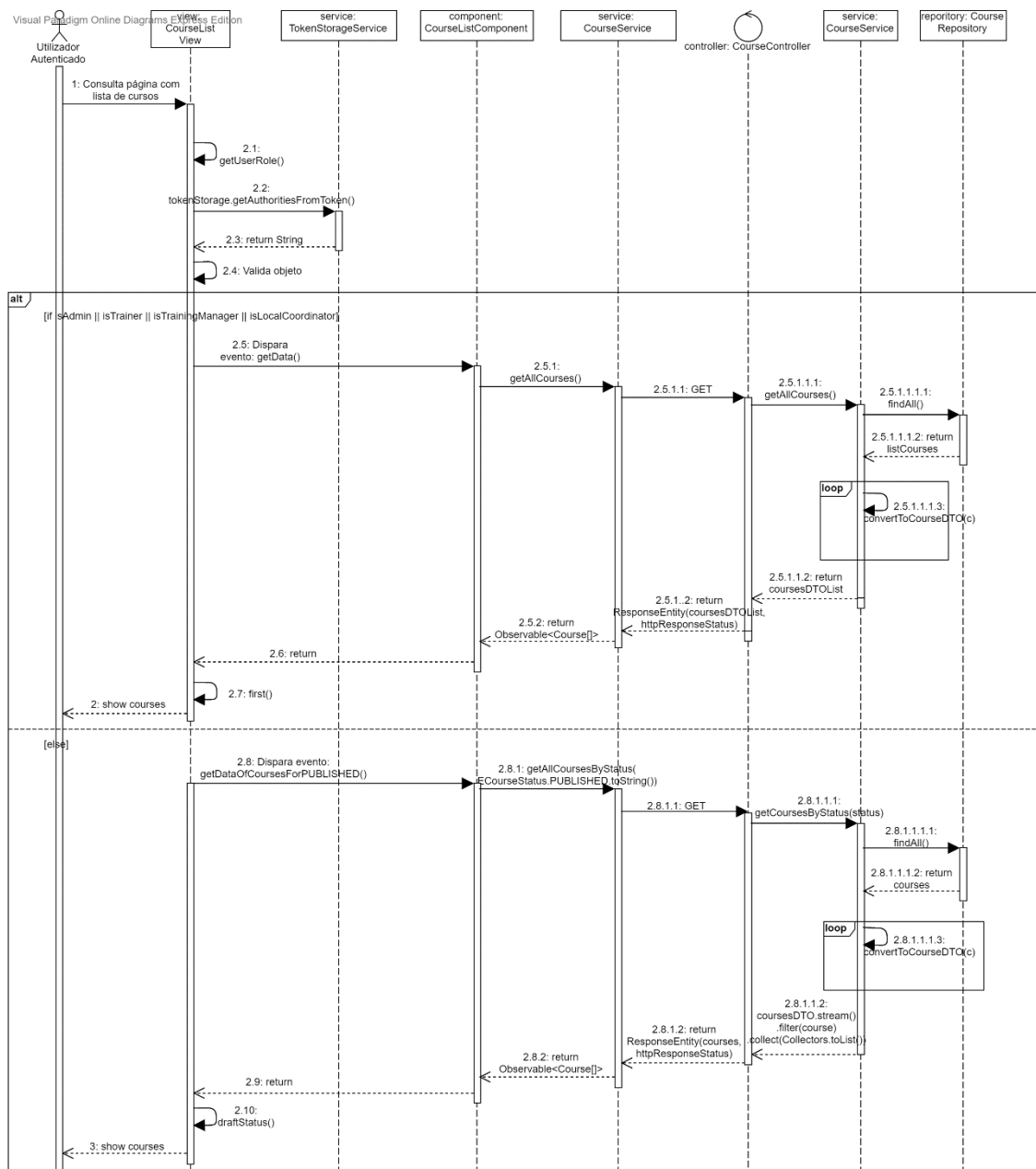


Figura 48 – Diagrama de sequência com o fluxo de execução para visualizar cursos

Iniciando o processo na camada de apresentação, quando o utilizador carrega na *dropdown* com o texto “Catalog”, é espoletado um evento no componente responsável por suportar a exibição da mesma. Em seguida, é chamado um método semelhante ao apresentado no Código 7 que permite identificar a *role* do utilizador e, conseqüentemente, efetuar um pedido HTTP do tipo GET à API. Caso a *role* do utilizador seja igual a *Administrator*, *Trainer*, *Training Manager* ou *Local Coordinator* o pedido efetuado à API retorna a lista de todos os cursos armazenados na base de dados. Contudo, se a *role* do utilizador for igual a *Trainee* ou *Team Manager* o pedido efetuado à API retorna apenas a lista de cursos armazenados na base de dados com estado PUBLISHED. Isto acontece porque aquilo que efetivamente se pretende como resultado, é que um utilizador autenticado possa visualizar as sessões que irão decorrer para esses cursos no

futuro e, se permitido, o mesmo possa enviar um pedido para se inscrever numa das sessões expostas.

O Código 14 apresenta o método do serviço de cursos da API que permite retornar uma lista de cursos com estado PUBLISHED. Reforçar que a implementação deste método é genérica, sendo também utilizado para obter cursos com outro estado.

```
// GET'S ALL THE COURSES WITH PASSED {STATUS}
// RETURN TYPE - CourseDTO
@Override
public List<CourseDTO> getCoursesByStatus(String status) {
    List<Course> courses = this.courseRepository.findAll();
    List<CourseDTO> coursesDTO = new ArrayList<>();
    if(courses == null)
        return null;
    for(Course c: courses)
        coursesDTO.add(c.convertToCourseDTO(c));
    // get list equals to given {status}
    return coursesDTO.stream().filter(course ->
        course.getStatus().name().toLowerCase().equalsIgnoreCase(status.toLowerCase())
    ).collect(Collectors.toList());
}
```

Código 14 – Implementação do método de obtenção da lista de cursos com o estado passado por parâmetro

Nest momento, a API retorna o código “HTTP OK” que indica que a requisição foi bem-sucedida e a lista de cursos com estado PUBLISHED retornada. A resposta irá cair no serviço e, posteriormente, enviada para a componente de cursos responsável da camada de apresentação. Os dados são guardados numa nova instância e, conseqüentemente, mostrados ao utilizador. Na Figura 49 é apresentada a página com o código HTML que mostra aos *Trainees* e *Team Managers* os cursos com estado PUBLISHED.

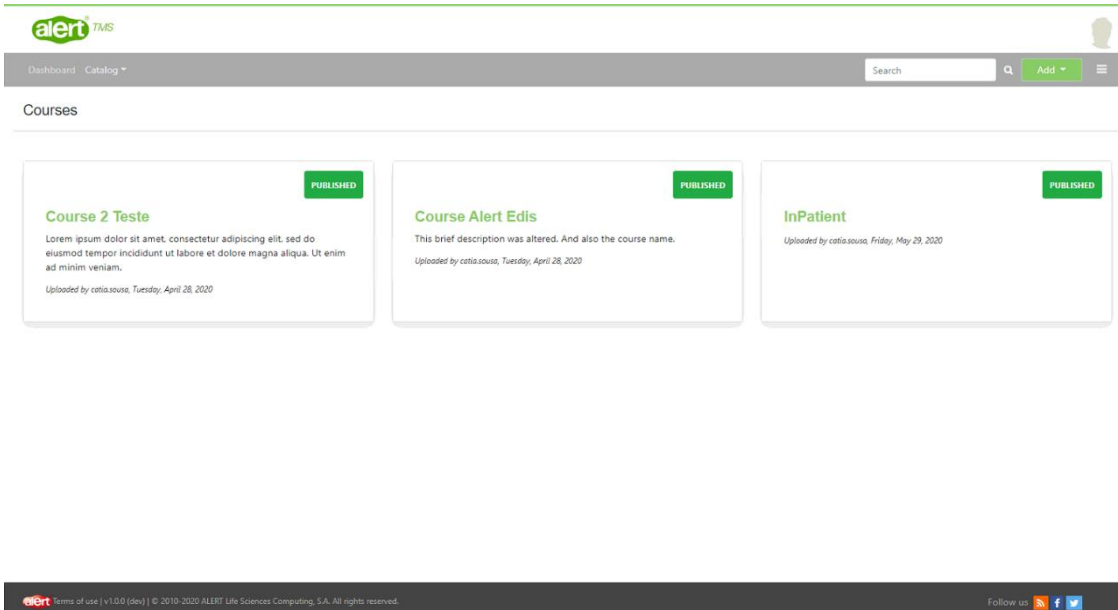
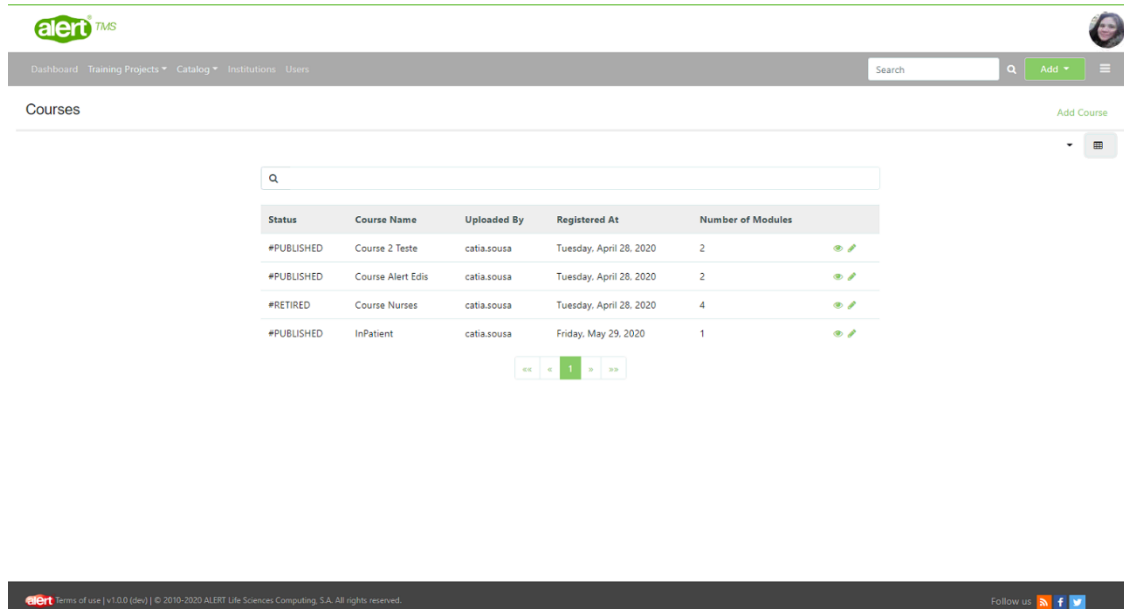










Figura 49 – Página com código HTML que lista cursos para *Trainees* e *Team Managers*

Similarmente, na Figura 50 é ilustrado a vista para *Administrator*, *Local Coordinator*, *Training Manager* e *Trainer*, que podem mudar a visualização dos cursos para tabela ou filtrar os mesmos por estados.



