



UNIVERSIDADE D  
COIMBRA

Inês Isabel Gaspar de Sousa Coelho Lopes

**SMS - ESAÚDE**  
**(SAÚDE, MENTE E SUCESSO)**  
SELEÇÃO, CUSTOMIZAÇÃO E EXTENSÃO DE UMA  
PLATAFORMA PARA PROMOÇÃO DA LITERACIA EM  
SAÚDE MENTAL

Dissertação no âmbito do Mestrado em Engenharia Informática, na área de  
Especialização em Engenharia de Software, orientada pelo Professor Doutor Mário  
Alberto da Costa Zenha Relá e apresentada ao Departamento de Engenharia  
Informática da Universidade de Coimbra.

Setembro de 2020

*Esta página foi intencionalmente deixada em branco.*

Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Informática

# SMS - eSaúde (Saúde, Mente e Sucesso)

Seleção, Customização e Extensão de uma  
Plataforma para Promoção da Literacia em Saúde  
Mental

Inês Isabel Gaspar de Sousa Coelho Lopes

Dissertação no âmbito do Mestrado em Engenharia Informática, na área de  
Especialização em Engenharia de Software, orientada pelo Professor Doutor Mário Alberto da  
Costa Zenha Relá e apresentada ao Departamento de Engenharia Informática da Universidade  
de Coimbra.

Setembro de 2020

1 2  9 0

UNIVERSIDADE D  
COIMBRA

*Esta página foi intencionalmente deixada em branco.*

## Resumo

O presente relatório descreve o trabalho realizado no âmbito da Tese de Mestrado em Engenharia Informática (MEI), na área de especialização em Engenharia de Software (ES), do Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC). É resultante da necessidade de investigadores da Faculdade de Psicologia e Ciências da Educação (FPCE) da Universidade de Coimbra (UC) possuírem uma plataforma digital que os auxilie na implementação e sustentabilidade do Projeto Saúde, Mente, Sucesso (SMS), de criação própria, assim como levar ao aumento da adesão ao mesmo por parte dos destinatários deste programa (alunos do 3<sup>o</sup> ciclo e respetivos encarregados de educação).

O Projeto SMS representa uma intervenção nas principais causas da sintomatologia depressiva nos adolescentes, através da realização de sessões presenciais por professores e psicólogos devidamente capacitados. A plataforma pretendida (plataforma SMS-eSaúde), serve, portanto, como um suporte à divulgação do programa nessas sessões. É também um recurso que continuará disponível para a comunidade escolar onde estes jovens se inserem, após o término da intervenção.

De forma a tornar a plataforma mais atrativa aos jovens, elementos de gamificação são incluídos. Ademais, outro aspeto característico, passa pelo foco na usabilidade, já que os investigadores, técnicos de saúde mental, professores, alunos e encarregados de educação não têm formação informática, e, portanto, a solução tem de ser intuitiva e ter uma perspetiva de “produto”.

Para responder às necessidades do cliente e tratando-se também de uma plataforma de cariz educativo, após realizada a exploração e comparação das funcionalidades dos vários *Content Management Systems* - Sistemas de Gestão de Conteúdos (CMS)s, nomeadamente *Learning Management Systems* - Sistemas de Gestão de Aprendizagem (LMS)s, existentes no mercado, foi selecionado o Moodle. Uma vez que certos requisitos da plataforma SMS não existiam previamente no Moodle, novos *plugins* foram desenvolvidos ou *plugins* já existentes foram reutilizados de maneira a cumprir com os requisitos (tanto para a plataforma web, como para a aplicação móvel, um elemento adicional).

A plataforma SMS-eSaúde encontra-se alojada num servidor remoto do DEI (<http://esms.dei.uc.pt>).

## Palavras-Chave

Gamificação, Moodle, Plataforma Open Source, Plugin, Sistema de Gestão de Aprendizagem.

*Esta página foi intencionalmente deixada em branco.*

### **Abstract**

*This report describes the work done within the scope of the Master in Informatics Engineering Thesis, Software Engineering specialization, of the Department of Informatics Engineering of the Faculty of Sciences and Technology of University of Coimbra. It is a digital platform that allows the implementation and sustainability of the SMS (Saúde, Mente e Sucesso) Project and it is a collaboration with the Faculty of Psychology and Educational Sciences' researchers from University of Coimbra.*

*The SMS Project (of their own creation) represents an intervention in the main causes of depressive symptoms in adolescents, through face-to-face sessions by duly trained teachers and psychologists. The intended platform (SMS-eSaúde platform), therefore, serves as a support for the dissemination of the program in these sessions. It is also a resource that will continue to be available to the school community where these young people are inserted, after the end of the intervention.*

*In order to make the platform more attractive to young people, gamification elements are included. Furthermore, another characteristic aspect is the focus on usability, hence researchers, mental health technicians, teachers, students and guardians do not have high technological literacy, and therefore the solution must be intuitive to use.*

*After exploring and comparing the functionalities of the CMSs (Content Management Systems) available in the market, namely LMSs (Learning Management Systems), Moodle was selected to respond to the client's needs. Because certain requirements for the SMS platform were not integrated into Moodle, new plugins were developed or existing plugins were customized.*

*The SMS-eSaúde platform is server hosted at DEI (<http://esms.dei.uc.pt>).*

### **Keywords**

*Gamification, Learning Management System, Moodle, Open Source System, Plugin.*

*Esta página foi intencionalmente deixada em branco.*



## Agradecimentos

Nada na vida se alcança sozinho e este projeto não foi exceção.

O presente projeto marca o final de mais uma etapa na minha vida, a minha vida académica. Deste modo, não posso deixar de agradecer a todos aqueles que, de alguma forma, contribuíram para que este momento fosse possível.

Em primeiro lugar, gostaria de agradecer ao Prof. Doutor Mário Relá, pela confiança depositada em mim para a realização deste projeto, por ter sempre uma palavra de incentivo e motivação nas fases mais difíceis do mesmo e por me guiar da melhor maneira possível.

Gostaria também de agradecer à Prof.<sup>a</sup> Doutora Ana Paula Matos, ao Prof. Doutor José Joaquim Costa, à Prof.<sup>a</sup> Doutora Rosário Pinheiro e à Jéssica Duarte por todo o apoio, disponibilidade e confiança ao longo do desenvolvimento deste que era também, o seu projeto.

O Departamento de Engenharia Informática, para além de tudo o que me ensinou, também me presenteou com três colegas fantásticas: Bárbara, Carolina e Sofia. Obrigada por caminharem ao meu lado ao longo destes anos, por me terem feito sentir integrada e em casa, e nunca me deixarem duvidar de mim.

Agradeço também às minhas melhores amigas de infância, minhas “amigas da Figueira”, minhas irmãs: Anita, Babica, Débora e Rita. Por todos os ensinamentos ao longo da minha vida, por influenciarem a pessoa que sou hoje, por toda a paciência, apoio e amizade incondicional.

Ao Zé Pedro, por todo o seu amor, pela compreensão nas ausências, pela sua paciência a lidar com o meu feitio particular, me ajudar a querer ser sempre mais e melhor e a não desistir dos meus objetivos.

Às minhas avós, “vó” Dete e “vó” Malena com quem gostaria de estar a partilhar este momento. Embora não estejam comigo, sei que hoje estariam orgulhosas.

Por fim, não poderia deixar de agradecer à minha família: aos meus irmãos, Martim e Bernardo mas, principalmente, aos meus pais, Isabel e Fernando, por tudo o que fizeram e fazem por mim, por proporcionaram tudo o que já fiz e conquistei e, assim, tornarem este momento possível. Devo-vos tudo na vida.

**Muito, muito, muito obrigada a todos!**

*Esta página foi intencionalmente deixada em branco.*

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos do Estágio . . . . .	2
1.3	Desafios . . . . .	4
1.4	Contribuição . . . . .	5
1.5	Apresentação da Estrutura do Documento . . . . .	5
<b>2</b>	<b>Levantamento de Requisitos</b>	<b>7</b>
2.1	Análise do Problema . . . . .	7
2.2	Documentação de Requisitos . . . . .	13
2.3	Requisitos de Negócio (BIZ) . . . . .	14
2.4	Requisitos de Usabilidade (U) . . . . .	14
2.5	Requisitos Multi-Plataforma (MP) . . . . .	15
2.6	Requisitos Funcionais (F) . . . . .	15
2.6.1	Integração de Gamificação . . . . .	23
2.6.2	Identificação dos Recursos de Gamificação a Integrar na Plataforma . . . . .	26
2.6.3	Requisitos Funcionais (F) Adicionais (de Gamificação) . . . . .	27
2.7	Lista Final de Requisitos . . . . .	30
<b>3</b>	<b>Seleção de Uma Plataforma <i>Open-Source</i></b>	<b>33</b>
3.1	Sistemas de Gestão de Conteúdos (CMS) . . . . .	33
3.1.1	Sistemas de Gestão de Aprendizagem (LMS) . . . . .	34
3.2	Seleção da Plataforma <i>Open-Source</i> a Adotar . . . . .	35
3.2.1	Identificação das Plataformas Candidatas . . . . .	36
3.2.2	Restrição do Número de Plataformas Candidatas . . . . .	36
3.2.3	Comparação das Plataformas Candidatas Finais e Devida Seleção . . . . .	36
<b>4</b>	<b>Arquitetura do Moodle</b>	<b>43</b>
4.1	Design e Descrição Arquitetural do Moodle . . . . .	43
4.1.1	Arquitetura 3- <i>Tier</i> . . . . .	43
4.1.2	Componentes . . . . .	45
4.1.3	Funcionamento . . . . .	45
4.1.4	Sistema de Permissões e Papéis . . . . .	46
4.1.5	Gerar <i>Output</i> . . . . .	48
4.1.6	Abstração da Base de Dados . . . . .	48
4.1.7	<i>Logging</i> e Sistema de Eventos . . . . .	49
4.2	<i>Plugins</i> . . . . .	51
4.2.1	<i>Plugin</i> Exemplo - <i>Plugin</i> do Tipo “Bloco” . . . . .	54
4.2.2	Pasta “db” (blocks/nomedoplugin/db) . . . . .	55
4.2.3	Pasta “lang” (blocks/nomedoplugin/lang) . . . . .	56
4.2.4	Ficheiro “version.php” (blocks/nomedoplugin/version.php) . . . . .	57
4.2.5	Ficheiro “block_nomedoplugin.php” (blocks/nomedoplugin/block_nomedoplugin.php) . . . . .	57
4.2.6	Outras Convenções . . . . .	57
4.3	Moodle <i>Mobile</i> - Aplicação Móvel . . . . .	58
4.3.1	Suporte para <i>Plugins</i> . . . . .	58

<b>5</b>	<b>Identificação das Funcionalidades Adicionais</b>	<b>61</b>
5.1	Funcionalidades Suportadas . . . . .	61
5.2	Análise das Funcionalidades Suportadas e Identificação das que Necessitam ser Adicionadas . . . . .	62
<b>6</b>	<b>Desenvolvimento das Funcionalidades Adicionais</b>	<b>67</b>
6.1	Plano para Integração das Funcionalidades Adicionais . . . . .	67
6.2	Ambiente de Desenvolvimento . . . . .	67
6.2.1	Papéis e Permissões Criados . . . . .	68
6.3	Processo de Desenvolvimento e Estratégias Adotadas . . . . .	69
6.3.1	Requisitos “REQ_F_10” . . . . .	70
6.3.1.1	Requisito “REQ_F_10.1” . . . . .	71
6.3.1.2	Requisito “REQ_F_10.2” . . . . .	74
6.3.1.3	Requisito “REQ_F_10” . . . . .	78
6.3.2	Requisitos “REQ_F_18” . . . . .	80
6.3.2.1	Requisito “REQ_F_18.1 - Atribuição de dois pontos” . . . . .	84
6.3.2.2	Requisito “REQ_F_18.2 - Atribuição de cinco pontos” . . . . .	84
6.3.2.3	Requisito “REQ_F_18.3 - Atribuição de dez pontos” . . . . .	90
6.3.2.4	Requisito “REQ_F_18.4 - Atribuição de vinte pontos” . . . . .	93
6.3.3	Requisitos “REQ_F_19” . . . . .	97
6.3.4	Requisito “REQ_MP_02” . . . . .	97
6.3.4.1	Requisito “REQ_F_10.1” (Versão Móvel) . . . . .	98
6.3.4.2	Requisito “REQ_F_10.2” (Versão Móvel) . . . . .	101
6.3.4.3	Requisito “REQ_F_18” (Versão Móvel) . . . . .	103
6.3.5	Requisito “REQ_F_13” . . . . .	105
6.3.6	Requisito “REQ_F_15” . . . . .	108
<b>7</b>	<b>Testes</b>	<b>111</b>
7.1	Testes Manuais . . . . .	111
7.2	Realização de Testes . . . . .	112
<b>8</b>	<b>Planeamento e Gestão do Projeto</b>	<b>113</b>
8.1	Avaliação dos Riscos . . . . .	113
8.2	Gestão do Projeto . . . . .	115
8.2.1	Calendarização Planeada . . . . .	116
8.2.2	Modificações no Planeamento . . . . .	118
<b>9</b>	<b>Conclusão</b>	<b>123</b>
	<b>Referências</b>	<b>125</b>
	<b>Apêndice A Módulos de Atividades e Recursos do Moodle</b>	<b>131</b>
	<b>Apêndice B Ficheiros Predefinidos Para a Criação de um <i>Plugin</i> de Tipo Bloco</b>	<b>139</b>
	<b>Apêndice C Estrutura e Apresentação dos <i>Plugins</i> Criados e Adicionados</b>	<b>141</b>
	<b>Apêndice D <i>Plugin</i> “<i>Level up! - Gamification</i>” (block/xp) - Lista de Eventos</b>	<b>149</b>
	<b>Apêndice E Tabelas de Testes Manuais Efetuados</b>	<b>155</b>

## Lista de Figuras

1	Diagrama de casos de uso criado para o ator “Administrador” . . . . .	9
2	Diagrama de casos de uso criado para o ator “Professor/Psicólogo” . . . . .	10
3	Diagrama de casos de uso criado para o ator “Encarregado de Educação” . . . . .	11
4	Diagrama de casos de uso criado para o ator “Aluno” . . . . .	12
5	Representação arquitetural do Moodle . . . . .	44
6	Representação dos componentes do Moodle, suas dependências e possível comunicação (baseado na figura 3 existente na documentação referente à apresentação da comunicação efetuada entre componentes do Moodle [19]) . . . . .	44
7	Contextos atribuídos de forma predefinida a um utilizador de papel “Professor (Editor)” - menu de administração . . . . .	47
8	Utilizadores criados no ambiente de desenvolvimento . . . . .	69
9	<i>Plugin</i> “Level up! - Gamification” (block_xp) após ser adicionado à página inicial do administrador . . . . .	81
10	<i>Plugin</i> “Level up! - Gamification” (block_xp) - níveis das condições para atribuição de pontos . . . . .	82
11	<i>Plugin</i> “Level up! - Gamification” (block_xp) - opção de regra para o cumprimento do REQ_F_18.1 . . . . .	84
12	<i>Plugin</i> “Level up! - Gamification” (block_xp) - primeira opção de regras para o cumprimento do REQ_F_18.2 . . . . .	90
13	<i>Plugin</i> “Level up! - Gamification” (block_xp) - segunda opção de regras para o cumprimento do REQ_F_18.2 - segunda opção . . . . .	90
14	<i>Plugin</i> “Level up! - Gamification” (block_xp) - opção de regras para o cumprimento do REQ_F_18.3 . . . . .	92
15	<i>Plugin</i> “Level up! - Gamification” (block_xp) - primeira opção de regra para o cumprimento do REQ_F_18.4 . . . . .	93
16	<i>Plugin</i> “Level up! - Gamification” (block_xp) - segunda opção de regras para o cumprimento do REQ_F_18.4 . . . . .	94
17	<i>Plugin</i> “Attendance” (mod_attendance) - predefinição da atribuição de “grades” .	95
18	<i>Plugin</i> “Attendance” (mod_attendance) - alteração da atribuição de “grades” para adicionar pontuação ao <i>plugin</i> “Level up! - Gamification” (block_xp) . . . . .	95
19	Templates para o tipo de <i>plugins</i> selecionado (baseado na figura existente na documentação referente ao suporte de <i>plugins</i> pela aplicação móvel do Moodle [48])	98
20	Envio de SMSs através do Twilio . . . . .	106
21	Receção de SMSs através do Twilio . . . . .	106
22	Primeiro passo da configuração do <i>plugin</i> . . . . .	107
23	Segundo passo da configuração do <i>plugin</i> . . . . .	107
24	Terceiro passo da configuração do <i>plugin</i> . . . . .	108
25	<i>Plugin</i> do tipo Bloco “HTML” após as configurações efetuadas - perfil de um professor/psicólogo . . . . .	109
26	Primeiro Semestre - planeamento estipulado . . . . .	117
27	Segundo Semestre - planeamento estipulado . . . . .	118
28	Primeiro Semestre - planeamento real . . . . .	119
29	Segundo Semestre - planeamento real . . . . .	122
C1	<i>Plugin</i> “Diário do Humor” (block_daily_activities) no perfil do aluno - versão web	142
C2	<i>Plugin</i> “Diário do Humor” (block_daily_activities) no perfil do aluno - versão <i>mobile</i>	142
C3	Apresentação do Bloco “Diário de Atividades” - Primeira Versão . . . . .	143

C4	Versão Web - Apresentação do Bloco “ <i>Diário de Atividades</i> ” no Perfil do Aluno .	145
C5	Versão <i>Mobile</i> - Apresentação do Bloco “ <i>Diário de Atividades</i> ” no Perfil do Aluno	145
C6	Apresentação do Bloco “Gráfico de Humor e Atividades” . . . . .	146
C7	<i>Plugin</i> “ <i>Level up! - Gamification</i> ” (block_xp) - apresentação do <i>plugin</i> na página inicial do aluno após a customização efetuada (versão web) . . . . .	147
C8	<i>Plugin</i> “ <i>Level up! - Gamification</i> ” (block_xp) - apresentação da tabela de crachás do <i>plugin</i> na página inicial do aluno após a customização efetuada (versão web) .	147
C9	<i>Plugin</i> “ <i>Level up! - Gamification</i> ” (block_xp) - apresentação na página inicial do aluno (versão <i>mobile</i> ) . . . . .	147

## Lista de Tabelas

1	Lista final de requisitos . . . . .	31
2	Lista final de requisitos - continuação . . . . .	32
3	Comparação dos LMSs selecionados . . . . .	37
4	Comparação do suporte aos requisitos entre as plataformas Moodle e Forma LMS	38
5	Comparação final entre as plataformas Moodle e FormaLMS - parâmetros utilizados para a seleção final . . . . .	40
6	Suporte aos requisitos pelo Moodle . . . . .	41
7	Apresentação de exemplos da convenção da nomenclatura adotada ( <i>Frankenstyle</i> ) para identificar <i>plugins</i> do Moodle . . . . .	52
8	Apresentação de exemplos da convenção da nomenclatura adotada ( <i>Frankenstyle</i> ) para identificar subsistemas do Moodle . . . . .	52
9	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Módulo de Atividades e Recursos” . . . . .	52
10	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Bloco” . . . . .	53
11	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Tema” . . . . .	53
12	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Formato de Curso” . . . . .	53
13	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Inscrição” . . . . .	53
14	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Autenticação” . . . . .	54
15	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Repositório” . . . . .	54
16	<i>Frankenstyle</i> de <i>plugins</i> do tipo “Filtros de Texto” . . . . .	54
17	Requisitos existentes e mapeamento com as funcionalidades do Moodle . . . . .	64
18	Requisitos existentes e mapeamento com as funcionalidades do Moodle - continuação	65
19	Estratégias adotadas nos requisitos a desenvolver . . . . .	69
20	Identificação <i>Frankenstyle - plugin</i> “daily_mood” . . . . .	71
21	Identificação <i>Frankenstyle - plugin</i> “daily_activities” . . . . .	74
22	Identificação <i>Frankenstyle - plugin</i> “mood_chart” . . . . .	78
23	Identificação <i>Frankenstyle - plugin</i> “block_xp” . . . . .	80
24	Identificação <i>Frankenstyle - plugin</i> “mod_videotime” . . . . .	80
25	Identificação <i>Frankenstyle - plugin</i> “mood_hvp” . . . . .	80
26	Identificação <i>Frankenstyle - plugin</i> “mood_attendance” . . . . .	80
27	Identificação <i>Frankenstyle - plugin</i> “block_sms” . . . . .	105
28	Tabela de preços praticados em Portugal para envio e receção de mensagens através do <i>Twilio</i> . . . . .	107
29	Identificação <i>Frankenstyle - plugin</i> “block_html” . . . . .	108
30	Teste T_01 . . . . .	112
31	Tabela de riscos e respetiva estratégia de mitigação . . . . .	114
D1	Eventos do <i>plugin</i> “ <i>Level up! - Gamification</i> ” (block/xp) - contexto de “Sistema”	149
D2	Eventos do <i>plugin</i> “ <i>Level up! - Gamification</i> ” (block/xp) - contexto de contexto de “Módulo de Atividades e Recursos” (Atividades) . . . . .	149
D3	Eventos do <i>plugin</i> “ <i>Level up! - Gamification</i> ” (block/xp) - contexto de contexto de “Módulo de Atividades e Recursos” (Recursos) . . . . .	153
E1	Teste T_01 . . . . .	155
E2	Teste T_02 . . . . .	156
E3	Teste T_03 . . . . .	157
E4	Teste T_04 . . . . .	157
E5	Teste T_05 . . . . .	158
E6	Teste T_06 . . . . .	158

E7	Teste T_07 . . . . .	159
E8	Teste T_08 . . . . .	159
E9	Teste T_09 . . . . .	160
E10	Teste T_10 . . . . .	160
E11	Teste T_11 . . . . .	161
E12	Teste T_12 . . . . .	161
E13	Teste T_13 . . . . .	162
E14	Teste T_14 . . . . .	162
E15	Teste T_15 . . . . .	163
E16	Teste T_16 . . . . .	163
E17	Teste T_17 . . . . .	164
E18	Teste T_18 . . . . .	164
E19	Teste T_19 . . . . .	164
E20	Teste T_20 . . . . .	165
E21	Teste T_21 . . . . .	166
E22	Teste T_22 . . . . .	167
E23	Teste T_23 . . . . .	168
E24	Teste T_24 . . . . .	168
E25	Teste T_25 . . . . .	169
E26	Teste T_26 . . . . .	169
E27	Teste T_27 . . . . .	170
E28	Teste T_28 . . . . .	170
E29	Teste T_28 . . . . .	171
E30	Teste T_29 . . . . .	171
E31	Teste T_30 . . . . .	172
E32	Teste T_31 . . . . .	172
E33	Teste T_32 . . . . .	173



## Acrónimos

- AICC** *Aviation Industry Computer-Based Training Committee* - Comité de Treino por Computador da Indústria da Aviação
- API** *Application Programming Interface* - Interface de Programação de Aplicações
- BLL** *Business Logic Layer* - Camada de Negócio/Lógica
- CDI** *Children's Depression Inventory* - Inventário de Depressão Infantil
- CMS** *Content Management Systems* - Sistemas de Gestão de Conteúdos
- DAL** *Data Access Layer* - Camada de Acesso aos Dados
- DEI** Departamento de Engenharia Informática
- DDL** *Data-driven Learning* - Aprendizagem Orientada a Dados
- DDNS** *Dynamic Domain Name System* - Sistema de Nomes de Domínio Dinâmico
- EPS** Escolas Promotoras de Saúde
- ES** Engenharia de Software
- FCTUC** Faculdade de Ciências e Tecnologia da Universidade de Coimbra
- FPCE** Faculdade de Psicologia e Ciências da Educação
- GDPR** *General Data Protection Regulation* - Regulamento Geral sobre a Proteção de Dados
- HTML** *HyperText Markup Language* - Linguagem de Marcação de Hipertexto
- HTTP** *Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto
- HTTPS** *Hypertext Transfer Protocol Secure* - Protocolo de Transferência de Hipertexto Seguro
- H5P** *HTML 5 Package* - Pacote HTML 5
- IIS** *Internet Information Services* - *Serviços de Informação da Internet*
- ILT** *Instructor-led Training* - Treino Conduzido por um Instrutor
- IMS** *Instructional Management System* - Sistema de Gestão Instrucional
- JSON** *JavaScript Object Notation* - Notação de Objetos JavaScript
- LAMP** Linux, Apache, MySQL/MariaDB, PHP
- LMS** *Learning Management Systems* - Sistemas de Gestão de Aprendizagem
- LTI** *Learning Tools Interoperability* - Interoperabilidade de Ferramentas de Aprendizagem
- MEI** Mestrado em Engenharia Informática
- Moodle** *Modular Object-Oriented Dynamic Learning Environment* - Ambiente de Aprendizagem Dinâmico Orientado a Objetos Modular e Dinâmico
- OSS** *Open-Source Software* - Software de Código Aberto
- PC** *Personal Computer* - Computador Pessoal
- REST** *Representational State Transfer* - Transferência Representacional de Estado

**RDBMS** *Relational Database Management System* - Sistema de Gestão de Base de Dados Relacional

**SCORM** *Sharable Content Object Reference Model* - Modelo de Referência de Objeto de Conteúdo Compartilhável

**SGBD** Sistemas de Gestão de Bases de Dados

**SGSIIC** Serviço de Gestão de Sistemas e Infraestruturas de Informação e Comunicação

**SMS** Saúde, Mente, Sucesso

**UC** Universidade de Coimbra

**UI** *User Interfacen Layer* - Camada de Apresentação/ Interface do Utilizador

**URL** *Uniform Resource Locator* - Localizador Uniforme de Recursos

**VM** *Virtual Machine* - Máquina Virtual

**XMLDB** *Extensible Markup Language Database* - Base de Dados de Linguagem de Marcação Extensível

# 1 Introdução

O presente capítulo tem como intuito expor as motivações que conduziram à existência deste projeto, os seus objetivos, desafios, contribuição e como a apresentação do trabalho realizado será estruturada nos capítulos seguintes.

## 1.1 Motivação

A literatura mais recente evidencia que os adolescentes, no seu dia-a-dia, enfrentam inúmeros desafios e pressões significativas, que acabam por influenciar a sua saúde mental, podendo impelir ao desenvolvimento de doenças como a depressão e, em casos mais extremos, levar ao suicídio. Deste modo, a promoção da saúde mental é considerada como uma matéria de urgente intervenção.

Investigadores da FPCE da UC, defrontando-se com a importância desta questão, empenharam-se em explorar medidas que a solucionassem, contribuindo para a formação de gerações futuras felizes e saudáveis.

Sustentando-se em evidências, concluíram que implementando ações precoces (através da melhoria na literacia em saúde mental, fomentando o desenvolvimento de competências de auto cuidado e atitudes positivas face a dificuldades e criando uma comunidade de entretajuda e confiança pelo envolvimento das respetivas escolas), seriam garantidos melhores resultados de bem-estar e saúde mental, solucionando distúrbios específicos como a depressão, como anteriormente referido.

Assim surgiu o Projeto SMS. Num âmbito global, este projeto representa uma iniciativa que teve origem na FPCE da UC, financiado pelo Estado Português, sendo uma intervenção multinível, longitudinal e baseada em evidências. É destinada a promover a saúde mental em geral (dos adolescentes nas escolas, respetivas famílias e comunidades) e a prevenir a depressão, em particular. Para tal, o programa concebido para o projeto segue diretrizes nacionais e internacionais, tendo sido aprovado pela Comissão de Ética da FPCE da UC.

A intervenção terá lugar em duas escolas do município da Figueira da Foz, integrando 460 alunos do 3º ciclo, comunidade discente, respetivos encarregados de educação, docentes e não docentes. Estas escolas irão, portanto, ser a estrutura comum para a união dos setores “Saúde e Educação”, funcionando como disseminadoras de saúde mental, em conformidade com os padrões globais para Escolas Promotoras de Saúde (EPS).

O projeto SMS baseia a parte fundamental da sua intervenção na aplicação de programas (já testados com este público-alvo): os programas “SMSjovens” (destinado aos alunos) e “SMSeducadores” (destinado aos respetivos encarregados de educação). Estes, para além dos professores e psicólogos que utilizarão a plataforma para aplicar os programas, serão os principais destinatários deste projeto. Os programas consistem em intervenções presenciais devidamente estruturadas e divididas por sessões. Cada sessão é organizada em função de um tema geral, guiado por objetivos, sendo realizadas atividades individuais, ou em grupos (como jogos, exercícios de análise, reflexão e debate). Os programas serão conduzidos por psicólogos/profissionais de saúde mental e professores, desde que previamente capacitados para o efeito.

Contudo, tratando-se o público alvo desta intervenção, de jovens, maioritariamente, os elementos responsáveis por este projeto sentiram a necessidade de possuir uma plataforma digital que:

1. Servisse de suporte à aplicação dos programas deste projeto nas sessões presenciais, aos quais os alunos e respetivos encarregados de educação teriam acesso;
2. Aumentasse a adesão ao mesmo por parte dos alunos;

3. Possibilitasse a criação de uma comunidade virtual, através de ferramentas potenciadoras de comunicação, como a existência de *chats* e ligação a redes sociais;
4. Integrasse o “Fórum Resiliência”, componente deste programa, onde tanto alunos, como encarregados de educação poderiam expor as suas histórias e testemunhos pessoais;
5. Proporcionasse o envio de mensagens curtas (*short message service*- SMSs) aos utilizadores com frases inspiradoras e motivacionais;
6. Que adotasse uma perspetiva de gamificação para, igualmente, aumentar a motivação dos jovens na utilização da plataforma;
7. Num melhor cenário, ainda possibilitar a existência de uma aplicação móvel.

## 1.2 Objetivos do Estágio

O objetivo primordial do estágio passa, portanto, pela apresentação de uma plataforma web que reúna as funcionalidades desejadas pelos elementos responsáveis pelo projeto (investigadores da FPCE) que, portanto, são o cliente. Assim, uma plataforma teria de ser instalada, configurada e lançada. Tencionou-se que esta plataforma estivesse alojada num servidor remoto (servidor *datacenter* da UC, DEI ou *cloud*) pelo que se necessitou de contactar as entidades competentes (Diretor do Serviço de Gestão de Sistemas e Infraestruturas de Informação e Comunicação (SGSIIC) da UC e *Helpdesk* do DEI), de modo a que uma máquina virtual fosse concedida.

Em primeiro lugar, face os requisitos da plataforma teve-se de equacionar as vantagens do desenvolvimento de uma plataforma de raiz, ou da adoção de um *Open-Source Software* - Software de Código Aberto (OSS).

Dada a natureza do projeto, considerou-se pertinente adotar um LMS. Através da comparação de diversas opções existentes, escolheu-se o *Modular Object-Oriented Dynamic Learning Environment* - Ambiente de Aprendizagem Dinâmico Orientado a Objetos Modular e Dinâmico (Moodle). Esta decisão revelou-se a mais significativa de todo o projeto, uma vez que influenciou e tornou claro o âmbito do mesmo. O presente projeto, apesar da finalidade de apresentar uma plataforma web que suportasse os requisitos desejados pelo cliente, motivou a que o verdadeiro objeto de estudo deste projeto fosse o Moodle - a sua arquitetura, *Application Programming Interface* - Interface de Programação de Aplicações (API) e das múltiplas estratégias que poderiam ser selecionadas para a integração de novas funcionalidades ou integração de recursos já existentes.

Considerando então os objetivos, a realização do projeto foi dividida nas seguintes etapas:

### 1. Levantamento dos Requisitos

Com o intuito de se perceber na totalidade as motivações e ambições para a plataforma SMS, houve necessidade de:

- Compreender as motivações do Projeto SMS;
- Realizar o levantamento dos requisitos da plataforma em conjunto com o cliente (sob a forma de *User Stories*);
- Atribuir a devida prioridade (através do método de MoSCoW);
- Afinar os requisitos, na medida em que a ideia por detrás de muitos dos requisitos desejados pelo cliente, se encontrava exposta de uma forma muito pouco concreta e explícita. Um exemplo passou pelos requisitos de gamificação: o cliente sabia que queria ver integrada “gamificação”, mas não sabia de que forma é que esta poderia ser integrada.