The screenshot displays the 'Courses' page in the alert TMS application. The page features a navigation bar with 'Dashboard', 'Training Projects', 'Catalog', 'Institutions', and 'Users'. A search bar and an 'Add' button are also present. The main content area shows a table of courses with the following data:

Status	Course Name	Uploaded By	Registered At	Number of Modules	
#PUBLISHED	Course 2 Teste	catia.sousa	Tuesday, April 28, 2020	2	 
#PUBLISHED	Course Alert Edis	catia.sousa	Tuesday, April 28, 2020	2	 
#RETIRED	Course Nurses	catia.sousa	Tuesday, April 28, 2020	4	 
#PUBLISHED	InPatient	catia.sousa	Friday, May 29, 2020	1	 

At the bottom of the page, there is a footer with the alert logo, version information (v1.0.0 dev), copyright notice (© 2010-2020 ALERT Life Sciences Computing, S.A. All rights reserved.), and social media links for YouTube, Facebook, and Twitter.

Figura 50 – Página com código HTML que lista cursos para *Administrators*, *Local Coordinators*, *Training Managers* e *Trainers*

6.6 Visualizar Detalhes de *Training Projects*

Nesta secção é apresentado o processo de visualização de *Training Projects*. Esta vista é uma das mais importantes desta plataforma uma vez que é a partir da mesma que se pode iniciar o processo de adição de *training plans*. Reforçar que a um *training project* estão associados vários *training plans*. Na Figura 51 são apresentados os fluxos de interações entre as diferentes componentes lógicas representadas na secção 5.4.

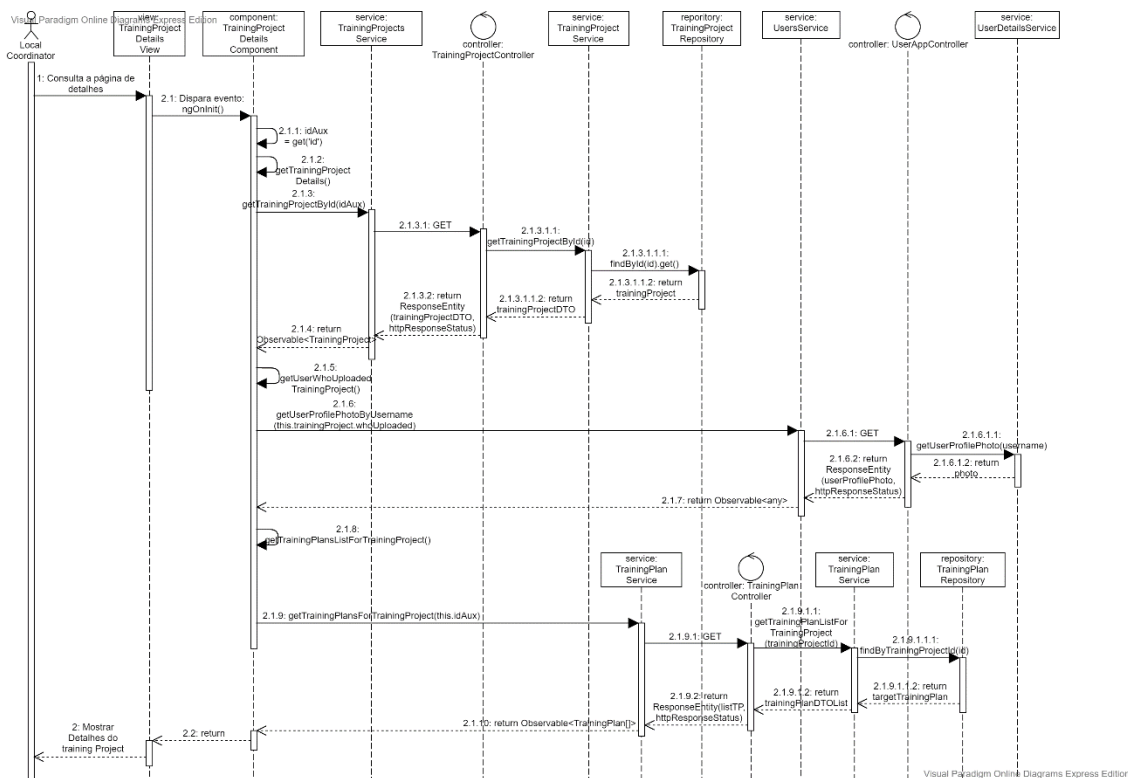


Figura 51 - Diagrama de sequência com fluxo de execução para o UC20

O processo que permite ao utilizador visualizar os detalhes de um *training project* inicia na respetiva componente da camada de apresentação. O Código 15 mostra como é efetuado o pedido à API, bem como o retorno do objeto pedido pelo utilizador autenticado. Em caso de sucesso, o código apresenta outros métodos (“*getUserWhoUploadedTrainingProject()*” e “*getTrainingPlansListForTrainingProject()*”) que também devem ser espoletados, permitindo que o pedido do utilizador seja concluído com sucesso.

```
// GET TRAINING PROJECT FROM THE SERVER
getTrainingProjectDetails(): void {
  this.trainingProjectsService.getTrainingProjectById(this.idAux).subscribe(
    (data) => {
      this.trainingProject = data;
      this.institutionTrainingProject = this.trainingProject.institution;
      this.loading = false;
      // gets photo of who uploaded the training project
      this.getUserWhoUploadedTrainingProject();
      // get training plans from the server with the id of the training
      project
      this.getTrainingPlansListForTrainingProject();
    }, error => {
      this.loading = false;
      this.errorMessageBool = true;
      this.errorMessageFromServer = error.error;
      return;
    }
  );
};
```

```
}
```

Código 15 – Implementação do método que chama o serviço de *training projects* na camada de apresentação de modo a espoletar o pedido HTTP à API

Tendo a resposta retornada o código “HTTP OK”, a segunda etapa mais importante é o pedido à API que permite receber a lista *training plans* associados. Isso acontece porque aquilo que efetivamente se pretende como resultado é que o utilizador tenha a possibilidade de visualizar todos os *training plans* numa tabela e observar os seus estados, quem criou ou consultar os detalhes dos mesmos. Mais explicitamente, uma instituição tem um ou mais *training projects*. Um desses tem um ou mais *training plans*, que, conseqüentemente, tem uma ou mais sessões. Estas relações representam a identidade de uma plataforma de gestão da formação.

Assim, seguindo boas práticas, o pedido HTTP do tipo GET deve seguir para o controlador de *training plan* da API, que obtém da base de dados uma lista de *training plans* a partir do id do *training project* passado por parâmetro. Este pedido é intermediado pelo respetivo serviço e repositório. O Código 16 apresenta o método do serviço de *training plan* associado a este pedido.

```
@Override
public List<TrainingPlanDTO> getTrainingPlanListForTrainingProject(long id)
{
    Set<TrainingPlan> targetTrainingPlan =
this.trainingPlanRepository.findByTrainingProjectId(id);
    if(targetTrainingPlan == null)
        return null;

    // transform to list dto
    List<TrainingPlan> trainingPlanList = new
ArrayList<>(targetTrainingPlan);
    List<TrainingPlanDTO> trainingPlanDTOList = new ArrayList<>();
    for(TrainingPlan tp: trainingPlanList)
        trainingPlanDTOList.add(tp.convertToTrainingPlanDTO(tp));

    return trainingPlanDTOList;
}
```

Código 16 – Implementação do método responsável por retornar a lista de training plans através do id do training project

Após retorno da lista do serviço de *training plan*, o controlador responde com um código “HTTP OK” que indica que a requisição foi concluída com sucesso, sendo o componente de *training project details* da camada de apresentação informado da resposta que com a lista. Posteriormente, e, uma vez que, todas as etapas foram terminadas sem erros, o componente informa a *view* do sucesso da página exibindo a informação relativa ao *training project* solicitado.

Na Figura 52 é apresentado o código da página HTML que exhibe a informação de um determinado *training project*. Reforçar que nesta página, existe um botão que permite adicionar prontamente um novo *training plan*, caso seja essa a intenção do utilizador autenticado.

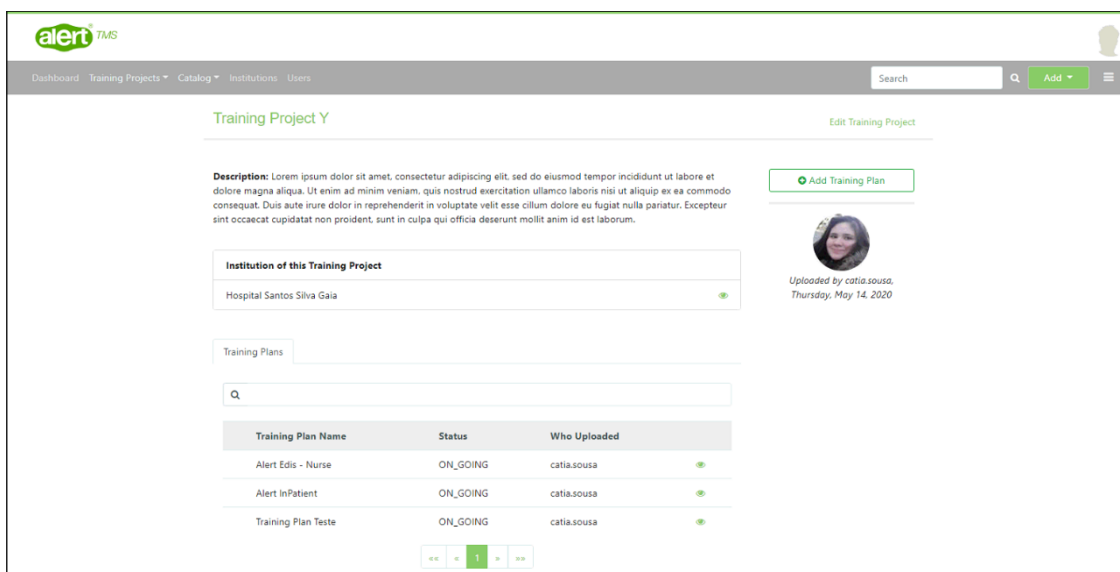


Figura 52 – Página com código HTML que exibe a informação relativa a um *Training Project*

6.7 Criar *Training Plan*

Nesta secção é descrito outro dos casos de uso mais importantes da aplicação. Como referido anteriormente, um *training plan* contém sessões e está sempre associado a um *training project*, ou seja, um utilizador autenticado e com permissão para adicionar um plano de formação, deve, obrigatoriamente, preencher o campo do formulário relativo ao *training project*. Para que tudo isto aconteça não só seguindo boas práticas, mas também que o fluxo normal que tem sido descrito a partir da secção 4 seja executado sem erros, iremos detalhar o fluxo de interações para este caso de uso como ator *Local Coordinator*.

Assim, antes de mostrar a página, é necessário carregar a *dropdown* com as instituições a que o mesmo pertence seguindo a mesma linha de interações descritas nas secções anteriores. Assim que a resposta é enviada e a lista carregada no componente de criação de *training plans*, o mesmo retorna os dados à *view* que exibe o formulário carregado. Neste momento, o utilizador irá, primeiramente, escolher uma instituição e, após esse acontecimento, é espoletado um evento ao componente que chama o respetivo serviço de modo a efetuar um pedido HTTP do tipo GET à API. Esse pedido é de extrema importância porque permite obter todos os *training projects* da instituição selecionada, pois só faz sentido mostrar na *dropdown* para o campo *training project* apenas os que pertencem àquela instituição.

Na Figura 53 são apresentados os fluxos de interações, nomeadamente, os dois descritos no parágrafo anterior e de forma ordenada, entre as diferentes componentes lógicas representadas na secção 5.4.

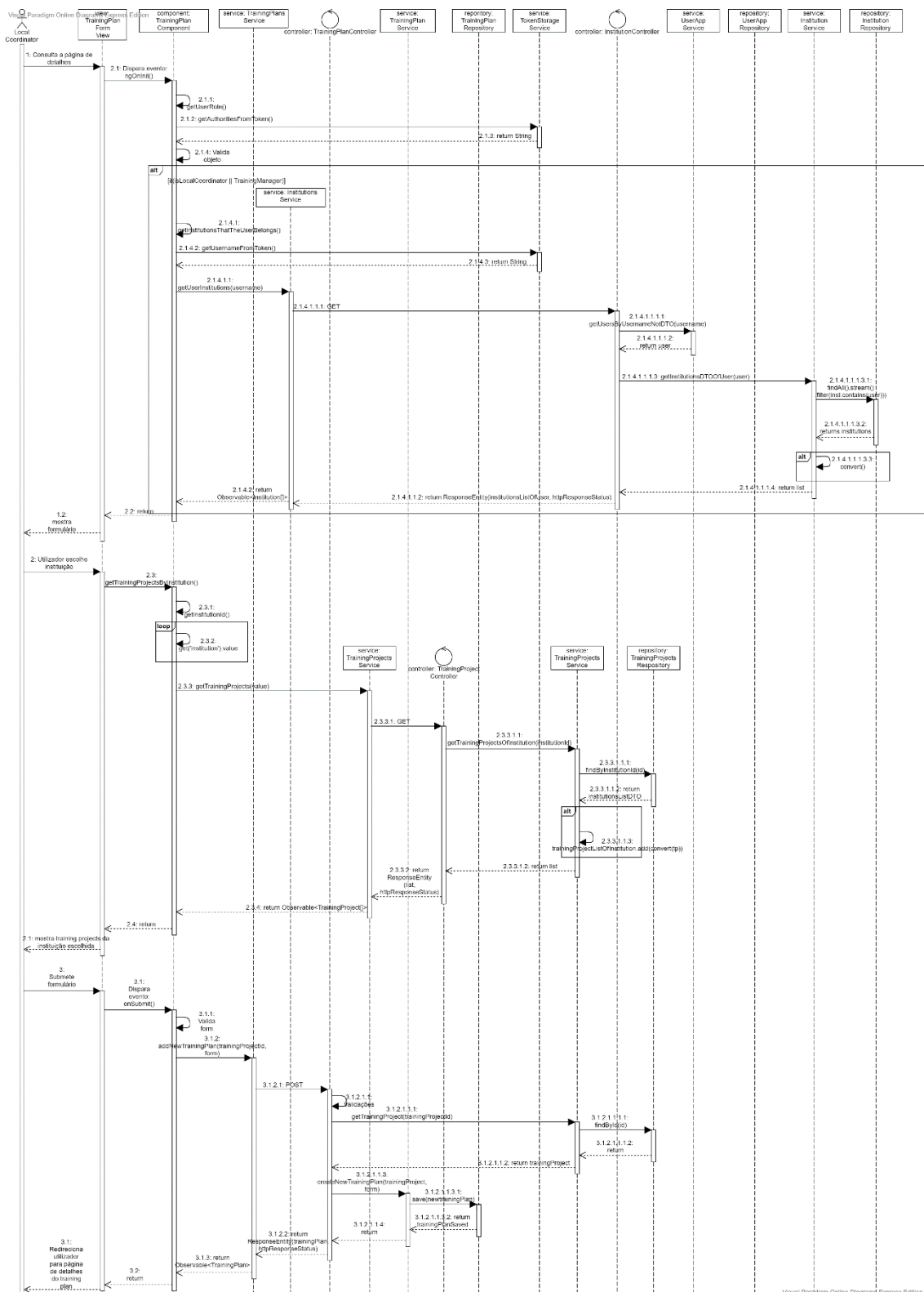


Figura 53 - Diagrama de sequência e fluxo de interação para o UC12

Quando o utilizador submete o formulário e, após as validações necessários no respetivo componente, este chama o serviço do *training plan* na camada da apresentação de forma a

enviar um pedido à API. O controlador responsável por este realiza as validações necessárias e, em caso de sucesso, chama o serviço do *training project* através do id recebido no *endpoint* como parâmetro. Quando o serviço do *training project* retornar o objeto solicitado com sucesso, o controlador do *training plan* reencaminha o pedido ao seu serviço para finalizar o processo e guardar o objeto com o *training project* a que ficará associado, bem como o nome do novo *training plan*. O Código 17 mostra o método do controlador do *training plan* responsável por estas transições.

```

@RequestMapping(value = "/trainingproject/create", method =
{ RequestMethod.GET, RequestMethod.POST })
public ResponseEntity<?>
createNewTrainingPlan(@RequestParam("trainingProjectId") String
trainingProjectId,

    @RequestBody TrainingPlanDTO form) {
    try {
        if(form == null) {
            return new ResponseEntity<>("Form passed is null.
Please make sure you provide" + " some fields.", HttpStatus.CONFLICT);
        }
        // search for Training Project
        TrainingProject trainingProject =
this.trainingProjectService.getTrainingProjectByIdReturnNotDTO(Long.parseLo
ng(trainingProjectId));

        if(trainingProject == null)
            throw new RuntimeException();
        // check if on a Array of training plans inside a training
project there are no training plan with that name
        Boolean flagName =
this.trainingPlanService.checksForName(trainingProject.getId(),
form.getName());
        if(flagName)
            return new ResponseEntity<>("The name you wrote
already exists for this specific " + "training projec. Please choose
another one.", HttpStatus.CONFLICT);

        TrainingPlanDTO trainingPlanDTOCreated =
this.trainingPlanService.createNewTrainingPlan(form, trainingProject);

        return new ResponseEntity<>(trainingPlanDTOCreated,
HttpStatus.OK);
    } catch(RuntimeException e) {
        return new ResponseEntity<>("Failed to create new training plan.",
HttpStatus.BAD_REQUEST);
    }
}

```

Código 17 - Método do controlador de *training plans* que recebe o pedido de adição

Por fim, a API retorna o código “HTTP CREATED” na resposta e a componente responsável na por concluir o processo na camada de apresentação retorna para a *view* a instância de uma *modal* (caixa com uma mensagem) centrada a informar o utilizador do sucesso da operação e, posteriormente, reencaminhar o mesmo para a página de detalhes do novo *training plan*.

A Figura 54 é o espelho do formulário que inicia os fluxos de interação explicados anteriormente.