## 2. Avaliação de OSSs existentes no mercado

Visto que, atualmente, se identificam maiores as vantagens de utilizar um OSS do que as de construir um software raíz, tomou-se a decisão de utilizar este tipo de *softwares*. Para tal, procedeu-se à avaliação das ofertas existentes no mercado: quais as funcionalidades que disponibilizavam e de que forma eram compatíveis com as desejadas para o projeto SMS.

## 3. Seleção de uma Plataforma

Após a avaliação realizada, selecionou-se a plataforma a adotar. Essa seleção foi realizada considerando as funcionalidades desejadas em comparação com as oferecidas pelas diferentes plataformas.

## 4. Instalação da plataforma no servidor da UC e, posteriormente no servidor do DEI

A plataforma SMS-eSaúde começou por estar alojada no servidor remoto da UC, sendo posteriormente transferida para o servidor do DEI. Em ambos os casos, foi necessário solicitar a disponibilidade das máquinas virtuais às entidades competentes.

## 5. Identificação das Funcionalidades Adicionais

Tendo em conta a plataforma selecionada, identificaram-se os requisitos não suportados para que, posteriormente, pudessem ser implementados. Assim, as etapas constituintes desta fase passaram por:

- Analisar e estudar as funcionalidades inerentes à plataforma selecionada;
- Mapear essas funcionalidades com os requisitos do produto final;
- Identificar as funcionalidades adicionais que seria necessário desenvolver.

## 6. Configuração e Implementação das Funcionalidades Adicionais

Tendo-se identificado os requisitos não suportados, passou-se ao processo de desenvolvimento dos mesmos, identificando-se, tendo em conta o OSS selecionado, a forma como tal poderia ser feito:

- Estudo da arquitetura e das possibilidades de integração de novas funcionalidades na plataforma selecionada:
  - Funcionamento do Moodle;
  - Os tipos de *plugins* e as suas dinâmicas;
  - Como desenvolver os tipos de *plugins* de acordo com os requisitos;
  - Como desenvolver versões *mobile* desses *plugins*;
  - Quando e como utilizar as suas mais diversas APIs (API de Manipulação de Dados do Moodle; API de *Output* do Moodle; API de Formulários do Moodle; API para Apresentação de Gráficos do Moodle; API de Conclusão de uma Atividade do Moodle - por exemplo).
- Avaliação da essência do requisito e seleção dos métodos para a sua integração:
  - **Estratégia 1:** Desenvolvimento de raíz;
  - **Estratégia 2:** Exploração, reutilização e customização de recursos já existentes.

### 1.3 Desafios

Fora diversos contratemplos que surgiram ao longo do processo de desenvolvimento do projeto (enunciados no capítulo 8, de gestão do projeto) apresenta-se de imediato os seus inúmeros desafios.

Trabalhando sobre o Moodle, uma plataforma com dezanove anos de desenvolvimento, que conta com mais de mil setecentos e catorze *plugins* criados e disponibilizados pelo mesmo (até ao momento), a probabilidade de existência de *plugins* que já solucionassem os requisitos identificados como “funcionalidades adicionais”, revelou-se imensa.

Em engenharia, torna-se inútil implementar, criar e refazer soluções se elas já existirem e forem eficientes. Assim, um desafio no desenvolvimento de produtos de software é saber reutilizar recursos já existentes e de os saber integrar. Utilizou-se este pensamento em duas situações:

1. Se optou pela adoção de um OSS (de forma mais concreta o LMS Moodle);
2. Se adotou soluções já existentes que possibilitassem a incorporação das inúmeras funcionalidades desejadas.

Relativamente à situação 2, a utilização de ferramentas já existentes foi realizada através da incorporação de *plugins*. Estes, na sua maioria, compreendem inúmeros anos de investimento e são desenvolvidos por muitos *developers*, o que, por um lado, dificultou a análise do seu código, e por outro, exigiu a sua adaptação, de acordo com os requisitos e com o próprio Moodle.

Nas situações em que houve necessidade de implementar um *plugin* de raiz, foi necessário saber integrá-lo corretamente com o LMS.

O desafio central assentou então nestas questões: o benefício de se adotar uma plataforma já existente com inúmeras das funcionalidades desejadas já suportadas, revelou-se igualmente um desafio na medida em que, tudo o que necessitasse de ser adicionalmente implementado, teria de ser realizado em conformidade e integridade com a plataforma, o que associou alguma complexidade (mesmo quando as tarefas, implementadas num outro ambiente e desintegrada a plataforma, fossem relativamente fáceis ou pouco complexas de conseguir).

Um exemplo de uma das dificuldades resultantes da integração com o Moodle, passou pela sua documentação. Aquando do seu estudo e integração das devidas APIs (accedidas através da documentação para *developers* do mesmo), essa documentação, recorrentemente, não se revelou suficiente. Apesar de o Moodle ser um *software* extremamente utilizado, para além de a sua documentação se encontrar desatualizada, era pobre em exemplos. A forma como se solucionaram muitos problemas que foram surgindo, foi literalmente através da aprendizagem “tentativa - erro”, o que atrasou o desenvolvimento e integração dos *plugins*. Alguns desses casos passaram por pormenores que tinham de ser levados em consideração como: pastas e ficheiros constituintes de um *plugin*, que se não tivessem um determinado nome, ou não se localizassem em diretorias específicas, eram motivos suficientes para um *plugin* não funcionar; passos que deviam ser incluídos, que não se encontravam especificados na documentação e que apenas eram descobertos pela análise de outros *plugins* existentes na diretoria de *plugins* do Moodle; pormenores como, na estruturação e definição dos *web services* e suas funções, da versão *mobile* dos *plugins* desenvolvidos, ser necessário definir o *array* de funções antes do *array* de serviços. Acrescenta-se a isto o facto de não se ter qualquer experiência prévia no Moodle, ou em qualquer outro tipo de LMS existente.

Contudo, embora ocupando muito mais tempo do que o previsto para a implementação de cada requisito, as dificuldades que foram surgindo foram eventualmente ultrapassadas.

Ademais, o facto de o presente projeto ser resultante das necessidades de um cliente real, adicionou-se o desafio de saber lidar com o mesmo: saber ouvir; saber adequar o discurso (uma

vez que Psicologia e Informática têm, evidentemente, linguagens e culturas diferentes); saber gerir as suas expectativas; encontrar sempre a melhor solução possível de acordo com os seus desejos; ter capacidade de adaptação quando este deseja fazer alterações.

Assim, em suma, enumeram-se os principais desafios do projeto:

1. Ter capacidade de saber reutilizar e integrar recursos já existentes;
2. Saber criar recursos novos e ter, igualmente, capacidade de os integrar;
3. Saber aplicar inúmeras tecnologias em conjunto (onde, até à data não havia qualquer experiência) e sempre em conformidade com as APIs do Moodle (PHP e Ionic Angular, por exemplo);
4. Saber lidar com um cliente real.

## 1.4 Contribuição

A contribuição oferecida por este projeto passa pelo facto de se apresentar um produto desejado por clientes reais (investigadores da FPCE da UC), para dar suporte a um projeto de grande dimensão (para já, a nível local, posteriormente a nível nacional), o Projeto SMS, que envolverá tanto escolas, como respetivos alunos, encarregados de educação, professores e psicólogos, numa intervenção de âmbito tão nobre como a promoção da saúde mental. O produto apresentado será utilizado tanto durante as sessões presenciais efetuadas e planeadas para a intervenção, como, futuramente, e de forma independente, pelos alunos e encarregados de educação do programa.

Por outro lado, uma vez que foram integradas várias ferramentas existentes em conjunto, para além da criação de novas, o trabalho realizado pode vir a ser utilizado por indivíduos de projetos com requisitos similares, tanto em web, como na aplicação móvel.

Inclusivamente, graças ao presente projeto foi possibilitada a existência de novos projetos futuros, focados no design e aspeto visual da plataforma.

## 1.5 Apresentação da Estrutura do Documento

Considerando os objetivos do estágio e, tendo em consideração que o presente relatório tem como propósito apresentar o trabalho realizado com o intuito de ver cumpridos esses objetivos, apresenta-se assim a sua estrutura e organização.

Em primeiro lugar, no capítulo 2 apresenta-se o levantamento e documentação dos requisitos do projeto.

No capítulo 3, procede-se à exploração dos OSSs existentes e seleciona-se a plataforma a adotar.

De seguida, no capítulo 4 revela-se a arquitetura, funcionamento e *design* da plataforma selecionada.

Posteriormente, no capítulo 5, identificam-se as funcionalidades a adicionar.

No capítulo 6, é exposta a forma como essas funcionalidades serão implementadas.

No capítulo 7 apresentam-se os testes efetuados.

No capítulo 8 apresentam-se o planeamento e a gestão do projeto.

Por fim, no capítulo 9, correspondente à conclusão, expõem-se as considerações finais sobre o trabalho desenvolvido.

*Esta página foi intencionalmente deixada em branco.*



## 2 Levantamento de Requisitos

Segundo a norma IEEE Std 1233-1996 [1], um requisito define-se como:

- uma condição ou capacidade necessária ao utilizador para que possa solucionar um problema ou atingir um certo objetivo;
- uma condição ou capacidade que deve ser satisfeita ou possuída por um sistema, ou componente de sistema, de maneira a satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto;
- uma representação documentada de uma condição ou capacidade como apresentado nas duas definições em cima.

De maneira a evitar futuras falhas ou o próprio insucesso no processo de desenvolvimento e, conseqüentemente, do próprio produto, procedeu-se à identificação e documentação dos requisitos em conjunto com o cliente (os investigadores da FPCE da UC, criadores do Projeto SMS).

Quanto mais cedo se proceder ao levantamento dos requisitos, maior será a probabilidade de o produto final apresentar uma boa qualidade - que vá ao encontro das expectativas e ambições dos *stakeholders*. Ademais, a redução dos custos inerentes ao desenvolvimento é promovida por se reduzir a probabilidade de se realizar ajustes e alterações numa fase posterior (e, portanto, mais crítica) do desenvolvimento.

Esta definição estabelece a base de acordo entre o *developer* e o cliente, fornecendo uma referência para a validação do produto final. Foi, portanto, necessário compreender na totalidade a visão e motivação do cliente para, para além de se identificar os requisitos, se compreender a importância atribuída a cada um (prioridade).

Assim sendo, o presente capítulo tem como propósito expor, de forma clara, a interpretação da imagem/visão da plataforma (*input*), de maneira a tornar possível a estruturação dos requisitos e respetivas prioridades (*output*).

É de notar que esses mesmos requisitos representam itens de alto nível, usados para selecionar o OSS a adotar.

### 2.1 Análise do Problema

Primeiramente considera-se necessário identificar os *stakeholders* do projeto, as características gerais do mesmo e a lista de atores do sistema.

#### ***Stakeholders***

Os *stakeholders* do problema identificado são, então:

- Cliente (investigadores da FPCE, criadores do Projeto SMS);
- Psicólogos das escolas onde irá ser realizada a intervenção;
- Professores das escolas intervencionadas;
- Alunos do 3<sup>o</sup> ciclo das respetivas escolas;
- Encarregados de Educação dos alunos;
- Equipa de desenvolvimento (a autora da Tese de Mestrado).

### Características Gerais

De uma forma geral, percebeu-se que o produto desejado pelo cliente passava por uma plataforma digital gratuita. Pretendia-se que:

- A plataforma fosse acessada por alunos, encarregados de educação, professores e psicólogos.
- Todos estes teriam as suas contas, sendo o acesso realizado através de *login*.
- O registo autónomo na plataforma por parte dos utilizadores não fosse necessário. Tratando-se de uma plataforma de cariz educativo, com utilizadores específicos e, por questões de segurança, de modo a garantir que apenas esses tinham acesso à plataforma, as suas contas seriam previamente criadas pelo administrador da mesma.
- A plataforma garantisse usabilidade, ou seja, que fosse eficaz, eficiente e assegurasse a satisfação do utilizador:
  - Um site eficaz para além de atender aos objetivos do negócio, possibilita que os utilizadores alcancem os seus objetivos no site.
  - Um site eficiente fornece respostas rapidamente, segue uma sequência lógica, não desperdiça recursos e requer menos tempo para a gestão de conteúdos.
  - Um utilizador satisfeito atinge o seu objetivo e aprecia a sua experiência.

Algo primordial para a garantia destas questões, passa pelo servidor onde o site se encontra alojado. Assim, o servidor deve ser confiável, assegurando uma melhor experiência do utilizador.

- Os professores e psicólogos inserissem os conteúdos na plataforma (*upload* de conteúdos ou criação dos mesmos através de ferramentas existentes no próprio site), para estes serem aplicados pelos mesmos nas sessões presenciais do Projeto SMS.
- Os conteúdos a ser carregados para a plataforma consistissem em:
  - Ficheiros de vídeo nos formatos .mov, .mp4 ou diretamente de sites para partilha de vídeo como o *Vimeo*, ou o *Youtube* (mas a serem acessados na própria plataforma);
  - Ficheiros de áudio (mp3, .aac, .flac, .m4a, .oga, .ogg, .wav);
  - Ficheiros de texto (.txt, .doc, .docx, .pdf).
- Os conteúdos fossem colocados a par com as sessões presenciais, que teriam lugar nas respetivas escolas dos alunos.
- Os conteúdos fossem acessados e utilizados pelos alunos de cada turma, diretamente na plataforma (sem necessidade de serem reencaminhados para outro “local”).
- O aluno só tivesse acesso a um dado conteúdo, após aceder ao conteúdo imediatamente anterior.
- A alguns professores/psicólogos fossem delegadas certas funcionalidades de administração da plataforma, tal como a gestão de alunos e cursos.
- A página principal (*landing page*) tivesse conteúdo geral para utilizadores não autenticados, havendo conteúdo privilegiado no caso contrário (isto é, para alunos e respetivos encarregados de educação, após realizarem o *login*).
- Houvesse conteúdo específico e direcionado para cada grupo (alunos e encarregados de educação).
- A plataforma tivesse recursos aplicáveis na gamificação.

- Ambos os grupos (alunos e encarregados de educação) recebessem notificações aquando da inserção na plataforma de novos conteúdos, ou de quaisquer outras necessidades informativas.
- A plataforma suportasse a existência de fóruns de discussão e *chats* de mensagens. O fórum (“Fórum Resiliência”) possibilitaria que os utilizadores escrevessem os seus testemunhos e opiniões pessoais. Exigia-se, contudo, que estes testemunhos tivessem aprovação (ou não) prévia dos Administradores, para poderem ficar visíveis/serem publicados na plataforma, prevenindo a publicação de conteúdos maliciosos/que não sigam o propósito da página. Quanto aos *chats*, estes cumpririam o propósito de responder a questões em tempo real, contando com a presença de profissionais de saúde mental.
- Os utilizadores recebessem nos seus telemóveis, regularmente, SMSs (*short message service*) com textos ou frases inspiradoras e motivacionais.
- Houvesse, se possível, ligação a redes sociais apenas através da partilha dos conteúdos da plataforma, nessas redes, por exemplo, não incluindo o *login* através dessas redes.
- Houvesse uma extensão da plataforma para telemóvel, através dos *browsers* utilizados pelos alunos, ou, até mesmo, de uma aplicação móvel.

### Atores

Face a descrição do problema referido, foram assim identificados os atores responsáveis pela interação com o sistema, direta ou indiretamente:

- **Administrador** - responsável pelas configurações gerais do sistema.
- **Professor e Psicólogo** - responsáveis por inserir e gerir os conteúdos na plataforma; adicionar e remover cursos; atribuir cursos aos alunos e encarregados de educação.
- **Aluno e Encarregado de Educação** - responsáveis por participar nos cursos e consumir os conteúdos a eles destinados.
- **Utilizador Não Autenticado** - estado de todos os utilizadores, antes de se efetuar o *login* na página; tipo de utilizador que apenas tem acesso a conteúdo disponibilizado de forma pública na *landing page*.

Considerando os atores e identificadas as características gerais da plataforma, é possível apresentar os diagramas de casos de uso (forma primária de representar os requisitos de um sistema), através da especificação dos relacionamentos entre os atores e a forma como interagem com o sistema. Assim, apresentam-se, de seguida, os diagramas de casos de uso criados para cada ator (figuras 1, 2, 3 e 4).



Figura 1: Diagrama de casos de uso criado para o ator “Administrador”

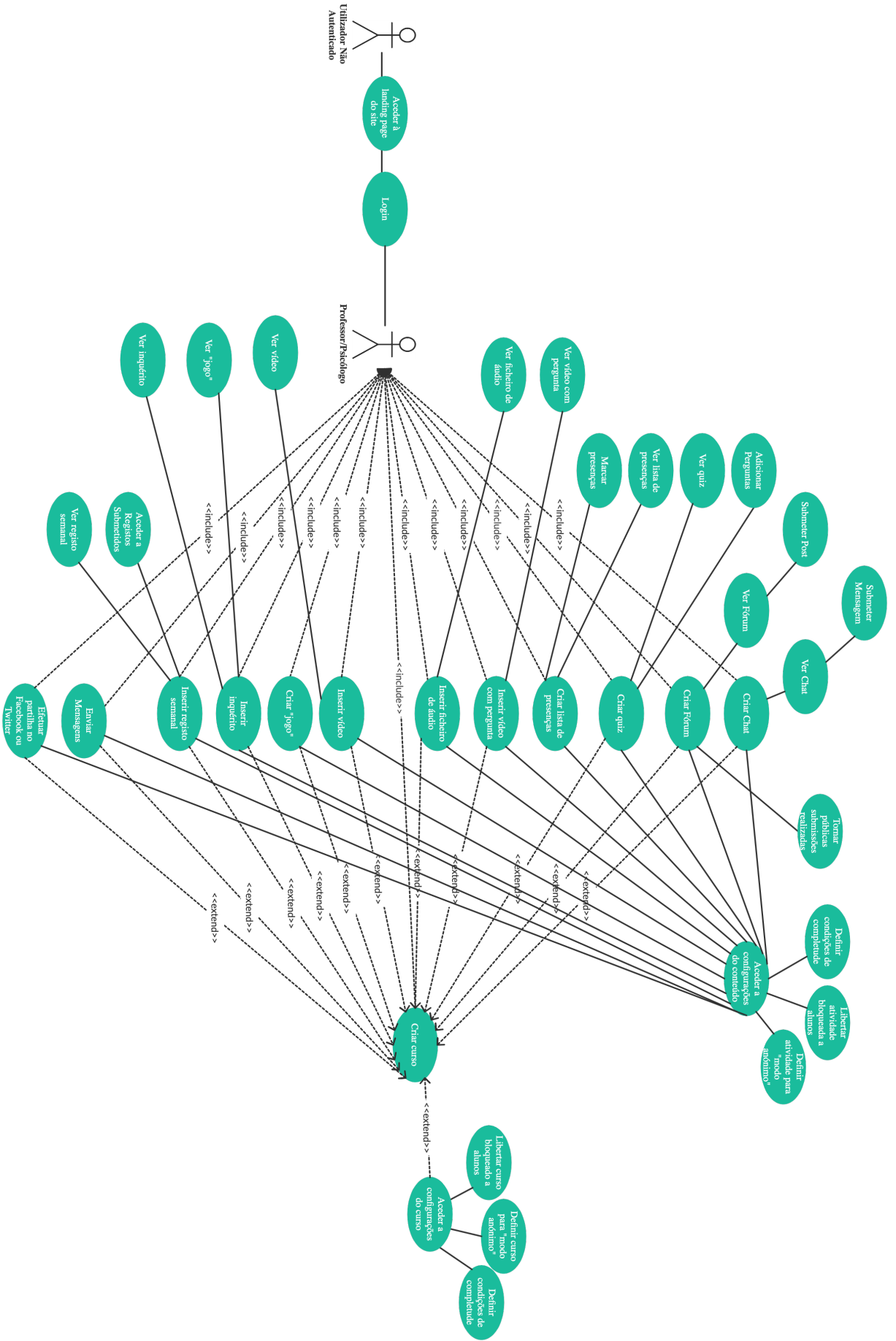


Figura 2: Diagrama de casos de uso criado para o ator “Professor/Psicólogo”

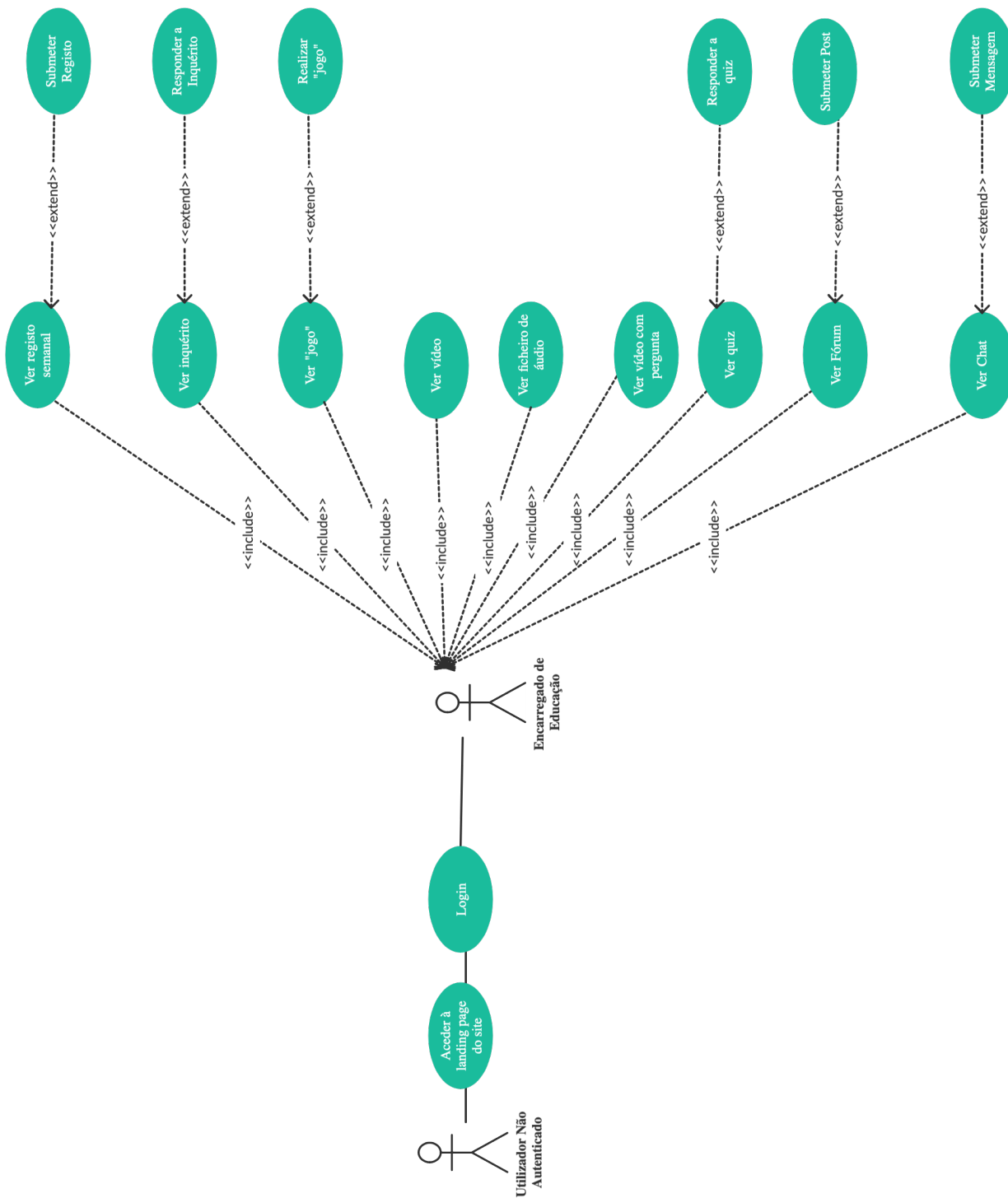


Figura 3: Diagrama de casos de uso criado para o ator “Encarregado de Educação”

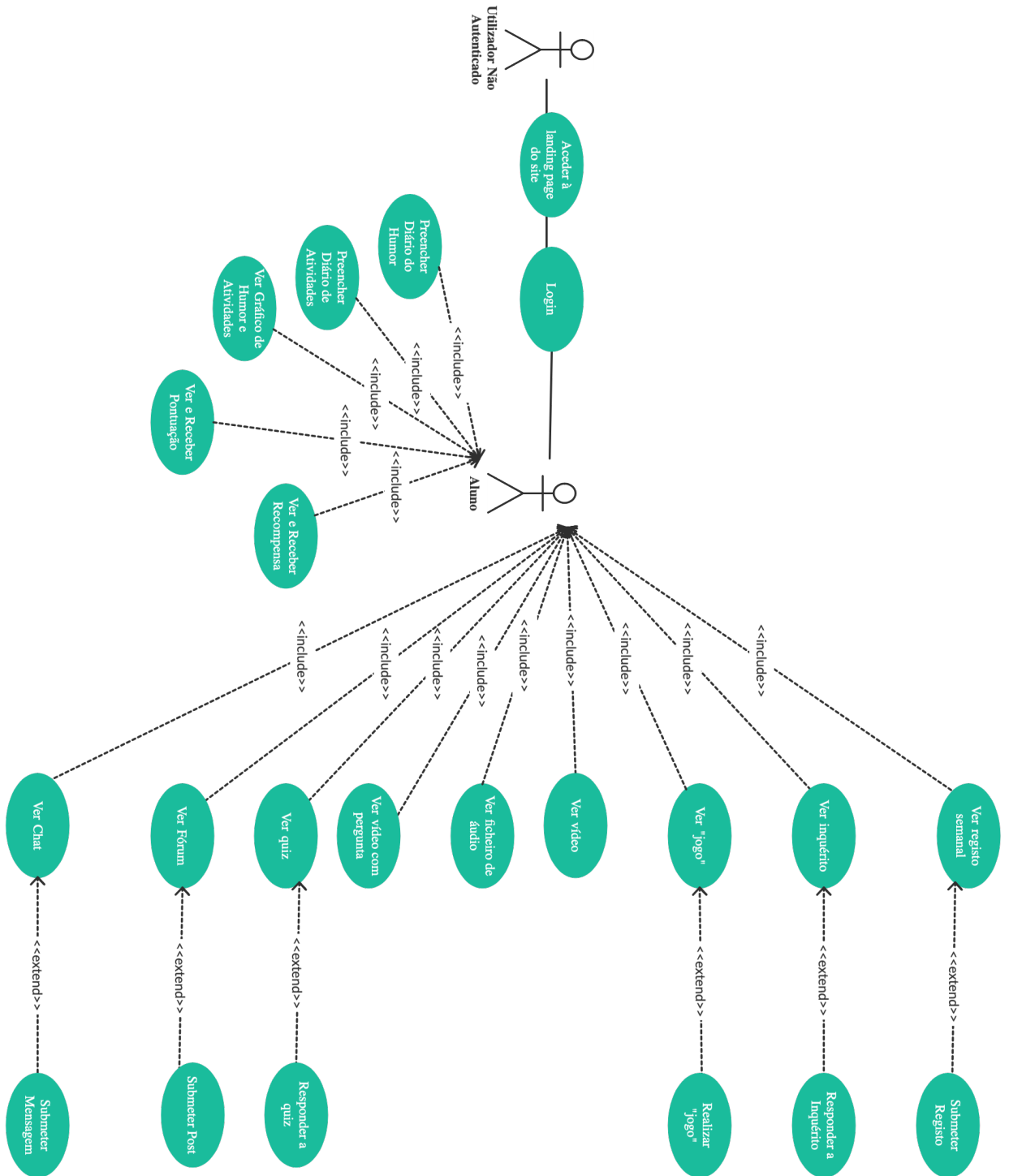


Figura 4: Diagrama de casos de uso criado para o ator "Aluno"

## 2.2 Documentação de Requisitos

Através da análise da informação recolhida e descrita na secção anterior, definiu-se assim a primeira lista dos requisitos e respetiva prioridade. O método de priorização de requisitos utilizado foi o método *MoSCoW* [2], bastante útil na identificação das tarefas críticas para o sucesso de um projeto, assim como para perceber aquelas que poderão ser descartadas, caso seja necessário (como por exemplo, em casos onde haja falta de recursos, como o tempo para a execução de um projeto).

Segundo o método *MoSCoW*, a prioridade de um requisito pode ser caracterizada da seguinte forma:

- **MUST (M)** - Requisitos de prioridade mais elevada. São requisitos obrigatórios e essenciais para o sucesso de um projeto, influenciando o prazo da sua entrega. No caso de um requisito com prioridade *MUST (M)* não estar incluído e/ou concluído, a entrega do projeto não pode ser considerada.
- **SHOULD (S)** - Requisitos que devem ser considerados ao máximo, mas que não impactam o sucesso do projeto. Embora os requisitos com esta prioridade possam, por vezes, ser tão importantes como requisitos de prioridade *MUST (M)*, não se revelam tão críticos a nível de tempo, uma vez que, podem ser interrompidos até um futuro prazo de entrega.
- **COULD (C)** - Requisitos desejáveis, mas que não são necessários. São incluídos apenas se o tempo e recursos existentes o permitirem.
- **WOULD (W)** - Requisitos considerados pelos *stakeholders* como os requisitos menos críticos (prioridade mais baixa). No caso de estes requisitos não serem incluídos, o sucesso do projeto não é impactado.

Tratando-se de “mundos” distintos (Psicologia *versus* Informática), com linguagens diferentes, algo extremamente importante foi a existência de uma adaptação e explicação de conceitos, por ambas as partes.

Para além disso, o cliente apresentou inúmeras ideias, tiradas de outras aplicações e plataformas. Foi necessário ainda, que as duas entidades estabelecessem uma boa articulação, de modo a assegurar que as vontades do cliente fossem devidamente compreendidas.

De modo a combater esta questão, a documentação de requisitos apresentada no presente capítulo, foi estruturada sob a forma de *User Stories*, por permitir uma exposição dos requisitos mais informal e natural, que facilmente pudesse ser compreendida pelo cliente, sendo expressa a partir da perspectiva do utilizador final.

Esta lista de requisitos encontra-se separada em:

- **Requisitos de Negócio (BIZ)** - Relacionados com as políticas associadas ao cumprimento dos objetivos do negócio, à satisfação dos clientes e a um bom uso dos recursos, obedecendo às leis ou convenções gerais do negócio.
- **Requisitos de Usabilidade (U)** - Relacionados com a forma como o utilizador interage com o sistema e o utiliza.
- **Requisitos Multi-Plataforma (MP)** - Relacionados com a forma como o sistema se comporta em múltiplas plataformas.
- **Requisitos Funcionais (F)** - Relacionados com tudo o que o sistema faz e oferece, em termos de tarefas e serviços.

### Convenção Utilizada

Cada requisito encontra-se definido da seguinte forma:

#ID DO REQUISITO - #NOME DO REQUISITO

O ID do requisito define-se como:

REQ\_#CATEGORIA DO REQUISITO\_#NÚMERO ATRIBUÍDO

Cada requisito contém então a sua prioridade, uma pequena descrição, *User Story* e o(s) devido(s) Critério(s) de Aceitação (lista de critérios que precisam de ser alcançados para que um *User Story* atenda ao requisito e seja aceite pelo cliente).

## 2.3 Requisitos de Negócio (BIZ)

- **REQ\_BIZ\_01 - Gratuidade**

- **Prioridade:** M
- **Descrição:** A utilização da plataforma deverá ser gratuita.
- **User Story:** Enquanto utilizador, gostaria de poder aceder à plataforma sem ter de pagar pela sua utilização.
- **Critérios de Aceitação:**
  1. O utilizador consegue interagir com a plataforma sem ter de pagar pela mesma.