The screenshot displays the ALERT TMS web application interface. At the top left is the 'alert TMS' logo. A navigation bar contains links for 'Dashboard', 'Training Projects', 'Catalog', 'Institutions', and 'Users'. On the right of the navigation bar is a search box with a magnifying glass icon and an 'Add' button. Below the navigation bar, the main content area is titled 'Register new Training Plan in ALERT® TMS'. This area contains three form fields: 'Institution (Client)*' with a dropdown menu showing 'Hospital Santos Silva Gaia', 'Training Project*' with a dropdown menu showing 'Training Project Y', and 'Training Plan Name*' with an empty text input field. Below these fields are two buttons: a green 'Register' button and a red 'Cancel' button. At the bottom of the page, a dark footer contains the 'alert' logo, terms of use information, and social media icons for YouTube, Facebook, and Twitter.

Figura 54 – Página com código HTML que permite ao utilizador criar um novo *training plan*

7 Testes

Na secção do presente documento são explicadas as diferenças entre testes unitários e testes de integração. Esta informação é importante para se compreender o propósito de cada tipo de teste, pois apesar de idênticos tem objetivos diferentes. Os testes são realizados para os controladores, serviços e repositórios de diferentes classes da API desenvolvida.

7.1 Comparação entre Testes Unitários e de Integração

Em palavras simples, quando se realiza um teste unitário, significa que se testa apenas uma unidade de código ou um método de cada vez, excluindo todos os outros componentes que interagem com aquilo que se está a testar.

Por outro lado, nos testes de integração, aquilo que efetivamente se pretende testar é a integração entre componentes. Se os testes unitários não falharem, significa que o comportamento individual dos componentes é o desejado, no entanto, não sabemos como devem funcionar de forma completa. Essa é a responsabilidade dos testes de integração.

7.2 Contextualização

Na Figura 55 é ilustrada a janela que permite visualizar o resultado da execução dos dois tipos de testes.

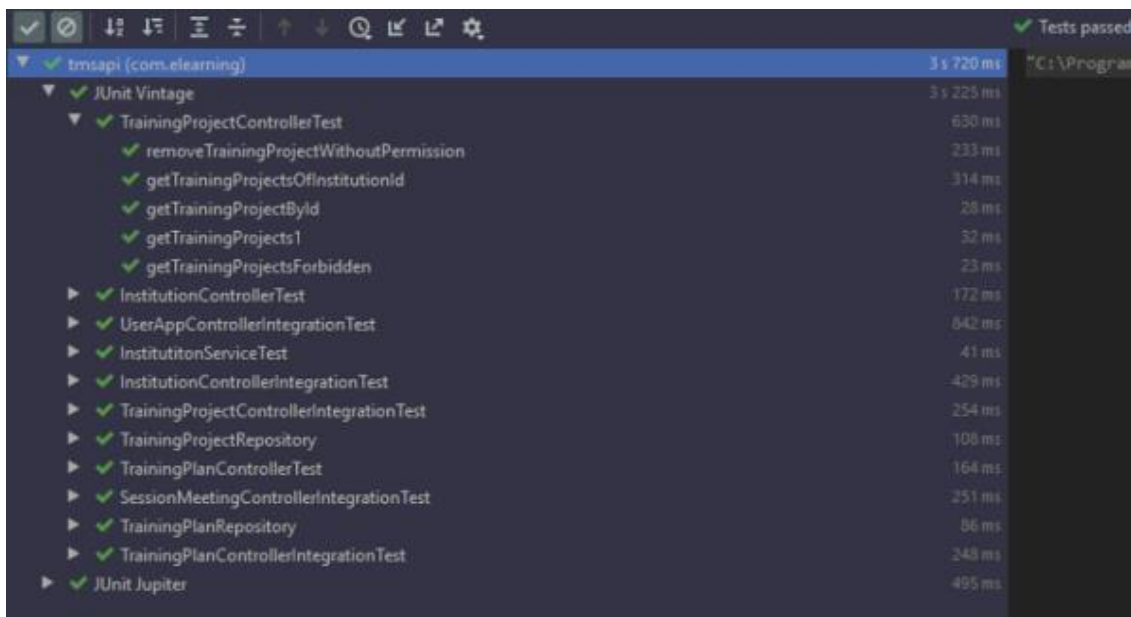


Figura 55 - Resultados da execução dos testes unitários e integração

Como é possível observar através da figura anterior (Figura 55), foram realizados alguns testes às camadas principais da API desenvolvida, isto é, aos controladores, serviços e repositórios. Para que tal fosse possível, foi adicionada a dependência `spring-boot-starter-test` disponibilizada pela Spring Framework que contém bibliotecas como JUnit, AssertJ, Mockito e JsonPath que permitem a realização de testes (baeldung, 2020).

Nas secções seguintes são descritos os detalhes de execução de testes unitários e testes de integração.

7.3 Testes Unitários

Nesta secção são utilizados *mocks* para testar unidades do sistema individualmente. O uso de *mocks* permite isolar o comportamento do objeto a testar, garantindo que o mesmo não tem dependências com outros mais complexos. Em suma, um *mock* serve para criar objetos que simulam o comportamento de outros reais. Para além disso, são utilizadas as seguintes anotações:

- “@MockBean” para adicionar objetos simulados ao contexto da aplicação;
- “@Before” no método que deve ser executado antes dos testes. Isto permitirá instanciar objetos que devem ser usados nos testes da unidade a ser testada.

7.3.1 Controladores

Como referido na secção 5, as classes designadas como controladores são responsáveis por receber e enviar pedidos de forma a comunicar com os serviços presentes na camada de apresentação. Na **Erro! A origem da referência não foi encontrada.** são apresentados os *mocks* criados para a classe "TrainingProjectController".

```
@RunWith(SpringRunner.class)
@SpringBootTest
@AutoConfigureMockMvc
public class TrainingProjectControllerTest {

    @Autowired
    private MockMvc mvc;
    @MockBean
    private ITrainingProjectService trainingProjectService;
    @MockBean
    private IInstitutionService institutionService;
    @MockBean
    private IUserAppService userAppService;
```

Figura 56 - Declaração de *mocks* para os testes unitários da classe "TrainingProjectController"

Através da figura anterior (Figura 56), compreende-se que:

- É usado o "MockMvc" para invocar e testar os *endpoints* do controlador de *training projects* sem inicializar um servidor (baeldung, 2020);
- É usada a anotação "@MockBean" que identifica os *mocks* criados para testar o comportamento dos serviços de *training projects*.

Na secção do presente documento, apenas será mostrado um dos testes unitários realizados para o controlador de *training projects*, nomeadamente, o teste que permite obter um *training project* através do seu identificador único (id). Na Figura 57 é apresentado esse teste.

```

/**
 * Teste para obter um training projet pelo seu id
 * @throws Exception
 */
@Test
public void getTrainingProjectById() throws Exception {

    Mockito.when(trainingProjectService.getTrainingProjectById(Long.parseLong("1"))).thenReturn(myTrainingProjectTest1);

    MvcResult result = mvc.perform(get( uriTemplate: "/api/trainingprojects/byId?id=1" )
        .header( name: "authorization", accessTokenUser1)
        .contentType(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath( expression: "$.name", is(myTrainingProjectTest1.getName())))
        .andReturn();
}

```

Figura 57 - Teste unitário que permite obter um *training project* através do seu id

Na figura anterior (Figura 57), com recurso à biblioteca Mockito é usado o *mock* do serviço de *training projects* para simular o comportamento dos objetos. Aquilo que efetivamente se pretende é chamar o método “getTrainingProjectById()” para retornar o objeto indicado por parâmetro. Posteriormente, é usado o “MockMvc” para realizar o pedido HTTP para o *endpoint* especificado através do `perform(get("/api/trainingprojects/byId?id=1"))`.

Com o `header("authorization", accessTokenUser1)` é definido o utilizador com permissão para realizar o pedido. Para além disso, através da chamada `contentType(MediaType.APPLICATION_JSON)` é determinado o `contentType` para JSON e, por último, é informada a resposta que se pretende receber (estado 200 que significa OK e o objeto retornado deve ser igual ao pedido).

7.3.2 Serviços

Na estrutura definida para o desenvolvimento da plataforma deste projeto (secção 5), é explicado que a camada de serviços da API deve comunicar com as classes DTO, modelos e repositórios. Na Figura 58 são apresentados os *mocks* criados para simular o comportamento dos repositórios para o serviço de instituições.

```

@RunWith(SpringRunner.class)
@SpringBootTest
public class InstituitonServiceTest {

    @MockBean
    private IInstitutionRepository institutionRepository;

    @Autowired
    private IInstitutionService institutionService;
}

```

Figura 58 – Declaração de *mocks* para os testes unitários da classe “InstitutionService”

Na figura seguinte (Figura 59), é apresentado um dos testes unitários realizados para o serviço de instituições, nomeadamente, o teste que permite criar uma instituição.

```
/**
 * Teste para criação de uma instituição
 * @throws Exception
 */
@Test
public void saveInstitution() throws Exception {

    Mockito.when(institutionRepository.save(Mockito.any(Institution.class)))
        .thenReturn(institution1);

    List<UserApp> usersInst1 = new ArrayList<>();
    usersInst1.add(user1);

    Institution institutionResult = institutionService
        .addInstitution(myInstitutionTest, usersInst1);

    assertEquals(institution1.getAcronym(), institutionResult.getAcronym());
    verify(institutionRepository, times(wantedNumberOfInvocations: 1))
        .save(Mockito.any(Institution.class));
}
```

Figura 59 – Teste unitário que permite criar uma instituição

Como se pode observar, é efetuada uma chamada ao repositório de instituições de modo a testar o seu comportamento, isto é, simular o pedido de atualização da instituição ao repositório. Após essa chamada é executado o método alvo deste teste e recebido o resultado.