## 2.4 Requisitos de Usabilidade (U)

- **REQ\_U\_01 - Gamificação**

- **Prioridade:** M
- **Descrição:** A plataforma deverá suportar recursos aplicados na gamificação.
- **User Story:** Enquanto aluno, gostaria de ter elementos de gamificação na minha plataforma de maneira a aumentar a minha motivação ao utilizá-la.
- **Critérios de Aceitação:**
  1. O aluno efetua o *login* na plataforma.
  2. Na página inicial do perfil do aluno, deve existir um local onde ele possa ver a sua pontuação.
  3. Na página inicial do perfil do aluno deve existir um local onde ele possa ver recompensas que lhe foram atribuídas.

- **REQ\_U\_02 - Garantir Usabilidade**

- **Prioridade:** M
- **Descrição:** A plataforma deverá ser eficaz, eficiente e garantir a satisfação do seu utilizador.
- **User Story:** Enquanto utilizador, de modo a ficar satisfeito, gostaria que a plataforma fosse eficaz e eficiente.



– **Critérios de Aceitação:**

1. Enquanto utilizador gostaria que a plataforma fosse rápida a responder aos meus pedidos e tivesse um design intuitivo.

## 2.5 Requisitos Multi-Plataforma (MP)

• **REQ\_MP\_01 - Responsividade - M**

– **Prioridade:** M

– **Descrição:** A plataforma deverá ser bem renderizada, independentemente do dispositivo e tamanho do ecrã do dispositivo que o utilizador esteja a utilizar.

– **User Story:** Enquanto utilizador da plataforma, gostaria que esta fosse bem renderizada para possibilitar a sua correta utilização em qualquer dispositivo.

– **Critérios de Aceitação:**

1. Um utilizador acede à plataforma num qualquer dispositivo à sua escolha.
2. A plataforma é bem renderizada, adaptando-se ao tamanho da ecrã.

• **REQ\_MP\_02 - Aplicação Móvel**

– **Prioridade:** W

– **Descrição:** A plataforma deverá ter aplicação móvel.

– **User Story:** Enquanto utilizador, gostaria de ter uma aplicação móvel para a plataforma, de maneira a me ser facilitado o acesso aos conteúdos em qualquer lugar.

– **Critérios de Aceitação:**

1. Deve haver uma versão móvel da plataforma.
2. O utilizador efetua o *download* da aplicação a partir dos serviços de distribuição digital de aplicações disponíveis (*App Store* e *Google Play*).

## 2.6 Requisitos Funcionais (F)

• **REQ\_F\_01 - Login**

– **Prioridade:** M

– **Descrição:** Os utilizadores da plataforma poderão efetuar *login* na mesma.

– **User Story:** Enquanto utilizador, gostaria de me poder autenticar na plataforma, de modo a poder aceder ao meu perfil pessoal, assim como aos conteúdos que me são disponibilizados.

– **Critérios de Aceitação:**

1. O administrador cria as contas de todos os utilizadores que se deseja que efetuem *login* na plataforma.
2. O utilizador não autenticado acede, na *landing page*, ao local para efetuar o seu *login*.
3. Nesse local aparecerá um formulário para o utilizador colocar o seu *username* e *password* correspondente.

4. Se o *username* e *password* inseridos coincidirem com uma das contas criadas, o *login* é efetuado com sucesso.
5. O utilizador (agora autenticado como professo/psicólogo, aluno, encarregado de educação, ou administrador) tem acesso à página principal da plataforma conseguindo ter acesso às funcionalidades a ele destinadas, assim como à sua informação pessoal.

• **REQ\_F\_02 - Anonimato**

- **Prioridade:** M
- **Descrição:** A identidade de todos os utilizadores da plataforma é mantida anónima.
- **User Story:** Enquanto utilizador, gostaria de poder interagir com a plataforma de forma anónima, para que os restantes utilizadores não tenham conhecimento da minha identidade.
- **Critérios de Aceitação:**
  1. O utilizador encontra-se na página de cursos.
  2. Ao aceder a uma atividade do curso, o utilizador consegue escolher em participar nessa atividade de forma anónima.
  3. A identidade do utilizador em questão é desconhecida aos restantes utilizadores.

• **REQ\_F\_03 - Inserção de Conteúdos**

- **Prioridade:** M
- **Descrição:** Os professores/psicólogos têm permissão para inserir conteúdos na plataforma (para além do administrador).
- **User Story:** Enquanto professor/psicólogo, gostaria de poder inserir os conteúdos na plataforma para que estes possam ser acedidos pelos alunos e encarregados de educação e utilizados pelos professores/psicólogos como ferramenta de apoio às sessões presenciais realizadas nas escolas.
- **Critérios de Aceitação:**
  1. No perfil de um professor/psicólogo é possibilitado o *upload* de conteúdos nos formatos .mov, .mp4, .mp3, .aac, .flac, .m4a, .oga, .ogg, .wav .txt, .doc, .docx, .pdf, ou diretamente de sites como o *Vimeo*, ou o *Youtube*.
  2. O professor/psicólogo pode, também, criar conteúdos através de recursos existentes na própria plataforma.
  3. O conteúdo é carregado corretamente (sem uma mensagem de erro).
  4. O conteúdo fica disponível na plataforma.

• **REQ\_F\_04 - Inserção de Conteúdos a Qualquer Momento**

- **Prioridade:** M
- **Descrição:** Os professores/psicólogos poderão adicionar conteúdos na plataforma em qualquer momento, mesmo em fases posteriores do projeto.

- **User Story:** Enquanto professor/psicólogo, gostaria de ter a possibilidade de, em qualquer momento que deseje, poder inserir novos conteúdos na plataforma, independentemente de os cursos estipulados inicialmente já terem sido criados.

- **Critérios de Aceitação:**

1. No perfil de um professor/psicólogo, mesmo que já tenham sido inseridos todos os conteúdos do curso inicial estipulado, é possível fazer *upload* de novos conteúdos, desde que nos formatos .mov, .mp4, .mp3, .aac, .flac, .m4a, .oga, .ogg, .wav .txt, .doc, .docx, .pdf, ou diretamente de sites como o *Vimeo*, ou o *Youtube*.
2. O professor/psicólogo pode, também, criar conteúdos através de recursos existentes na própria plataforma.
3. O novo conteúdo é carregado corretamente (sem uma mensagem de erro).
4. O novo conteúdo fica disponível na plataforma.

- **REQ\_F\_05 - Inserção de Conteúdos Direcionados**

- **Prioridade:** M

- **Descrição:** Os professores e psicólogos, ao inserir os conteúdos na plataforma poderão selecionar se estes se destinam aos encarregados de educação ou aos alunos.

- **User Story:** Enquanto professor/psicólogo, gostaria de poder selecionar o destinatário do conteúdo que coloco, de maneira a direcionar o tipo de intervenção feita (tendo em conta se o utilizador se trata de um aluno, ou de um encarregado de educação).

- **Critérios de Aceitação:**

1. No perfil de um professor/psicólogo é possibilitado o *upload* de conteúdos nos formatos .mov, .mp4, .mp3, .aac, .flac, .m4a, .oga, .ogg, .wav .txt, .doc, .docx, .pdf, ou diretamente de sites como o *Vimeo*, ou o *Youtube*.
2. O professor/psicólogo pode, também, criar conteúdos através de recursos existentes na própria plataforma.
3. O conteúdo é carregado corretamente.
4. O conteúdo fica disponível na plataforma.
5. O professor/psicólogo consegue selecionar os utilizadores aos quais pretende disponibilizar o conteúdo carregado.
6. Esses conteúdos podem ser acedidos nos perfis dos utilizadores associados.
7. Os alunos têm acesso a conteúdo exclusivo.
8. Os alunos não têm acesso aos conteúdos destinados aos encarregados de educação.
9. Os encarregados de educação têm conteúdo exclusivo.
10. Os encarregados de educação não têm acesso aos conteúdos destinados aos alunos.

- **REQ\_F\_06 - Histórico de Conteúdos**

- **Prioridade:** M

- **Descrição:** Tanto os alunos, como os encarregados de educação têm acesso a todos os conteúdos e tarefas que já tenham realizado/concluído.

- **User Story:** Enquanto aluno/encarregado de educação, quero poder aceder a conteúdos aos quais já acedi na plataforma, de modo a poder vê-los novamente.

- **Critérios de Aceitação:**

1. O aluno/encarregado de educação encontra-se no seu perfil.
2. O aluno/encarregado de educação acede às disciplinas nas quais está associado.
3. O aluno/encarregado de educação consegue visualizar todos os conteúdos pertencentes a essa disciplina.
4. Ao selecionar um desses conteúdos, mesmo que já tenha sido aberto anteriormente, o aluno/encarregado de educação consegue abri-lo novamente.

- **REQ\_F\_07 - Completude Sucessiva das Tarefas**

- **Prioridade:** M

- **Descrição:** Tanto os alunos, como os encarregados de educação só têm acesso aos materiais disponibilizados de forma sucessiva, isto é, para se ter permissão para aceder/abrir um determinado conteúdo, o anterior terá de ter sido realizado/concluído.

- **User Story:** Enquanto professor, tenciono que os alunos e encarregados de educação só tenham acesso aos conteúdos de forma sucessiva, para poderem consolidar melhor os seus conhecimentos transmitidos.

- **Critérios de Aceitação:**

1. O aluno/encarregado de educação encontra-se no seu perfil.
2. O aluno/encarregado de educação acede às disciplinas nas quais está associado.
3. Os conteúdos existentes na respetiva disciplina, à exceção do primeiro, estão bloqueados.
4. Só após se concluir o primeiro conteúdo/tarefa é que o seguinte é desbloqueado, possibilitando o acesso ao mesmo por parte do aluno/encarregado de educação (e assim sucessivamente).
5. Um aluno/encarregado de educação não consegue aceder a um conteúdo se este estiver bloqueado.
6. Um aluno/encarregado de educação não consegue aceder a um conteúdo se o anterior não foi concluído.
7. Um aluno/encarregado de educação só consegue aceder a um conteúdo que só fica desbloqueado se o anterior estiver concluído.

- **REQ\_F\_08 - Conceder Acesso a Conteúdo Sem o Anterior Estar Concluído**

- **Prioridade:** M

- **Descrição:** Os professores/psicólogos podem conceder acesso aos alunos ou encarregados de educação a sessões seguintes, mesmo que as anteriores não tenham sido realizadas/concluídas.

- **User Story:** Enquanto aluno/encarregado de educação, gostaria ter acesso a um conteúdo, apesar de não ter realizado o anterior, para poder acompanhar o que está a ser realizado na atual sessão.

– **Critérios de Aceitação:**

1. No perfil do professor/psicólogo, este pode aceder aos conteúdos que disponibilizou.
2. Selecionando um determinado conteúdo, o professor/psicólogo tem acesso aos utilizadores inscritos na respetiva disciplina onde o conteúdo se encontra.
3. O professor/psicólogo pode adicionar alunos/encarregados de educação a essa lista.
4. O aluno/encarregado de educação recebe a permissão.
5. O aluno/encarregado de educação adicionado passa a conseguir aceder ao conteúdo, mesmo se o anterior não estiver concluído.

• **REQ\_F\_09 - Notificações**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar o envio de notificações aos alunos e encarregados de educação sempre que novo conteúdo seja colocado na plataforma.

– **User Story:** Enquanto aluno/encarregado de educação, gostaria de poder receber notificações quando um novo conteúdo é inserido na plataforma, para me poder manter sempre atualizado.

– **Critérios de Aceitação:**

1. Todos os utilizadores têm o seu e-mail associado à sua conta.
2. O professor/psicólogo encontra-se no seu perfil.
3. O professor/psicólogo acede ao local para *upload* de conteúdos.
4. No momento em que um professor/psicólogo faz *upload* de um conteúdo na plataforma, o aluno/encarregado de educação recebe uma notificação com essa informação.
5. O aluno/encarregado de educação recebe, por e-mail e/ou via web (na própria plataforma) a notificação, ficando a par da informação nela contida.

• **REQ\_F\_10 - Existência de um gráfico de Humor e Atividades**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar o acesso a um gráfico onde os alunos poderão comparar a evolução do seu humor em relação às atividades que realizam.

– **User Story:** Enquanto aluno, gostaria de aceder a um gráfico que me permita comparar a evolução do meu humor em relação ao número de atividades que realizei.

– **Critérios de Aceitação:**

1. O aluno, no seu perfil, tem um local onde preenche diariamente o seu humor e as atividades diárias.
2. O aluno, no seu perfil, pode aceder ao gráfico que relaciona a informação recolhida nos dois diários (relação entre o número de tarefas realizadas e o seu humor).
3. No eixo dos *xx* do gráfico estão representados os dias da semana corrente.

4. No eixo dos *yy* estão representados os valores possíveis de ser registados no “Diário de Humor” (zero a cem) e no “Diário de Atividades” (de zero a cinquenta).
5. No gráfico encontram-se visualizadas sob a forma de duas linhas os valores inseridos pelo aluno, em cada dia, nos diários referidos.

• **REQ\_F\_10.1 - Existência de um “Diário de Humor”**

- **Prioridade:** M
- **Descrição:** A plataforma deverá possibilitar a existência de um “Diário de Humor”, onde o aluno selecionará um valor de zero a cem, correspondente ao valor do seu humor (de zero a cem, sendo zero, mau, e cem, eufórico) no respetivo dia.
- **User Story:** Enquanto aluno, gostaria de ter um local que me permitisse selecionar o valor do meu humor diário.
- **Critérios de Aceitação:**
  1. O aluno encontra-se no seu perfil.
  2. O aluno acede ao diário onde preenche diariamente o valor do seu humor sob a forma de um *range slider*.
  3. O valor do humor varia entre zero e cem.
  4. O aluno seleciona um valor.
  5. O aluno submete o valor que selecionou.
  6. O valor selecionado aparecerá no “Gráfico de Humor e Atividades” no dia correspondente.
  7. O aluno só tem possibilidade de realizar uma submissão por dia.

• **REQ\_F\_10.2 - Existência de um “Diário de Atividades”**

- **Prioridade:** M
- **Descrição:** A plataforma deverá possibilitar a existência de um “Diário de Atividades”, onde o aluno pode fazer um registo diário das atividades que realizou no respetivo dia.
- **User Story:** Enquanto aluno, gostaria de ter um local que me permitisse preencher as atividades que realizei durante o dia.
- **Critérios de Aceitação:**
  1. O aluno, no seu perfil, acede ao seu diário de atividades.
  2. O local de registo assume a forma de uma tabela.
  3. Cada linha da tabela deverá representar uma atividade e as colunas os dias da semana.
  4. O número de atividades a preencher deverão ser cinco, mas o aluno tem uma opção para adicionar linhas (máximo de cinquenta linhas e, portanto, cinquenta atividades).
  5. O aluno deverá poder apagar linhas (apagar atividades).

6. O registo efetuado em cada dia aparecerá no “Gráfico do Humor e Atividades”.

- **REQ\_F\_11 - Fórum**

- **Prioridade:** S

- **Descrição:** A plataforma deverá suportar a existência de um fórum onde alunos e encarregados de educação poderão submeter testemunhos e opiniões pessoais.

- **User Story:** Enquanto aluno/encarregado de educação, gostaria de poder aceder a um fórum de maneira a poder deixar os meus testemunhos e histórias e ajudar alguém com o relato das minhas experiências.

- **Critérios de Aceitação:**

1. No seu perfil, um aluno/encarregado de educação acede ao fórum.
2. O aluno/encarregado de educação pode escrever no fórum o que deseja submeter.
3. O aluno/encarregado de educação submete o que escreveu.
4. O professor/psicólogo tem acesso a esta submissão.

- **REQ\_F\_12 - Seleção das Submissões a Tornar Públicas Na Plataforma**

- **Prioridade:** S

- **Descrição:** Os professores/psicólogos terão acesso às submissões realizadas pelos alunos e encarregados de educação no fórum e poderão selecionar aquelas que tencionam tornar públicas na plataforma.

- **User Story:** Enquanto professor/psicólogo, quero selecionar as submissões que desejo tornar públicas.

- **Critérios de Aceitação:**

1. O professor/psicólogo tem acesso, no seu perfil, às submissões realizadas no fórum.
2. O professor/psicólogo seleciona as submissões que deseja ver públicas.
3. As submissões que selecionou tornarem-se públicas tornam-se, efetivamente e para todos os utilizadores.
4. Não são tornadas públicas submissões que o professor/psicólogo não selecionou.

- **REQ\_F\_13 - Envio de Mensagens**

- **Prioridade:** C

- **Descrição:** Deve ser possibilitado o envio de mensagens de texto aos alunos e encarregados de educação.

- **User Story:** Enquanto professor/psicólogo, gostaria que os alunos e encarregados de educação recebessem mensagens de texto (*short message service* - SMSs).

- **Critérios de Aceitação:**

1. O aluno/encarregado de educação recebe mensagens de texto (SMSs).
2. O aluno/encarregado de educação consegue abrir as mensagens recebidas.

- **REQ\_F\_14 - Feedback**

- **Prioridade:** S
- **Descrição:** Os alunos/encarregados de educação receberão *feedback* sempre que concluírem uma tarefa.
- **User Story:** Enquanto aluno/encarregado de educação, gostaria de receber *feedback* sempre que concluo uma tarefa.
- **Critérios de Aceitação:**
  1. O aluno/encarregado de educação encontra-se no seu perfil.
  2. O aluno/encarregado de educação acede às disciplinas nas quais está associado.
  3. O aluno/encarregado de educação seleciona um conteúdo com uma tarefa para realizar.
  4. O aluno/encarregado de educação conclui uma tarefa.
  5. O aluno/encarregado de educação recebe *feedback* (automático), quando a tarefa é concluída.

- **REQ\_F\_15 - Redes Sociais**

- **Prioridade:** W
- **Descrição:** Os professores/psicólogos poderão ter uma ligação direta ao *Facebook* e ao *Twitter* através de um botão de partilha para as respetivas redes sociais.
- **User Story:** Enquanto professor/psicólogo, gostaria de poder partilhar a plataforma através de botões de partilha nas respetivas redes sociais.
- **Critérios de Aceitação:**
  1. O professor/psicólogo encontra-se no seu perfil.
  2. O perfil do professor/psicólogo fornece dois botões: um de partilha no *Facebook* e outro de partilha no *Twitter*.
  3. O professor/psicólogo clica no botão de partilha para o *Facebook* ou para o *Twitter*.
  4. No caso de o professor/psicólogo não ter efetuado o *login* na sua rede social, é aberta uma nova página no *browser* para efetuar o *login* na respetiva rede. No caso de o professor/psicólogo já ter efetuado o respetivo *login*, a nova janela aberta já permite a partilha diretamente.
  5. O professor/psicólogo consegue fazer uma partilha da plataforma na respetiva rede social (ao clicar no botão do *Facebook* faz uma partilha no seu perfil do *Facebook*; ao clicar no botão do *Twitter*, faz uma partilha no seu perfil do *Twitter*).

- **REQ\_F\_16 - Chats**

- **Prioridade:** W
- **Descrição:** Os professores/psicólogos poderão abrir *chats* aos quais os alunos e encarregados de educação poderão aceder.



- **User Story:** Enquanto professor/psicólogo, gostaria de criar *chats* com uma entidade competente, para os alunos poderem fazer perguntas às quais o psicólogo responderia.
- **Critérios de Aceitação:**
  1. Os professores/psicólogos encontram-se no seu perfil.
  2. Os professores/psicólogos, no seu perfil, têm uma opção para criar *chats*.
  3. O professor/psicólogo deve seleccionar os alunos/encarregados de educação que podem ter acesso ao *chat*.
  4. Os alunos/encarregados de educação seleccionados têm permissão para aceder ao *chat*.
  5. Alunos/encarregados de educação que não foram seleccionados não têm acesso ao *chat*.
  6. Os alunos/encarregados de educação conseguem escrever e enviar mensagens no *chat*.
  7. Os restantes utilizadores do *chat* recebem, em tempo-real, as mensagens que são escritas no *chat*.
  8. Os professores/psicólogos têm maneira de encerrar o *chat* quando desejarem.
  9. A sequência de mensagens existente no *chat* não é guardada.

- **REQ\_F\_17 - Progresso na Tarefa**

- **Prioridade:** C
- **Descrição:** Os alunos/encarregados de educação terão possibilidade de avaliar a sua evolução, dentro de uma tarefa específica.
- **User Story:** Enquanto aluno/encarregado de educação, gostaria de ver em que “ponto” vou numa determinada tarefa, para ter a percepção do meu progresso dentro da mesma e o quão perto estou de a concluir.
- **Critérios de Aceitação:**
  1. O aluno/encarregado de educação encontra-se no seu perfil.
  2. O aluno/encarregado de educação acede às disciplinas nas quais está associado.
  3. O aluno/encarregado de educação selecciona um conteúdo com uma tarefa para realizar.
  4. Dentro do conteúdo o aluno/encarregado de educação consegue ter acesso ao seu progresso na tarefa (por exemplo: percentagem de perguntas realizadas numa tarefa; ponto 2 de 5; página 3 de 4).

### 2.6.1 Integração de Gamificação

O cliente da plataforma SMS-eSaúde (os investigadores da FPCE) tenciona ver integrada uma perspectiva de gamificação. Contudo, o cliente não era detentor de uma ideia concreta da forma como essa característica poderia ser aplicada na plataforma e, prova disso, foi o requisito de gamificação enunciado na secção anterior (REQ\_U\_01), que dado o tema, se revela um pouco vago.

Inicialmente, o cliente tinha referido que desejava ver integrados “jogos” na plataforma. Contudo, após uma análise dos “jogos” desejados, esses não passavam de simples *quizzes*.

De maneira a definir melhor o que se pretendia com “integração de gamificação na plataforma” investigou-se o conceito de “gamificação” e de que forma poderia ser aplicado.

### **Definição de Gamificação**

Considera-se que o mercado de softwares inspirados em videojogos teve, nos últimos anos, uma multiplicação em massa. Esta tendência, designada comumente como “gamificação” tem ganho atenção significativa entre praticantes e estudiosos de jogos [3], correlacionando um conjunto considerável de conceitos.

São, deste modo, identificadas diversas formas de os jogos serem aplicados:

- Na interação humano-computador (como por exemplo, em jogos de realidade virtual ou seja, bastante imersivos);
- Em jogos que possibilitam transmitir conteúdo de treino/educacional, considerada a área mais ampla dos jogos existentes (“*serious games*”);
- Em jogos em que a sua experiência é estendida para o mundo real, isto é, onde o mundo fictício em que o jogo ocorre, se mistura com o mundo físico (“*pervasive games*”).

Contudo, não está claro como o conceito de gamificação se relaciona com estes jogos, se se trata somente de um fenómeno novo e como pode ser definido.

O primeiro uso documentado de gamificação [4] data o ano de 2008, tendo sido aplicado na indústria da média digital. Contudo, a maior adoção do termo revelou-se dois anos depois.

Na atualidade, o termo “gamificação” já é extramamente comum, embora contestado, devido à sua aplicação “desmesurada” derivada de simplificações excessivas e diferentes interpretações (levando a que o termo esteja constantemente a ser cunhado de forma distinta). Um exemplo passa pelos vendedores e consultores de um determinado produto, que tendem a descrever a gamificação nos termos e benefícios que querem vender aos seus clientes.

Na indústria, as utilizações do termo variam entre dois conceitos, essencialmente. O primeiro é referente à crescente adoção, presença e o impacto de videojogos no dia-a-dia [5], na medida em que estes representam culturalmente uma parte central das nossas vidas. Este fator influencia fortemente o seu consumo, leva à criação de comunidades e até formação de identidades e molde de carácter [5]. O segundo, uma noção mais específica, é que como os videojogos são projetados para o entretenimento (sendo esse o seu foco principal), visam a motivar os seus utilizadores para um maior envolvimento e continuidade no uso das aplicações. Assim, os elementos de jogo selecionados devem tornar os produtos e serviços não relacionados a jogos mais agradáveis e envolventes [6].

“Gamificação” pode, assim, ser definida [7] como o uso de recursos utilizados no design e conceção de jogos, em contextos não relacionados a jogos, consisendo-se que estes elementos motivam e aumentam a atividade e fidelização do utilizador.

A melhoria na experiência do utilizador, o seu envolvimento e, principalmente, motivação para utilizar um dado produto, representam os casos de uso que predominam perante essa definição, sendo estes os principais objetivos a alcançar ao inserir a gamificação na plataforma SMS-eSaúde.

Sendo assim, torna-se necessário identificar que recursos motivacionais, aplicados na gamificação, farão mais sentido empregar na plataforma SMS-eSaúde.

## **Recursos Motivacionais Aplicados na Gamificação**

De maneira a que o fator “motivação” seja potenciado e eficazmente aplicado, há vários recursos que são utilizados [8]:

### **- Pontuação [9]**

No contexto da plataforma SMS seria benéfica a atribuição de pontos aos alunos em atividades tais como: presença na aula/sessão, contribuição com respostas a perguntas, tutoriais, tarefas e testes, trabalhos e sua conclusão.

### **- Classificação [9]**

Após visualizar o seu resultado individual, através de tabelas de classificações, o aluno poderia comparar o seu desempenho com os restantes, potenciando a comunicação entre eles. Estas tabelas de classificação também poderiam ser utilizadas como um registo e suporte para *feedback*.

### **- Conquistas/Distintivos/Recompensas [10] [11]**

Outra abordagem comum para motivar diretamente os utilizadores é a atribuição de distintivos e/ou recompensas após o utilizador realizar determinadas conquistas. Num âmbito geral, estas recompensas podem incluir um aumento na avaliação do progresso do utilizador, ou o desbloqueio de níveis.

### **- Níveis [10]**

Quanto maior for a pontuação do utilizador, maior o nível onde este se encontrará.

### **- História/Tema [11]**

É importante contextualizar o utilizador na história/tema do sistema, para que se sinta inserido e compreenda o seu papel na plataforma.

### **- Objetivos claros [11]**

Semelhantemente com o tema, a presença de objetivos claros é importante para que os utilizadores entendam a tarefa a que estão destinados concluir.

### **- *Feedback* [11]**

Os videojogos típicos, baseando-se nas informações reunidas de progresso e pontuação, por exemplo, fornecem *feedback* imediato aos utilizadores, sendo essa interação em “tempo real” benéfica para que o utilizador sinta um contacto mais próximo e direto com a plataforma.

### **- Progresso [11]**

Um jogo ajuda os utilizadores a aumentar de forma contínua as suas competências e habilidades pela divulgação constante e progressiva do conhecimento. Para além disto, com este recurso tenciona-se que o utilizador tenha constantemente a noção da sua evolução em cada tarefa da plataforma.

### **- Desafio [11]**

A existência de um cronómetro (pressão do tempo), por exemplo, pode também ser um aspeto importante em jogos: adicionar esta pressão estabelece metas claras e desafiadoras.

Considera-se que a existência de pontos, tabelas de classificação e distintivos se revelam os componentes mais utilizados [8].

### 2.6.2 Identificação dos Recursos de Gamificação a Integrar na Plataforma

Dados os recursos apresentados na secção anterior e de acordo com os requisitos descritos no subcapítulo 2.6, é possível perceber que recursos relativos ao “*Feedback*” e “*Progresso*” já tinham sido identificados aquando da documentação dos requisitos (requisitos “REQ\_F\_14” e “REQ\_F\_17”, respetivamente). O mesmo acontece com os recursos “*Objetivos Claros*” e “*História/Tema*”, uma vez que o âmbito do Projeto SMS será dado a conhecer aos seus utilizadores, assim como com as tarefas e exercícios a eles destinados, que serão sempre devidamente explicados.

Deste modo, os restantes recursos de gamificação a ser implementados foram:

1. Pontuação
2. Atribuição de Recompensas

Em conjunto com o cliente, definiram-se os critérios para o uso destes recursos de gamificação:

#### **CrITÉRIOS de Atribuição de Pontuação:**

- O aluno ganha **2 pontos** quando:
  - Preenche uma alínea de um “jogo” (um *quiz*).
- O aluno ganha **5 pontos** quando:
  - Efetua *login* na plataforma;
  - Vê um vídeo sem pergunta;
  - Realiza uma tarefa diária:
    1. Registo no diário do humor;
    2. Registo no diário de atividades.
- O aluno ganha **10 pontos** quando:
  - Realiza um inquérito;
  - Vê um vídeo com pergunta embutida;
  - Responde a um *quiz*;
  - Submete um registo semanal.
- O aluno ganha **20 pontos** quando:
  - Ouve um ficheiro de áudio;
  - Participa na sessão presencial.

#### **CrITÉRIOS de Atribuição de Recompensa:**

- Atribuir ao aluno um “**crachá pequeno**” quando:
  - O aluno atinge 100 pontos.

- Atribuir ao aluno um “**crachá grande**” quando:
  - O aluno atinge 200 pontos.

Estas condições acrescentam novos Requisitos Funcionais (F) à lista anteriormente apresentada, correspondendo assim a funcionalidades extra que foram necessárias implementar.

### 2.6.3 Requisitos Funcionais (F) Adicionais (de Gamificação)

- **REQ\_F\_18 - Pontuação**

- **Prioridade:** M
- **Descrição:** A plataforma deverá suportar a atribuição de pontuação aos alunos após a realização de uma tarefa.
- **User Story:** Enquanto aluno, gostaria de receber pontos sempre que concluo uma dada tarefa.
- **Critérios de Aceitação:**
  1. O aluno encontra-se no seu perfil.
  2. O aluno consegue ver no seu perfil a sua pontuação.

- **REQ\_F\_18.1 - Atribuição de dois pontos**

- **Prioridade:** M
- **Descrição:** A plataforma deverá suportar a atribuição de dois pontos aos alunos após o preenchimento de uma pergunta de um “jogo”.
- **User Story:** Enquanto aluno, gostaria de receber 2 pontos por resposta a uma pergunta de um “jogo”.
- **Critérios de Aceitação:**
  1. O aluno encontra-se no seu perfil.
  2. O aluno acede aos conteúdos a ele destinados.
  3. O aluno seleciona um “jogo”.
  4. A pontuação do aluno deve aumentar 14 pontos (dois pontos por sete perguntas) quando o aluno conclui o “jogo”.

- **REQ\_F\_18.2 - Atribuição de cinco pontos**

- **Prioridade:** M
- **Descrição:** A plataforma deverá suportar a atribuição de 5 pontos aos alunos após a visualização de um vídeo (sem pergunta) ou a realização de uma das tarefas diárias.
- **User Story:** Enquanto aluno, gostaria de receber 5 pontos sempre que entro na plataforma, vejo um vídeo (sem pergunta) ou realizo uma das tarefas diárias.
- **Critérios de Aceitação:**

**1º Caso - Aluno efetua *login* na plataforma**

  1. O aluno efetua o *login* na plataforma.

2. A pontuação do aluno aumenta 5 pontos.

**2º Caso** - Aluno vê vídeo sem pergunta

1. O aluno encontra-se no seu perfil.
2. O aluno acede aos conteúdos a ele destinados.
3. O aluno seleciona um vídeo sem pergunta.
4. A pontuação do aluno aumenta 5 pontos.

**3º Caso** - Aluno realiza uma tarefa diária

1. O aluno encontra-se no seu perfil.
2. O aluno seleciona a tarefa diária onde pretende fazer o registo (Diário do Humor; Diário de Atividades).
3. O aluno efetua o registo e submete.
4. A pontuação do aluno aumenta 5 pontos.

**REQ\_F\_18.3 - Atribuição de dez pontos**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar a atribuição de 10 pontos aos alunos após o preenchimento de um inquérito; visualização de um vídeo com pergunta embutida; resposta a um *quiz*; realização de um registos semanal.

– **User Story:** Enquanto aluno, gostaria de receber 10 pontos sempre que preencho um inquérito, vejo um vídeo (com pergunta), respondo a um *quiz* e faço os registos semanais.

– **Critérios de Aceitação:**

**1º Caso** - Aluno preenche inquérito

1. O aluno encontra-se no seu perfil.
2. O aluno acede aos conteúdos a ele destinados.
3. O aluno seleciona um inquérito.
4. O aluno preenche e submete o inquérito.
5. A pontuação do aluno aumenta 10 pontos.