A lógica de negócio do método “addInstitution()” da classe “InstitutionService” recebe como parâmetros o objeto do tipo “Institution” que se pretende adicionar e a lista do tipo “UserApp” atualizada. Esta lista contém os utilizadores que pertencem à nova instituição. Dentro do método, é criado um objeto onde se atualiza os dados passados e, em seguida, é invocado o repositório de instituições e guardado o objeto.

Assim, para validar o funcionamento do método referido anteriormente, o mesmo é chamado e, conseqüentemente, passado os valores criados (“institution1” e “usersInst1”). Em seguida, é efetuada a verificação, ou seja, se o método testado retornou o resultado esperado através do **assertEquals(institution1.getAcronym(), institutionResult.getAcronym())** e se chamou exatamente o método de atualização da instituição através do **verify(institutionRepository, times(1)).save(Mockito.any(Institution.class))**.

7.3.3 Repositórios

Como referido na secção 5, para testar os repositórios da API desenvolvida, é necessário usar a própria base de dados Oracle em funcionamento. Contudo, para que os dados armazenados na mesma não sejam alterados ou removidos, é adicionada a anotação “@DataJpaTest” no topo da classe de teste do repositório de *training plans* para que a execução dos mesmos, bem como todas as transações realizadas possam ser revertidas. Na Figura 60 são apresentadas essas anotações e a definição dos repositórios fundamentais para os testes desta classe.

```
@RunWith(SpringRunner.class)
@DataJpaTest
@AutoConfigureTestDatabase(replace = Replace.NONE)
public class TrainingPlanRepository {

    @Autowired
    private ITrainingPlanRepository trainingPlanRepository;
    @Autowired
    private ITrainingProjectRepository trainingProjectRepository;
    @Autowired
    private IIInstitutionRepository institutionRepository;
}
```

Figura 60 – Configuração da classe “TrainingPlanRepository”

Na secção 4.5, é apresentada a relação entre a tabela “TRAINING_PLAN” e a tabela “TRAINING_PROJECT”, onde se explica que a primeira tabela possui uma coluna para guardar a chave primária da segunda tabela. Assim, na figura seguinte (Figura 61/Figura 59), é apresentado um dos testes unitários realizados para o repositório de *training plans*, nomeadamente, o teste que permite obter os metadados de *training plans* da base de dados através do id do *training project* pedido.

```
/**
 * Teste para obter os metadados dos training plans da base de dados através do id do seu training project
 * @throws Exception
 */
@Test
public void getTrainingPlanByTrainingProjectId() throws Exception {

    Set<TrainingPlan> result = trainingPlanRepository
        .findByTrainingProjectId(myTrainingPlanTest.getTrainingProject().getId());

    List<TrainingPlan> listResult = new ArrayList<>(result);

    assertEquals(listResult.get(0).getName(), listResult.get(0).getName());
    assertEquals(listResult.get(0).getRegisteredAt(), listResult.get(0).getRegisteredAt());
    assertEquals(listResult.get(0).getStatus(), listResult.get(0).getStatus());
    assertEquals(listResult.get(0).getWhoUploaded(), listResult.get(0).getWhoUploaded());
}
```

Figura 61 – Teste unitário que permite obter os metadados de *training plans* através do id do *training project* pedido

Analisando o teste apresentado na figura anterior (Figura 61), inicialmente são armazenados na base de dados utilizadores, instituições, *training projects* e *training plans* pertencentes a uma lista. Esses dados, como referido anteriormente, são criados no início da classe deste teste no método com a anotação “@Before”. Assim que esses elementos estiverem armazenados na base de dados, é possível executar este teste, invocando o método “findByTrainingProjectId()” do repositório de *training plans* que espolta uma *query* à base de dados para obter a coleção não ordenada de *training plans* com o id do *training project* passado por parâmetro. Em seguida, é convertida a coleção não ordenada numa lista de modo a que na verificação seja mais fácil obter qualquer índice dessa lista.

Em suma, tendo obtido a lista de *training plans* como resultado do método testado, em particular para o exemplo apresentado, só são efetuadas comparações entre as propriedades do primeiro índice da lista retornada, ou seja, se o *training plan* obtido corresponde ao *training plan* esperado.

7.4 Testes de Integração

Na Figura 62 são apresentadas as configurações necessárias para a execução dos testes de integração do controlador de sessões.

```
@RunWith(SpringRunner.class)
@SpringBootTest
@AutoConfigureMockMvc
@Transactional
public class SessionMeetingControllerIntegrationTest {

    @Autowired
    private MockMvc mvc;
    @Autowired
    private ITrainingPlanRepository trainingPlanRepository;
    @Autowired
    private ITrainingProjectRepository trainingProjectRepository;
    @Autowired
    private IIstitutionRepository institutionRepository;
    @Autowired
    private ISessionMeetingRepository sessionMeetingRepository;
```

Figura 62 - Declaração de *mocks* para os testes de integração da classe “SessionMeetingController”

Como referido na secção 7.3.3, o JPA não oferece gestão das transações realizadas à base de dados. Por causa disso, é necessário adicionar no início da classe a ser testada (neste exemplo, “SessionMeetingControllerIntegrationTest”), a anotação “@Transactional” que trata automaticamente das transações executadas à base de dados sem perda de informação.

Na Figura 63 é apresentado um dos testes de integração realizados ao controlador de sessões, em particular, o teste que permite obter todas as sessões existentes na base de dados.

```
/**
 * Teste para obter todas as session meetings existentes
 * @throws Exception
 */
@Test
public void getSessionMeetings() throws Exception {
    mvc.perform(get( urlTemplate: "/api/sessionmeetings" )
        .header( name: "authorization", accessTokenUser1 )
        .contentType( MediaType.APPLICATION_JSON )
        .andExpect( status().isOk() )
        .andExpect( jsonPath( expression: "$" ).isEmpty() );
}
```

Figura 63 – Teste de integração que permite obter todas as sessões existentes na base de dados

Em primeiro lugar, é executado o método com a anotação “@Before” que permite armazenar na base de dados uma lista de sessões. Após essa chamada, é utilizado o “MockMvc” para simular o pedido HTTP à API que permite obter todas as sessões existentes na base de dados. De facto, o teste da figura anterior (Figura 63) apresenta a mesma estrutura do teste da Figura 57. A grande diferença entre os dois é que no teste desta secção não são criados *mocks* e os cenários são executados de ponta a ponta, isto é, passa por todas as camadas da API desenvolvida.

8 Experimentação e Avaliação

Neste capítulo, é realizada uma contextualização do problema a resolver, bem como o processo adotado para efetuar a experimentação e avaliação do projeto desenvolvido. Por fim, são apresentados os resultados e conclui-se se a plataforma desenvolvida tem uma boa usabilidade.

A ALERT possui uma plataforma e-learning que tem como funcionalidade principal a disponibilização de conteúdos relativos aos vários produtos clínicos da empresa, de modo a manter os seus colaboradores e clientes atualizados. Com isto, a ALERT consegue disponibilizar, visualizar, criar, editar ou eliminar conteúdos dos módulos dos cursos frequentados pelo utilizador. Contudo, a empresa sente necessidade de realizar atividades (exercícios práticos) ou gerir inscrições durante a ação de formação em sala. Assim, este requisito foi considerado o mais importante para a empresa.

O produto desenvolvido será utilizado diariamente através de um dispositivo com ligação à Internet. Desse modo, foi realizado um inquérito para apurar a perceção de usabilidade da plataforma Web desenvolvida. Usabilidade é um termo utilizado para descrever a qualidade de interação dos utilizadores com algum tipo de interface gráfica.

Para alcançar uma boa usabilidade, a plataforma desenvolvida deve carregar rapidamente, ou seja, o tempo de resposta de um sistema para que o utilizador não perca o interesse deve ser curto. Problemas de usabilidade ocorrem quando determinada característica do sistema acaba por prejudicar ou inviabilizar a realização de uma tarefa, aborrecendo o utilizador que o utiliza.

De modo geral, é possível identificar um problema de usabilidade durante a interação utilizador-sistema, podendo a sua origem estar no levantamento de funcionalidades. Exemplos de problemas de usabilidade podem variar desde a ligação à Internet (lenta ou rápida) até à incompatibilidade entre os *browsers* que não suportam de igual forma a mesma tecnologia para o desenvolvimento de interfaces gráficas.

Da mesma forma, um utilizador pode perder o interesse em utilizar o sistema se o mesmo for difícil de compreender, falhar no levantamento das suas funcionalidades (o que oferece), se o utilizador se perder durante a navegação ou se a informação for difícil de ler ou não responder ao que esperava. Assim, qualquer aplicação Web deve ser submetida a testes de usabilidade e outras técnicas de validação para melhorar e monitorizar o seu uso.

8.1 Objetivo

Face ao exposto, a usabilidade é um requisito crítico para o sucesso de qualquer produto. A avaliação da mesma tem por objetivo avaliar a interação do sistema a desenvolver nos seguintes aspetos (McLellan, Muddimer, & Peres, 2012):

- Eficácia – estimar a interação do utilizador com o sistema;
- Eficiência – refere-se às funcionalidades oferecidas e se através delas o utilizador consegue atingir os objetivos, como por exemplo, rapidez com que o utilizador executa as tarefas;
- Satisfação – refere-se à reação positiva do utilizador quando interage com o sistema;
- Verificar se após a primeira interação, o utilizador consegue lembrar o comportamento do sistema sem necessitar de pedir nova ajuda.

As avaliações de usabilidade para o produto desenvolvido permitem obter resultados como: registar problemas efetivos de usabilidade durante a interação; calcular métricas quanto à eficácia, eficiência e produtividade do utilizador aquando a sua interação com o sistema proposto; conhecer a opinião do utilizador sobre o sistema; perceber se o levantamento de requisitos foi efetuado com sucesso ou não (McLellan, Muddimer, & Peres, 2012).