**2º Caso** - Aluno vê vídeo com pergunta

1. O aluno encontra-se no seu perfil.
2. O aluno acede aos conteúdos a ele destinados.
3. O aluno seleciona um vídeo com pergunta.
4. A pontuação do aluno aumenta 10 pontos.

**3º Caso** - Aluno realiza um *quiz*

1. O aluno encontra-se no seu perfil.
2. O aluno acede aos conteúdos a ele destinados.
3. O aluno seleciona um *quiz*.

4. O aluno preenche e submete o *quiz*.
5. A pontuação do aluno aumenta 10 pontos.

**4º Caso** - Aluno preenche um registo semanal

1. O aluno encontra-se no seu perfil.
2. O aluno acede ao local dos registos semanais.
3. O aluno seleciona o registo semanal que tenciona preencher.
4. O aluno preenche e submete o registo semanal.
5. A pontuação do aluno aumenta 10 pontos.

• **REQ\_F\_18.4 - Atribuição de vinte pontos**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar a atribuição de 20 pontos aos alunos quando estes ouvem um ficheiro de áudio e participam na sessão presencial.

– **User Stories:**

**1º Caso** - Enquanto aluno, desejo receber 20 pontos sempre que ouço um ficheiro de áudio.

**Critérios de Aceitação:**

1. O aluno encontra-se no seu perfil.
2. O aluno acede aos conteúdos a ele lhe destinados.
3. O aluno seleciona um ficheiro de áudio.
4. A pontuação do aluno aumenta 20 pontos.

**2º Caso** - Enquanto professor/psicólogo, gostaria de atribuir 20 pontos aos alunos que participaram na sessão presencial.

**Critérios de Aceitação:**

1. No perfil do professor/psicólogo, este tem um local para assinalar presenças.
2. Todos os alunos assinalados (como “Presente” ou “Atrasado”) receberão 20 pontos.
3. A pontuação dos alunos que não foram assinalados não aumenta.

• **REQ\_F\_19 - Atribuição de Recompensas**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar a atribuição de uma recompensa aos alunos quando atingem uma determinada pontuação.

– **User Story:** Enquanto aluno, gostaria de receber uma recompensa após realizar uma dada conquista.

– **Critérios de Aceitação:**

1. O aluno encontra-se no seu perfil.
2. O aluno vê no seu perfil a sua recompensa mais recente.

3. O aluno consegue aceder à tabela de recompensas.

• **REQ\_F\_19.1 - Atribuição de “crachá pequeno”**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar a atribuição de um “crachá pequeno” aos alunos quando atingem 100 pontos.

– **User Story:** Enquanto aluno, gostaria de receber um “crachá pequeno” quando atinjo 100 pontos.

– **Critérios de Aceitação:**

1. A pontuação do aluno atinge mais 100 pontos.
2. O aluno recebe um “crachá pequeno”.
3. O novo crachá fica visível no seu perfil.

• **REQ\_F\_19.2 - Atribuição de “crachá grande”**

– **Prioridade:** M

– **Descrição:** A plataforma deverá suportar a atribuição de um “crachá grande” aos alunos quando atingem 200 pontos.

– **User Story:** Enquanto aluno, gostaria de receber um “crachá grande” quando atinjo 200 pontos.

– **Critérios de Aceitação:**

1. A pontuação do aluno atinge mais 200 pontos.
2. O aluno recebe um “crachá grande”.
3. O novo crachá fica visível no seu perfil.

## 2.7 Lista Final de Requisitos

A lista final dos requisitos desejados pelo cliente, encontra-se exposta nas tabelas 1 e 2, estando os requisitos ordenados de acordo com a sua categoria.

É possível observar a existência de:

- Trinta e dois requisitos.
- Vinte e quatro requisitos de prioridade *MUST* (M) - 75% dos requisitos.
- Três requisitos com prioridade *SHOULD* (S).
- Dois requisitos de prioridade *COULD* (C).
- Três requisitos de prioridade *WOULD* (W).

Estes requisitos impostos pelo cliente constituíram a base para a seleção da plataforma (OSS) a adotar, que se apresentará no próximo capítulo (capítulo 3). Foi essencial estabelecer com o cliente o compromisso de que o âmbito destes requisitos não seria alterado, pelo facto de a seleção da plataforma estar fortemente dependente destes requisitos. A sua alteração, numa fase posterior, poderia tornar a escolha efetuada inútil, comprometendo assim o trabalho já desenvolvido.



Tabela 1: Lista final de requisitos

Alteração	Criação	ID	Categoria	Descrição	Prioridade atribuída (Método Moscow)
30-set.-2019	30-set.-2019	1.	Negócio (BIZ)	<b>Gratuidade</b> - a plataforma deverá ser gratuita.	M
10-nov.-2019	30-set.-2019	1.	Usabilidade (U)	<b>Gamificação</b> - a plataforma deverá suportar mecanismos de gamificação.	M
10-nov.-2019	30-set.-2019	2.	Usabilidade	<b>Garantir Usabilidade</b> - a plataforma deverá ser eficaz, eficiente e garantir a satisfação do seu utilizador.	M
30-set.-2019	30-set.-2019	1.	Multiplataforma (MP)	<b>Responsividade</b> - a plataforma deverá ser bem renderizada, independentemente do dispositivo e, assim, tamanho do ecrã que o utilizador esteja a utilizar.	M
19-fev.-2020	19-fev.-2020	2.	Multiplataforma	<b>Aplicação Móvel</b> - a plataforma deverá ter aplicação móvel.	W
30-set.-2019	30-set.-2019	1.	Funcional (F)	<b>Login</b> - os utilizadores da plataforma podem efetuar login na plataforma.	M
10-nov.-2019	30-set.-2019	2.	Funcional	<b>Anonimato</b> - a identidade dos utilizadores da é mantida anónima.	M
30-set.-2019	30-set.-2019	3.	Funcional	<b>Inserção de Conteúdos</b> - os professores e psicólogos têm permissão para inserir conteúdos na plataforma.	M
10-nov.-2019	30-set.-2019	4.	Funcional	<b>Inserção de Conteúdos em Qualquer Momento</b> - os professores e psicólogos poderão adicionar conteúdos em qualquer momento, mesmo em fases posteriores do projeto.	M
30-set.-2019	30-set.-2019	5.	Funcional	<b>Inserção de Conteúdos Direcionados</b> - os professores e psicólogos, ao inserir os conteúdos poderão selecionar se este se destina aos encarregados de educação ou aos alunos.	M
30-set.-2019	30-set.-2019	6.	Funcional	<b>Histórico de Conteúdos</b> - tanto os alunos, como os encarregados de educação têm acesso a todos os conteúdos e tarefas que já tenham realizado/concluído.	M
10-nov.-2019	10-nov.-2019	7.	Funcional	<b>Completo Sucessiva das Tarefas</b> - tanto os alunos, como os encarregados de educação só têm acesso aos materiais disponibilizados de forma sucessiva, isto é, para se ter permissão para aceder/abrir um determinado conteúdo, o anterior terá de ser sido realizado/concluído.	M
10-nov.-2019	10-nov.-2019	8.	Funcional	<b>Conceder Acesso a Conteúdo Não Completo</b> - os professores e psicólogos podem conceder acesso ao alunos ou encarregados de educação a sessões seguintes, mesmo que as anteriores não tenham sido realizadas/concluídas.	M
10-nov.-2019	10-nov.-2019	9.	Funcional	<b>Notificações</b> - a plataforma deverá suportar o envio de notificações aos alunos e encarregados de educação sempre que novo conteúdo seja colocado na plataforma.	M
10-nov.-2019	10-nov.-2019	10.	Funcional	<b>Gráficos</b> - a plataforma deverá suportar o acesso a gráficos onde os alunos poderão comparar a evolução do seu humor em relação às atividades que realizam.	M
10-nov.-2019	4-mar.-2020	10.1.	Funcional	<b>Diário de Humor</b> - a plataforma deverá possibilitar ao aluno inserir um valor, de 0 a 100, representativo do seu humor no dia respetivo.	M
10-nov.-2019	8-jun.-2020	10.2.	Funcional	<b>Diário de Atividades</b> - a plataforma deverá possibilitar ao aluno a inserção das atividades que realizou no dia respetivo.	M
30-set.-2019	30-set.-2019	11.	Funcional	<b>Fórum</b> - a plataforma deverá suportar a existência de um fórum onde alunos e encarregados de educação poderão submeter testemunhos e opiniões pessoais.	S
30-set.-2019	30-set.-2019	12.	Funcional	<b>Seleção das Submissões a Tornar Públicas Na Plataforma</b> - os professores e psicólogos terão acesso às submissões realizadas pelos alunos e encarregados de educação no fórum e poderão selecionar aquelas que tencionam ver tomadas públicas na plataforma.	S

Tabela 2: Lista final de requisitos - continuação

Alteração	Criação	ID	Categoria	Descrição	Prioridade atribuída (Método Moscow)
30-set.-2019	30-set.-2019	13.	Funcional	<b>Envio de Mensagens</b> - deve ser possibilitado o envio de SMSs aos alunos/encarregados de educação.	C
22-fev.-2020	22-fev.-2020	14.	Funcional	<b>Feedback</b> - os alunos receberão feedback ao concluir uma tarefa da plataforma.	S
30-set.-2019	30-set.-2019	15.	Funcional	<b>Redes Sociais</b> - os professores/psicólogos terão, no seu perfil, um botão para partilha da plataforma nas suas redes sociais.	W
10-nov.-2019	30-set.-2019	16.	Funcional	<b>Chats</b> - os professores e psicólogos poderão criar chats aos quais os devidos utilizadores poderão aceder.	W
30-set.-2019	30-set.-2019	17.	Funcional	<b>Progresso na Tarefa</b> - os alunos e encarregados de educação terão possibilidade de avaliar a sua evolução, dentro de uma tarefa específica.	C
24-fev.-2019	24-fev.-2019	18.	Funcional	<b>Pontuação</b> - a plataforma deverá suportar a atribuição de pontuação aos alunos após a realização de uma determinada tarefa.	M
24-fev.-2019	24-fev.-2019	18.1.	Funcional	<b>Atribuição de dois pontos</b> - a plataforma deverá suportar a atribuição de dois pontos aos alunos após o preenchimento de uma alínea de um "jogo".	M
24-fev.-2019	24-fev.-2019	18.2.	Funcional	<b>Atribuição de cinco pontos</b> - a plataforma deverá suportar a atribuição de 5 pontos aos alunos após a visualização de um vídeo (sem pergunta).	M
24-fev.-2019	24-fev.-2019	18.3.	Funcional	<b>Atribuição de dez pontos</b> - a plataforma deverá suportar a atribuição de 10 pontos aos alunos após o preenchimento de um inquérito.	M
				<b>Atribuição de dez pontos</b> - a plataforma deverá suportar a atribuição de 10 pontos aos alunos após a visualização de um vídeo com pergunta embuída.	
				<b>Atribuição de dez pontos</b> - a plataforma deverá suportar a atribuição de 10 pontos aos alunos após realização de um quiz.	
				<b>Atribuição de dez pontos</b> - a plataforma deverá suportar a atribuição de 10 pontos aos alunos após realização de um registo semanal.	
				<b>Atribuição de vinte pontos</b> - a plataforma deverá suportar a atribuição de 20 pontos aos alunos por ouvir um ficheiro de áudio.	
24-fev.-2019	24-fev.-2019	19.	Funcional	<b>Recompensa</b> - a plataforma deverá suportar a atribuição de uma recompensa aos alunos quando atingem uma determinada pontuação.	M
24-fev.-2019	24-fev.-2019	19.1.	Funcional	<b>Atribuição de "crachá pequeno"</b> - a plataforma deverá suportar a atribuição de um "crachá pequeno" aos alunos quando atingem 100 pontos.	M
24-fev.-2019	24-fev.-2019	19.2.	Funcional	<b>Atribuição de "crachá grande"</b> - a plataforma deverá suportar a atribuição de um "crachá grande" aos alunos quando atingem 200 pontos.	M

### 3 Seleção de Uma Plataforma *Open-Source*

As aplicações proprietárias encontram-se, cada vez mais, a ser substituídas por OSSs. Estes, em maior parte dos casos e com vista à redução dos custos, passaram a ser a primeira escolha. Ademais, associando o contributo dado pelo envolvimento de *developers* ao longo dos anos, o seu destaque tem vindo a ser solidificado.

Estes sistemas, apresentam vários benefícios:

- O facto de possuírem fortes fontes de suporte;
- Apresentarem uma maior confiabilidade;
- Reduzirem os custos investidos;
- Aumentarem a produtividade;
- Revelarem-se vantajosos quando o período para desenvolvimento é relativamente curto.

Percebendo-se a importância desses benefícios, assumiu-se que os OSSs representariam a melhor solução para o Projeto SMS.

Com base nos requisitos reunidos até ao momento, a fase seguinte do desenvolvimento do projeto caracterizou-se pela exploração de OSSs que reunissem o máximo das funcionalidades desejadas. Uma vez que se tencionava que a plataforma SMS-eSaúde fosse um meio para que os professores e psicólogos distribuíssem e gerissem os conteúdos aos restantes utilizadores (alunos e encarregados de educação), iniciou-se então a pesquisa por Sistemas de Gestão de Conteúdos (CMS) *open-source*.

#### 3.1 Sistemas de Gestão de Conteúdos (CMS)

É notório o quanto a tecnologia se encontra enraizada na nossa vida quotidiana, tendo transformado na totalidade a forma como pensamos e interagimos com o mundo à nossa volta. A área da educação não é exceção [12], sendo que esta junção (tecnologia e educação) se revela benéfica em cumprir o principal objetivo de sistemas educativos: um maior progresso na aquisição de conhecimentos e, portanto, uma melhor aprendizagem [13].

Muitos foram os sistemas que, ao longo dos anos, têm sido criados e evoluindo com vista a fornecer produtos educativos inovadores e com cada vez mais funcionalidades.

Posto isto, e sendo o Projeto SMS de cariz pedagógico, a criação de uma plataforma *online* faz todo o sentido, pois promove a facilidade de transmissão dos conhecimentos que se pretende que tanto alunos como encarregados de educação adquiram. Estes conhecimentos passarão, para além das sessões presenciais, por conteúdos inseridos na plataforma pelos professores e psicólogos.

Ao se explorar soluções que permitissem a inserção e gestão de conteúdos, a ser “consumidos”, percebeu-se que os Sistemas de Gestão de Conteúdos (CMS) permitem:

- Adicionar conteúdos;
- Visualizar conteúdos;
- Gerir conteúdos (edição, remoção, etc.).

Deste modo, procedeu-se à pesquisa e exploração de CMSs *open-source*, por forma a avaliar as suas funcionalidades e comparar com os requisitos recolhidos para, por fim, selecionar uma plataforma.

Durante o processo de pesquisa, concluiu-se que seria mais benéfico optar por soluções mais “restritas” e direcionadas ao que se procurava para a plataforma SMS-eSaúde. Optou-se então por Sistemas de Gestão de Aprendizagem (LMSs), um tipo específico de CMS.

### 3.1.1 Sistemas de Gestão de Aprendizagem (LMS)

Como os LMSs permitem gerir e disponibilizar conteúdos destinados à aprendizagem, de forma digital, tratando-se esse do objetivo primordial deste projeto, considerou-se que optar por estes sistemas produziria melhores resultados. Os LMSs, para além de conterem, obviamente, as ferramentas oferecidas pelos CMSs, proporcionam outras [13][14] como: hospedar cursos; organizar os conteúdos por cursos e/ou disciplinas; distribuir os cursos de acordo com os utilizadores; acompanhar o progresso dos alunos; possuir métodos de avaliação; possibilitar envio de notificações; suportar recursos de gamificação; possibilitar um maior envolvimento com e do aluno; conter uma interface *user-friendly*.

Geralmente, as características de um LMSs podem ser distinguidos como básicos ou avançados.