Assim, o inquérito a realizar foi distribuído por seis colaboradores da ALERT, uma vez que, a aplicação ainda não se encontra disponível para todos. Assim, os inquiridos foram também aqueles que de forma direta ou indireta ajudaram no levantamento de requisitos e funcionalidades implementadas no produto desenvolvido.

8.2 Metodologia

System Usability Scale (SUS) é um método criado por John Brooke em 1986 e pode ser utilizado para avaliar produtos e serviços (McLellan, Muddimer, & Peres, 2012). Os critérios do SUS ajudam a avaliar a efetividade (quais são os utilizadores que conseguem completar os seus objetivos), eficiência (quais são os esforços e recursos necessários para isso) e satisfação (se a experiência foi positiva) (McLellan, Muddimer, & Peres, 2012).

Deste modo, a SUS é um inquérito com dez (10) afirmações, as quais os utilizadores podem responder utilizando a escala de Likert desenvolvida também por John Brooke, que vai desde um (1) a cinco (5), sendo que 1 significa “Discordo Completamente” e 5 significa “Concordo Completamente” (McLellan, Muddimer, & Peres, 2012). A Figura 64 mostra a escala de Likert.



Figura 64 - Escala de Likert. Adaptado de Sam, Andrew e Camille (2012)

As perguntas base do SUS podem ser modificadas de acordo com o contexto do produto em desenvolvimento. A utilização deste método ajuda a compreender se os requisitos impostos continuam a ser os mais adequados, ou se é necessário repensar e implementar de forma diferente, eliminando pontos negativos, como por exemplo, o facto de o utilizador não concordar com a interface gráfica apresentada para uma funcionalidade em específico (McLellan, Muddimer, & Peres, 2012).

Como referido anteriormente, o SUS é constituído por 10 afirmações, que se encontram formuladas em Inglês. Assim, foi necessário alterar as mesmas para a versão Portuguesa. Este processo de tradução e adaptação resultou noutras 10 afirmações, consideradas equivalentes ao item correspondente do SUS original (versão inglesa). A Tabela 34 mostra a questão em inglês e respetiva tradução.

Tabela 34 - Item original vs item correspondente em Português Europeu (Martins, Rosa, Queirós, Silva, & Rocha, 2015)

Item Original	Item correspondente em Português Europeu
I think that I would like to use this system frequently.	Acho que gostaria de utilizar este produto com frequência.
I found the system unnecessarily complex.	Considerei o produto mais complexo do que necessário.
I thought the system was easy to use.	Achei o produto fácil de utilizar.
I think that I would need the support of a technical person to be able to use this system.	Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto.
I found the various functions in this system were well integrated.	Considerei que as várias funcionalidades deste produto estavam bem integradas.
I thought there was too much inconsistency in this system.	Achei que este produto tinha muitas inconsistências.

Item Original	Item correspondente em Português Europeu
I would imagine that the most people would learn to use this system very quickly.	Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto.
I found the system very cumbersome to use.	Considereei o produto muito complicado de utilizar.
I felt very confident using the system.	Senti-me muito confiante a utilizar este produto.
I needed to learn a lot of things before I could get going with this system.	Tive que aprender muito antes de conseguir lidar com este produto.

Tendo em consideração a tradução apresentada anteriormente, e, uma vez que, as questões do SUS podem ser adaptadas de modo a melhor se encaixarem no contexto deste projeto, foi necessário substituir a palavra “produto” por “plataforma”. Esta pequena alteração pode ser visualizada no Anexo A – Questionário de Usabilidade.

O questionário foi realizado recorrendo ao Google Forms, onde cinco dos seis inquiridos eram utilizadores da plataforma com perfil *Trainee* e o último, com perfil *Administrator* (descrição presente na secção 4.1). Quanto à amostragem, as idades dos seis elementos selecionados encontravam-se compreendidas entre os vinte e quatro (24) e os quarenta e dois (42) anos.

8.3 Execução

Como referido anteriormente, primeiramente foi necessário estudar o SUS e compreender se o mesmo satisfazia o objetivo/propósito desta secção. Tendo-se verificado que sim, procedeu-se ao levantamento da tradução das dez afirmações e, conseqüentemente, foi escolhido uma plataforma online que permitia disponibilizar ao inquirido uma hiperligação que redirecionava para o questionário online. Em seguida, foi pedida a colaboração dos seis elementos e, posteriormente, reunidos os respetivos contactos. Por fim, foi elaborado um texto que solicitava, agradecia, informava a data limite para responder e disponibilizava o endereço por correio eletrónico.

8.4 Resultados

Nesta secção são analisados os resultados obtidos no questionário, tendo sido verificada uma taxa de aprovação de 100%, ou seja, os seis inquiridos responderam não só ao questionário, como também a todas as questões presentes no mesmo. Uma das conclusões após análise dos resultados que efetivamente, o SUS não fornece informações sobre problemas específicos da plataforma, no entanto, provê *feedback* sobre a facilidade geral de utilização da plataforma da perspetiva do utilizador.

Na Tabela 35 são apresentados os dados obtidos das respostas dadas pelos inquiridos, sendo mostrado, por questão, o número de votos, a percentagem e a negrito a opção mais votada.

Tabela 35 - Dados das respostas do questionário

Questão	1	2	3	4	5
1 - Acho que gostaria de utilizar esta plataforma com frequência.	0 (0%)	0 (0%)	0 (0%)	1 (16,7%)	5 (83,3%)
2 - Considerei a plataforma mais complexa do que necessário.	4 (66,7%)	2 (33,3%)	0 (0%)	0 (0%)	0 (0%)
3 - Achei a plataforma fácil de utilizar.	0 (0%)	0 (0%)	0 (0%)	1 (16,7%)	5 (83,3%)
4 - Acho que necessitaria de ajuda de um técnico para conseguir utilizar esta plataforma.	3 (50%)	3 (50%)	0 (0%)	0 (0%)	0 (0%)
5 - Considerei que as várias funcionalidades desta plataforma estavam bem integradas.	0 (0%)	0 (0%)	0 (0%)	2 (33,3%)	4 (66,7%)
6 - Achei que esta plataforma tinha muitas inconsistências.	4 (66,7%)	2 (33,3%)	0 (0%)	0 (0%)	0 (0%)
7 - Suponho que a maioria das pessoas aprenderia a utilizar rapidamente esta plataforma.	0 (0%)	0 (0%)	1 (33,3%)	1 (16,7%)	4 (50%)
8 - Considerei a plataforma muito complicada de utilizar.	4 (66,7%)	2 (33,3%)	0 (0%)	0 (0%)	0 (0%)
9 - Senti-me muito confiante a utilizar este produto.	0 (0%)	0 (0%)	0 (0%)	2 (33,3%)	4 (66,7%)
10 - Tive que aprender muito antes de conseguir lidar com este produto.	4 (66,7%)	2 (33,3%)	0 (0%)	0 (0%)	0 (0%)

Observando as respostas de um inquirido e a pontuação numérica correspondente para cada resposta, é possível calcular a pontuação geral do SUS usando a seguinte estrutura (McLellan, Muddimer, & Peres, 2012):

- Para questões ímpares, subtrair o valor um (1) à pontuação que o utilizador atribuiu;
- Para questões pares, subtrair a resposta ao valor cinco (5), ou seja, se o inquirido respondeu dois (2), então deve ser contabilizado três (3). Se o inquirido respondeu quatro (4), então deve ser contabilizado 1;
- Somar as pontuações tanto das questões ímpares, como das perguntas pares e multiplicar por 2,5;

- O resultado pode variar entre zero (0) e cem (100). Este permite compreender o valor da usabilidade da plataforma.

A pontuação média do SUS é 68. Isso simplesmente significa que uma pontuação com esse valor na classificação final, é considerada aceitável. Para John Brooke, se o resultado obtido for inferior a 68 significa que a plataforma está a passar por problemas de usabilidade. Na Tabela 36 é apresentada a orientação geral sobre a interpretação do resultado do SUS.

Tabela 36 - Valores de referência para interpretação do resultado do SUS (McLellan, Muddimer, & Peres, 2012)

Pontuação	Nota	Adjetivo de Classificação
> 80.3	A	Excelente
68 – 80.3	B	Bom
68	C	Aceitável
51 - 65	D	Fraco
< 51	F	Péssimo

Na Figura 65 são apresentadas as respostas de cada um dos inquiridos, bem como a data e hora a que os mesmos responderam, segundo o processo descrito nos parágrafos anteriores.

Carimbo de data/hora	1. Acho q	2. Consid	3. Achei a	4. Acho qu	5. Conside	6. Achei qu	7. Supor	8. Conside	9. Senti-me	10. Tive q
2020/06/22 15:11:19	4	2	5	2	4	2	5	1	5	1
2020/06/22 15:32:47	5	2	4	2	4	2	4	2	4	2
2020/06/25 08:54:10	5	1	5	1	5	1	5	1	5	1
2020/06/26 13:07:39	5	1	5	1	5	1	5	1	5	1
2020/06/26 10:35:52	5	1	5	2	5	1	3	2	4	2
2020/06/26 17:36:39	5	1	5	1	5	1	5	1	5	1

Figura 65 – Inserção das respostas dos inquiridos no Excel

Tendo as respostas foi possível calcular o resultado bruto do SUS (usando a estrutura enumerada nos parágrafos anteriores, onde às respostas ímpares se subtrai pelo valor 1 e às respostas pares se subtrai pelo valor 5), bem como o resultado final do SUS (resultado bruto multiplicado por 2,5). A Figura 66 apresenta esses cálculos.

	SUS Resultado Bruto	SUS Resultado Final	Nota	Classificação
	35	87,5	A	Excelente
	31	77,5	B	Bom
	40	100	A	Excelente
	40	100	A	Excelente
	34	85	B	Bom
	40	100	A	Excelente
Média	36,66666667	91,66666667	A	Excelente

Figura 66 – Resultados obtidos da pontuação SUS

Analisando os resultados, nomeadamente e a média efetuada para cada coluna (“SUS Resultado Bruto”, “SUS Resultado Final”, “Nota” e “Classificação”), verifica-se uma satisfação por parte dos inquiridos face à plataforma desenvolvida. Como é possível verificar, a pontuação mais baixa é igual a 77,5, o que corresponde a uma nota “B” e classificação “Bom”. Apesar de ainda existir uma pontuação com classificação igual a essa, as restantes apresentam todas uma nota “A” e uma classificação “Excelente”. Quanto ao resultado bruto do SUS, é possível observar que o mesmo foi aproximadamente 36,7 e o resultado final do SUS foi aproximadamente 91,7.

8.5 Conclusão

Com base no estudo realizado nas secções anteriores, é possível aferir que o SUS é um método rápido e confiável para avaliar a usabilidade das soluções desenhadas. Assim, conclui-se que o SUS serve apenas para ajudar os designers e “investigadores” a compreenderem o quão grave é o problema de usabilidade de uma determinada plataforma. Para este projeto, o resultado final do SUS é de aproximadamente 91,7, o que permite aferir que a plataforma desenvolvida não enfrenta problemas graves de usabilidade.

9 Conclusão

Neste último capítulo são apresentadas as conclusões relativamente ao projeto desenvolvido. Posteriormente, são referidos os objetivos alcançados, bem como as principais limitações que surgiram durante o desenvolvimento da solução adotada para este projeto. Por último, são feitas algumas apreciações sobre o resultado.

9.1 Objetivos Alcançados

O grande objetivo deste projeto passava pelo desenvolvimento de uma nova plataforma a executar num ambiente *web*. Como a própria palavra indicou, este projeto não apresentava versões antigas, o que exigiu um estudo intensivo de plataformas idênticas à desenvolvida, nomeadamente, *layout*, tecnologias e arquiteturas.

Não obstante, foi necessário estudar a área de *training* no seu grosso modo, mas acima de tudo, da ALERT. O mais importante era compreender a formação e tudo aquilo que a envolvia e implicava, em particular, gestão de *training projects*, *training plans*, *sessions*, inscrições em cursos, associação de um utilizador a uma instituição, visualização das formações a que deve atender, entre outros. Com isto, é possível aferir que o objetivo foi bem-sucedido.

Os requisitos definidos foram implementados e encontram-se em funcionamento apenas para um grupo restrito de colaboradores, uma vez que, se encontra em fase de testes e ainda se pretende adicionar mais *features* antes da disponibilização da plataforma para todos os colaboradores ALERT.

Relativamente às tecnologias selecionadas, foram claramente um processo demorado, uma vez que, me eram desconhecidas. Nos dias correntes, Java é uma linguagem popular, flexível e bem documentada. Quanto à linguagem utilizada para o *front-end*, o Angular 8, contrariamente a outras, permitia a atualização dinâmica e apenas referente à solicitação em

questão, sem o recarregamento total da página. Isso reduz a quantidade de dados transmitidos, o tempo de processamento e elimina a necessidade de recarregar a página inteira sempre que qualquer requisição for efetuada.

Dessa forma, e experiência do utilizador, como estudado, tornou-se mais objetiva, ou seja, a comunicação entre o *browser* e o servidor é praticamente transparente, sendo portanto o objetivo de usabilidade, performance, desempenho, configurabilidade e privacidade bem-sucedidos.

Por fim, no que respeita às práticas e ao desenvolvimento de conhecimento sobre as componentes referidas na secção 1.5, foram alcançadas com sucesso. Tendo também em consideração o estado da plataforma até ao momento de submissão deste documento, a secção 8 mostra a utilização do SUS de modo a obter o valor de usabilidade da plataforma. A utilização deste método ajudou a obter uma visão geral para que no futuro se continue no mesmo caminho, no entanto, com novas *features*.

9.2 Limitações e Trabalho Futuro

Em relação à plataforma desenvolvida, comprovou-se que a mesma apresenta um vasto leque de potencialidades, ou seja, à medida que os requisitos foram sendo desenvolvidos, surgiam outros que se adequavam às necessidades de formação da ALERT. Naturalmente, o desenvolvimento deste novo produto tornou-se num processo contínuo cujas funcionalidades extra identificadas e não implementadas, serão desenvolvidas no futuro.

Até ao momento de submissão deste documento, não foi possível realizar a comunicação entre os produtos ALERT® eLEARNING e ALERT® TMS, uma vez que, o primeiro apresenta um tipo de tecnologia e linguagens antigas. De facto, no estudo do estado da arte (secção 2) foi perceptível que um LMS deve comunicar com um TMS, e vice-versa, complementando-se. Consequentemente, como ambos os produtos não se encontram no mesmo avanço tecnológico, é necessário em primeiro lugar alterar ou adaptar o produto e-learning de modo a que ambos possam comunicar.

9.3 Apreciação Final

Em suma, o balanço final do trabalho é bastante positivo. Ficou o sentimento de dever cumprido e o reconhecimento por parte da ALERT, através de uma proposta de emprego para integrar a organização e continuar a desenvolver a plataforma. Apesar de, como já referido, de o mesmo se encontrar disponível a um número restrito de pessoas, é claramente reconhecível o seu potencial e os problemas que solucionará aquando da disponibilização a todos os colaboradores ALERT. Reforçar que os objetivos desta dissertação foram concluídos, despertando um interesse acrescido para continuar a trabalhar no desenvolvimento de plataformas deste género.

10 Referências

- (2020). Obtido de MockFlow: <https://mockflow.com/apps/wireframepro/>
- Adobe. (2017). *Adobe Captivate Prime Security Overview*. USA. Obtido em 2019, de <https://www.adobe.com/content/dam/acom/en/security/pdfs/ADB-Captivate-Security-WP.pdf>
- Alff, F. (14 de Fevereiro de 2018). *Balsamiq mockups: ferramenta de protótipos de interface obrigatória para todo analista de software*. Obtido em 22 de Fevereiro de 2020, de Análise de Requisitos: <https://analisederequisitos.com.br/balsamiq-ferramenta-de-prototipos-para-analistas-e-arquitetos-de-software/>
- Arterburn, R. (Dezembro de 2019). *What is Driving the Learning Management System Market*. Obtido de Fresno Observer: <https://fresnoobserver.com/what-is-driving-the-learning-management-system-market/2892/>
- Atlassian. (2020). *Jira Software*. Obtido em 18 de Junho de 2021, de <https://www.atlassian.com/software/jira>
- baeldung. (26 de Abril de 2020). *Testing in Spring Boot*. Obtido em 25 de Junho de 2020, de Baeldung: <https://www.baeldung.com/spring-boot-testing>
- Bhoir, P., & Patil, H. (2018). *Evolution of Version Control Systems and a study on TortoiseSVN*. International Research Journal of Engineering and Technology (IRJET). Obtido em 29 de Dezembro de 2019, de <https://www.irjet.net/archives/V5/i6/IRJET-V5I6301.pdf>

- Bhutanie, A., & Bhardwaj, P. (Fevereiro de 2019). *E-learning Market Size*. Obtido de Global Markets Insights: <https://www.gminsights.com/industry-analysis/elearning-market-size>
- Cameron, J. T. (2011). *The Role of Online Grading Programs in School/Community Relations* (Vol. 1). Julho. Obtido em 27 de Novembro de 2019, de <https://link.springer.com/content/pdf/10.1186%2F2229-0443-1-2-91.pdf>
- Castells, M. (2000). *La era de la información: economía, sociedad y cultura*. Madrid: Alianza Editorial.
- Computing, A. L. (1 de 10 de 2019). Plataforma da Gestão da Formação. p. 2. Obtido de <https://www4.dei.isep.ipp.pt/gestmei/files/proposta190924050604.pdf>
- Corporate E-learning Market*. (Janeiro de 2020). Obtido de Market Research Future: <https://www.marketresearchfuture.com/reports/corporate-e-learning-market-1381>
- Courses*. (2020). Obtido em 29 de junho de 2020, de Adobe: <https://helpx.adobe.com/ie/captivate-prime/learners/feature-summary/courses.html>
- Diagrama de Gantt*. (3 de Julho de 2017). Obtido em 20 de Janeiro de 2020, de CCM: <https://br.ccm.net/contents/581-diagrama-de-gantt>
- Einstein quote about Education*. (18 de Julho de 2011). Obtido de Woodward English: <https://www.woodwardenglish.com/einstein-quote-about-education/>
- Freeman, R. (2001). *A Stakeholder Approach to Strategic Management*. Virginia: Oxford: Blackwell Publishing. Obtido em 25 de Novembro de 2019, de https://www.researchgate.net/publication/228320877_A_Stakeholder_Approach_to_Strategic_Management
- Gomes, P. C. (2 de Dezembro de 2016). *Amazon Web Services: Como funcionam a AWS*. Obtido de OP Services: <https://www.opservices.com.br/amazon-web-services/>
- grokonez. (2020). *Spring Security - JWT Authentication Architecture | Spring Boot*. Obtido em 24 de Junho de 2020, de grokonez: <https://grokonez.com/spring-framework/spring-security/spring-boot-spring-security-jwt-authentication-architecture-tutorial>
- Group, P. (2018). *System Administrator: System Settings*. Obtido em 15 de Dezembro de 2019, de http://julie-grant.com/psu_book/guides/System_Administrator_System_Settings/index.html
- Hampton, L. F., Anderson, C., & Sigman, D. (2002). *The Impact on student academic achievement using an online process provided to students and parents*. ERIC Education Resources Information Centre. Obtido de <http://www.eric.ed.gov/ERICWebPortal/search/detailmini.jsp?>

_nfpb=true&_ERICExtSearch_SearchValue_0=ED475933&ERICExtSearch_SearchType_0=no&accno=ED475933

- Heiser, D., & Farah, B. N. (2020). *Value Analysis*. Obtido em 31 de Janeiro de 2020, de Reference for Business: <https://www.referenceforbusiness.com/management/Tr-Z/Value-Analysis.html>
- INSIA. (2020). *FORINSIA software de formação*. Obtido em 8 de Fevereiro de 2020, de INSIA sistemas de informação: <https://insia.pt/Produtos/FORINSIA>
- Institute, P. M. (Ed.). (2017). *A Guide to the Project Management Body of Knowledge* (Vol. sixth). Obtido em 29 de Novembro de 2019
- INTUO. (2020). Obtido em 7 de Fevereiro de 2020, de Intuo Guides: https://apidocs.intuo.io/reference#student_enrollmentsid-1
- INTUO Learn*. (2020). Obtido em 6 de Fevereiro de 2020, de GetApp: <https://www.getapp.pt/software/105399/intuo-learn>
- Kent. (7 de Outubro de 2017). *Duolingo App Review*. Obtido de Learn to Language: <http://learntolanguage.com/duolingo-review/>
- Koen, P. (Março-Abril de 2001). Provideing Clarity And a Common Language to the "Fuzzy Front End". Obtido em 31 de Janeiro de 2020, de https://web.stevens.edu/cce/NEW/PDFs/Clarity_FEE.pdf
- LaGrone, R. (2020). *Instancy Classroom Training Management Software, Solution, System*. Obtido em 5 de Novembro de 2019, de instancy: <https://www.instancy.com/classroom-training-management.html>
- Learning Programs*. (28 de Janeiro de 2020). Obtido em 7 de Fevereiro de 2020, de Adobe: <https://helpx.adobe.com/captivate-prime/administrators/feature-summary/learning-programs.html>
- Martins, A. I., Rosa, A. F., Queirós, A., Silva, A., & Rocha, P. N. (2015). *European Portuguese validation of the System Usability Scale (SUS)*. Obtido em 29 de Junho de 2020, de <https://reader.elsevier.com/reader/sd/pii/S1877050915031191?token=8F5CDAD86CA0B3AC14D2EEFD61B1CA08BC23D76E82935F5915610EADF2F353B5E8A5C031AEB1EB2BA1BB99E88FCF48A>
- McLellan, S., Muddimer, A., & Peres, C. S. (2012). *The effect of experience on system usability scale ratings*. Obtido em 17 de Fevereiro de 2020, de Semantic Scholar: <https://pdfs.semanticscholar.org/3710/c1fd770aebc06d3dc9b1392ead94caf54bc2.pdf>

- Michelle, E. (3 de Julho de 2019). *The Gamut And Size Of The eLearning Market*. Obtido em 1 de Dezembro de 2020, de eLearning Industry: <https://elearningindustry.com/growth-of-the-elearning-market-gamut-size>
- Microsoft. (2020). *O que é a cloud?* Obtido em 5 de Fevereiro de 2020, de Microsoft Azure: <https://azure.microsoft.com/pt-pt/overview/what-is-the-cloud/>
- Milener, G., Hamilton, B., & Rabeler, C. (19 de Janeiro de 2017). *What is ODBC*. Obtido em 27 de Dezembro de 2019, de Microsoft: <https://docs.microsoft.com/en-us/sql/odbc/reference/what-is-odbc?view=sql-server-ver15>
- Moreira Kenski, V. (2003). *Aprendizagem Medida pela Tecnologia*. Brasil: Revista Diálogo Educacional.
- Nicola, S. (2019). *Análise de Valor*. ISEP - Instituto Superior de Engenharia do Porto, Porto. Obtido em 1 de Fevereiro de 2020
- Oliveira, W. (6 de Julho de 2018). *Diferenças entre OLTP e OLAP*. Obtido em Dezembro de 2019, de TechTem: <https://www.techtem.com.br/diferencas-entre-oltp-e-olap/>
- Pappas, C. (2020). *Adobe Captivate Prime*. Obtido em 7 de Fevereiro de 2020, de eLearning Industry: <https://elearningindustry.com/directory/elearning-software/adobe-captivate-prime>
- Pappas, C. (2020). *TalentLMS Overview*. Obtido em 7 de Fevereiro de 2020, de eLearning Industry: <https://elearningindustry.com/directory/elearning-software/talentlms>
- Pappas, C. (2020). *TalentLMS Reviews 2020*. (eLearning Industry's Network) Obtido em 6 de Fevereiro de 2020, de eLearning Industry: <https://elearningindustry.com/directory/elearning-software/talentlms/reviews>
- Portela, F., & Queirós, R. (2018). *Introdução ao Desenvolvimento Moderno para a Web*. Lisboa: FCA - Editora de Informática.
- Portela, F., & Queirós, R. (2018). *Introdução ao Desenvolvimento Moderno para a Web*. Lisboa: FCA - Editora de Informática.
- PowerSchool. (s.d.). *System Administrator: System Settings*. Obtido em 15 de Dezembro de 2019, de http://julie-grant.com/psu_book/guides/System_Administrator_System_Settings/
- Ramos, P. E. (s.d.). *O professor frente às novas tecnologias de informação e comunicação*. Obtido em 5 de Fevereiro de 2020, de SEDUC: <http://www2.seduc.mt.gov.br/-/o-professor-frente-as-novas-tecnologias-de-informacao-e-comunicac-1>
- Rich, N., & Holweg, M. (Janeiro de 2000). Value Analysis, Value Engineering. p. 32. Obtido em 1 de Fevereiro de 2020, de https://www.urenio.org/tools/en/value_analysis.pdf

- Saaty, T. L. (2008). Decision making with the analytic hierarchy process. (I. J. Sciences, Ed.) 1(1), pp. 83 - 97. Obtido em 1 de Fevereiro de 2020, de <http://www.rafikulislam.com/uploads/resourses/197245512559a37aadea6d.pdf>
- Souza, J. E. (2007). *JDBC tutorial*. Obtido em Dezembro de 2019, de DevMedia: <https://www.devmedia.com.br/jdbc-tutorial/6638>
- TalentLMS. (2020). *Cloud, Lean and Complete LMS with an Emphasis on Usability and Easy Course Creation*. Obtido em 6 de Fevereiro de 2020, de <https://www.talentlms.com/pages/docs/TalentLMS-API-Documentation.pdf>
- The global e-learning market is expected to reach \$238 billion by 2024*. (14 de Agosto de 2019). Obtido de CISION PR Newswire: <https://www.prnewswire.com/news-releases/the-global-e-learning-market-is-expected-to-reach-238-billion-by-2024-300901582.html>
- Three-tier architectures*. (28 de Agosto de 2019). Obtido em 23 de Dezembro de 2019, de JReport: https://www.ibm.com/support/knowledgecenter/en/SS7K4U_8.5.5/com.ibm.websphere.zseries.doc/ae/covr_3-tier.html
- Training Management Software Systems*. (s.d.). Obtido em 18 de junho de 2020, de arlo: <https://www.arlo.co/training-management-system>
- (2007). *WARWICK MANUFACTURING GROUP. Product Excellence using Six Sigma (section 6)*. University of Warwick, Warwick Manufacturing Group School of Engineering. Obtido em 11 de Fevereiro de 2020, de DOCPLAYER: <https://docplayer.net/17354923-Warwick-manufacturing-group-product-excellence-using-six-sigma.html>
- Woodall, T. (2003). Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis. *Academy of Marketing Science Review*, 42. Obtido em 31 de Janeiro de 2020
- Zeithaml, V. A. (1988). *Consumer Perceptions of Price, Quality and Value: A Means-End Model and Synthesis of Evidence*. *Journal of Marketing*.

11 Anexo A – Questionário de Usabilidade

Caros colegas,

No âmbito da Tese de Mestrado sobre o tema Plataforma de Gestão da Formação, desenvolvida no ramo de Engenharia de *Software* do Departamento de Engenharia Informática (DEI) do Instituto Superior de Engenharia do Porto (ISEP), venho por este meio pedir a vossa colaboração no preenchimento do questionário que se segue.

A plataforma ALERT® TMS foi desenvolvida como uma componente independente para responder às necessidades da empresa sobre a gestão dos seus colaboradores e clientes. Este inquérito serve para compreender a perceção de usabilidade da plataforma ALERT® TMS.

Este questionário consiste no questionário System Usability Scale (SUS) e é constituído por 10 afirmações. A vossa participação voluntária neste questionário online é altamente desejável e, por isso, gostaria de pedir cerca de 5 minutos do vosso tempo para responder a todas as questões tendo em conta uma escala de 1 a 5, sendo que 1 significa discordo completamente e 5 concordo completamente. Por favor, não existe necessidade de pensar muito sobre as respostas.

As informações serão tratadas de forma confidencial e anónima e serão objeto único de tratamento estatístico. Se alguém tiver alguma dúvida sobre este questionário, entre em contacto comigo através do Skype ou do correio eletrónico.

Muito obrigada pela vossa participação,

Cátia Pinto

1. Acho que gostaria de utilizar esta plataforma com frequência.
2. Considerei a plataforma mais complexa do que necessário.
3. Achei a plataforma fácil de utilizar.
4. Acho que necessitaria de ajuda de um técnico para conseguir utilizar esta plataforma.
5. Considerei que as várias funcionalidades desta plataforma estavam bem integradas.
6. Achei que esta plataforma tinha muitas inconsistências.
7. Suponho que a maioria das pessoas aprenderia a utilizar rapidamente esta plataforma.
8. Considerei a plataforma muito complicada de utilizar.
9. Senti-me muito confiante a utilizar esta plataforma.
10. Tive que aprender muito antes de conseguir lidar com esta plataforma.