#### Características Inerentes aos LMSs

No que diz respeito às funções mais elementares [14][15] dos LMSs, podemos então considerar:

- Gestão de cursos, possibilitando o fornecimento (*upload*) de materiais de aprendizagem e atribuição dos seus respetivos destinatários;
- Gestão de utilizadores (adicionar utilizadores, editá-los, atribuir-lhes funções e agrupá-los, por exemplo);
- Acompanhamento do progresso do aluno, na medida em que o seu desempenho é medido e avaliado.

Relativamente a funcionalidades consideradas mais avançadas [15], podemos referir:

- Gamificação, ao introduzir elementos como pontuação, classificação e medalhas, melhorando o envolvimento e motivação dos alunos [12];
- Criação de conteúdos, na medida em que algumas plataformas possuem um editor interno integrado, que permite criar cursos e até testes simples;
- Aprendizagem móvel, visto que a maioria dos LMSs viabiliza que os alunos acedam aos conteúdos em qualquer hora, lugar e dispositivo. Alguns sistemas possuem, inclusivamente, aplicações móveis próprias, de acesso *online* e *offline*;
- Aprendizagem social, dado que os alunos são incentivados a partilhar os seus conhecimentos e experiências através da conexão com redes sociais. Esta integração possibilita a publicação de artigos, permite deixar comentários nas publicações dos colegas, assim como partilhá-las;
- Incorporação de biblioteca e do histórico dos cursos criados;
- Salas de aula *Instructor-led Training* - Treino Conduzido por um Instrutor (ILT), dado que existem plataformas que apoiam iniciativas de aprendizagem baseadas nas que se aplicam nas próprias salas de aula (gestão de horários de salas de aula, monitorização da presença e desempenho dos alunos, por exemplo);

- *White-labeling* (quando um produto ou serviço remove a marca e o logótipo do produto final e usa a marca solicitada pelo comprador) e personalização, permitindo a criação de uma plataforma de aprendizagem totalmente única e exclusiva, mantendo a individualidade da experiência de *e-learning*;
- Certificação, quando há a possibilidade de emissão de certificados após a conclusão de cursos ou testes (útil para empresas que realizam regularmente certificações profissionais);
- Sistema que suporte integração de terceiros, permitindo que os dados estejam sincronizados com o LMS (como ferramentas de videoconferência);
- *E-commerce*, pois há plataformas que facultam a venda de cursos, acompanhando as mesmas sem a necessidade de integrar um *back-end* de comércio eletrónico;
- Notificações, mantendo os alunos atualizados, seja a nível de atribuição e inserção de novas tarefas, conclusões de seus cursos, comentários e eventos futuros;
- Ferramentas de colaboração, pois alguns LMSs oferecem recursos de aprendizagem públicos, incluindo partilha de conhecimento através de repositórios, redes sociais e fóruns de discussão.

### Utilizadores Padrão

De uma forma geral, novamente, também se considera que existem dois tipos principais de utilizadores de LMSs:

#### 1. Gestor de Cursos

Entidade responsável pela gestão do LMS, através da combinação de várias tarefas: criação dos cursos e planos de aprendizagem; atribuição de grupos de alunos específicos a planos de aprendizagem também específicos; acompanhamento do progresso dos alunos. Além da coordenação dos conteúdos, um gestor de cursos pode adicionar e remover utilizadores aos seus cursos, reuni-los em grupos e atribuir-lhes certas funções.

#### 2. Alunos

Recetores das ferramentas de aprendizagem. É para eles que estes programas se destinam. O “aluno” representa o papel mais comum. Os alunos que têm acesso ao LMS poderão ver os cursos nos quais estão inscritos, as atividades existentes em cada um e concluí-las, assim como a quaisquer outros conteúdos a eles destinados. Podem também medir o seu próprio progresso.

No caso da plataforma SMS-eSaúde, os utilizadores padrão com funções de gestão de cursos, serão os professores/psicólogos. Os utilizadores padrão com funções inerentes a um aluno, que são responsáveis por “consumirem” os conteúdos, serão os alunos e os encarregados de educação.

Posto isto, pode-se então perceber que um software deste tipo é extremamente benéfico e cumpre com o que se pretende para a plataforma SMS-eSaúde, tanto nas funcionalidades que possui, como nos utilizadores e seu papel no sistema.

### 3.2 Seleção da Plataforma *Open-Source* a Adotar

A partir do momento em que se assumiu que a plataforma a adotar seria um LMS, passou-se à fase seguinte: a de escolher o LMS, efetivamente (a plataforma que serviria de base para a plataforma SMS-eSaúde).

Assim, para chegar à plataforma a adotar, de uma forma resumida, os subseqüentes passos foram seguidos:

### 1. Selecionar LMSs

Em primeiro lugar, selecionaram-se diversos LMS *open-source*.

### 2. Comparar LMSs Selecionados

De seguida, avaliaram-se os prós e contras de cada plataforma (em função dos requisitos e respetiva prioridade) e averiguou-se de que forma, cada uma, suportaria melhor a visão do projeto SMS.

### 3. Selecionar Plataforma a Adotar

Por fim, com base nessa avaliação, uma plataforma foi selecionada.

#### 3.2.1 Identificação das Plataformas Candidatas

Detalhando os pontos atrás descritos, com base em listagens presentes em vários sites de comparação de softwares (o “Capterra”<sup>1</sup>, predominantemente), recomendação dada por utilizadores e análise de características que coincidiram com as pretendidas, as plataformas que se destacaram e acabaram, portanto, por ser selecionadas como plataformas candidatas, foram as seguintes:

- ATutor<sup>2</sup>;
- Chamilo<sup>3</sup>;
- Dokeos<sup>4</sup>;
- Forma LMS<sup>5</sup>;
- Moodle<sup>6</sup>.

#### 3.2.2 Restrição do Número de Plataformas Candidatas

Passou-se seguidamente a uma fase de eliminação, onde se compararam as características suportadas pelas plataformas referidas em cima (tabela 3).

A Dokeos foi imediatamente eliminada por apenas possibilitar a inscrição de cinco utilizadores, característica que comprometia demasiado o que se deseja para a plataforma SMS-eSaúde.

A ATutor, também foi eliminada por não ter uma interface gráfica fácil de operar e que precisa ser atualizada.

A Chamilo também foi eliminada, por não permitir a personalização dos elementos de gamificação, um fator igualmente fundamental para o projeto.

Restaram assim as plataformas Forma LMS e Moodle.

#### 3.2.3 Comparação das Plataformas Candidatas Finais e Devida Seleção

Através da listagem e categorização que tinha sido realizada (tabela 3), não era possível optar por uma delas. Para solucionar esta questão, uma vez que a escolha da plataforma está altamente relacionada com o quão bem suporta os requisitos do projeto, foi criada uma tabela para efetuar essa avaliação (tabela 4).

---

<sup>1</sup>Capterra - LMSs Open Source

<sup>2</sup>ATutor

<sup>3</sup>Chamilo

<sup>4</sup>Dokeos

<sup>5</sup>Forma LMS

<sup>6</sup>Moodle

Tabela 3: Comparação dos LMSs selecionados

Plataforma		<i>ATutor</i>	<i>Dokeos</i>	<i>Forma LMS</i>	<i>Moodle</i>	<i>Chamilo</i>
Características	<i>Open-Source</i>	X	X	X	X	X
	<i>Gratuidade</i>	X	X	X	X	X
	<i>Usabilidade</i>		X	X	X	X
	<i>Gamificação</i>			X	X	
	<i>Responsividade</i>			X	X	
	<b>Prós</b>	<ol style="list-style-type: none"> <li>Ligação à <i>Dropbox</i>.</li> <li>Considerado um LMS único dado os seus recursos de acessibilidade.</li> </ol>	<ol style="list-style-type: none"> <li>Existência de aplicação móvel.</li> <li>Possibilita criar questionários usando gráficos 2D e 3D.</li> </ol>	<ol style="list-style-type: none"> <li>Possibilita a integração de <i>plugins</i>.</li> </ol>	<ol style="list-style-type: none"> <li>Existência de aplicação móvel.</li> <li>Integração e customização de inúmeras características desejadas através de <i>plugins</i>.</li> <li>Interface simples, dinâmica e altamente configurável.</li> </ol>	<ol style="list-style-type: none"> <li>Existência de aplicação móvel.</li> <li>Compatibilidade com cloud.</li> </ol>
	<b>Contras</b>	<ol style="list-style-type: none"> <li>Sem aplicação móvel.</li> <li>Interface gráfica do utilizador é antiquada e difícil de operar e necessita ser atualizada.</li> </ol>	<ol style="list-style-type: none"> <li>Só é gratuita para cinco utilizadores.</li> </ol>	<ol style="list-style-type: none"> <li>Sem aplicação móvel.</li> </ol>	<ol style="list-style-type: none"> <li>Necessita de upgrades para memória e armazenamento.</li> </ol>	<ol style="list-style-type: none"> <li>Pesado em termos de memória.</li> </ol>
<b>Particularidades</b>	<ol style="list-style-type: none"> <li>Adequado a pessoas invisuais/ baixa visão.</li> </ol>	<ol style="list-style-type: none"> <li>LMS adequado a empresas farmacêuticas/ de saúde.</li> <li>Até ao momento, único LMS que integra <i>Shopify</i>.</li> </ol>	<ol style="list-style-type: none"> <li>Possível integrar sistema de videoconferência (criação de <i>webinars</i> e salas de aula virtuais).</li> </ol>	<ol style="list-style-type: none"> <li>LMS mais utilizado por todo o mundo.</li> </ol>	<ol style="list-style-type: none"> <li>Possibilita videoconferências <i>mobile</i>.</li> </ol>	

A tabela criada (tabela 4), é composta pelas duas plataformas candidatas finais e por todos os requisitos levantados até ao momento desta seleção. Posteriormente, fez-se uma avaliação quantitativa, atribuindo-se uma pontuação, de zero a três, relativamente à forma como cada plataforma suportava cada requisito. Zero (0) significa que a plataforma em questão não suporta o requisito e três (3), que suporta o requisito na totalidade. Assim, do melhor cenário para o pior, teríamos:

- Três (3);
- Dois (2);
- Um (1);
- Zero (0).

Por fim, somou-se, para cada plataforma, os valores atribuídos, de maneira a que cada uma ficasse com uma pontuação final que se pudesse usar para comparar. A plataforma com a pontuação mais elevada, significava que suportava melhor um maior número de requisitos e, portanto, seria essa a selecionada.

Tabela 4: Comparação do suporte aos requisitos entre as plataformas Moodle e Forma LMS

ID	Categoria	Requisito	Prioridade (MoSCoW)	Suporte do Requisito (0 - não suportar o requisito; 3 - suportar o requisito na totalidade)		Situação (Prioridade vs Suporte do Requisito)	
				FormaLMS	Moodle		
1	BIZ	Negócio (BIZ)	Gratuidade - a plataforma deverá ser gratuita.	M	3	3	Muito Favorável
1	U	Usabilidade (U)	Gamificação - a plataforma deverá suportar mecanismos de gamificação.	M	2	2	Favorável
2	U	Usabilidade	Garantir Usabilidade - a plataforma deverá ser eficaz, eficiente e garantir a satisfação do seu utilizador.	M	2	2	Favorável
1	MP	Multiplataforma (MP)	Responsividade - a plataforma deverá ser bem renderizada, independentemente do dispositivo/tamanho de ecrã que o utilizador esteja a utilizar.	M	3	3	Muito Favorável
1	F	Funcional (F)	Login - os utilizadores da plataforma poderão efetuar login na plataforma.	M	3	3	Muito Favorável
2	F	Funcional	Anonimato - a identidade dos utilizadores é mantida anónima.	M	2	3	Favorável
3	F	Funcional	Inserção de Conteúdos - os professores/psicólogos têm permissão para inserir conteúdos na plataforma.	M	3	3	Muito Favorável
4	F	Funcional	Inserção de Conteúdos em Qualquer Momento - os professores/psicólogos poderão adicionar conteúdos na plataforma em qualquer momento, mesmo em fases posteriores do projeto.	M	3	3	Muito Favorável
5	F	Funcional	Inserção de Conteúdos Direcionados - os professores/psicólogos, ao inserir os conteúdos na plataforma poderão selecionar se este se destina aos encarregados de educação ou aos alunos.	M	3	3	Muito Favorável
6	F	Funcional	Histórico de Conteúdos - tanto os alunos, como os encarregados de educação têm acesso a todos os conteúdos e tarefas que já tenham realizado/concluído.	M	3	3	Muito Favorável
7	F	Funcional	Completação Sucessiva das Tarefas - os alunos/encarregados de educação só têm acesso aos materiais disponibilizados de forma sucessiva (para se ter permissão para aceder/abrir um determinado conteúdo, o anterior terá de ser sido realizado/concluído).	M	3	3	Muito Favorável
8	F	Funcional	Conceder Acesso a Conteúdo Não Completo - os professores e psicólogos podem conceder acesso ao alunos ou encarregados de educação a sessões seguintes, mesmo que as anteriores não tenham sido realizadas/concluídas.	M	3	3	Muito Favorável
9	F	Funcional	Notificações - a plataforma deverá suportar o envio de notificações aos alunos e encarregados de educação sempre que novo conteúdo seja colocado na plataforma.	M	3	3	Muito Favorável
10	F	Funcional	Gráficos - a plataforma deverá suportar o acesso a gráficos onde os alunos poderão comparar a evolução do seu humor em relação às atividades que realizam.	M	3	3	Muito Favorável
11	F	Funcional	Fórum - a plataforma deverá suportar a existência de um fórum onde os utilizadores poderão submeter testemunhos e opiniões pessoais.	S	3	3	Muito Favorável
12	F	Funcional	Seleção das Submissões a Tornar Públicas Na Plataforma - os professores/psicólogos terão acesso às submissões realizadas no fórum e poderão selecionar as que tencionam ver tornadas públicas na plataforma.	S	3	3	Muito Favorável
13	F	Funcional	Envio de Mensagens - deve ser possibilitado o envio de mensagens de texto aos alunos/encarregados de educação.	C	0	0	Desfavorável
14	F	Funcional	Feedback - os alunos receberão feedback ao concluir uma tarefa da plataforma.	S	3	3	Muito Favorável
15	F	Funcional	Redes Sociais - os professores/psicólogos poderão partilhar a plataforma nas suas redes sociais.	W	2	2	Favorável
16	F	Funcional	Chats - os professores/psicólogos poderão criar chats aos quais os alunos e encarregados de educação poderão aceder.	W	3	3	Muito Favorável
17	F	Funcional	Progresso na Tarefa - os alunos e encarregados de educação terão possibilidade de avaliar a sua evolução, dentro de uma tarefa específica.	C	3	3	Muito Favorável
PONTUAÇÃO FINAL:					56	57	
PONTUAÇÃO MÁXIMA POSSÍVEL:					63		



Como se pode observar na tabela 4, obteve-se, para cada plataforma, pontuações muito próximas. Para além disso, fez-se uma avaliação qualitativa relativamente à situação gerada pela necessidade dos requisitos, face à forma como eram sustentados pelas plataformas e até para fundamentar a escolha destas duas plataformas. Não havendo nenhuma diferença acentuada, na comparação do suporte fornecido pelas duas plataformas, a avaliação qualitativa foi realizada englobando as duas plataformas. Para esta avaliação qualitativa, atribui-se, da melhor situação, para a pior (e respetivos cenários onde foram aplicadas):

- **Muito Favorável**

Quando:

- Se tratava de um requisito de importância máxima (M - *Must*) e era suportado ao máximo pelas plataformas (3).

- **Favorável**

Quando:

- Se tratava de um requisito de importância máxima (M - *Must*) e era suportado quase na totalidade pelas plataformas (2).
- Se um requisito de prioridade *Must* (M) fosse suportado na totalidade por uma das plataformas (3), mas a outra suportasse com o valor “2”.
- Se era um requisito de prioridade *Should* (S) e era suportado na totalidade pelas plataformas (3).

- **Desfavorável**

Quando:

- Um requisito apresenta o terceiro nível de prioridade (*Could* - C), mas que não é suportado pelas plataformas (0).

A avaliação quantitativa nas duas plataformas (cinquenta e seis pontos no caso da Forma LMS e cinquenta e sete pontos no caso do Moodle) não permitiu efetuar uma distinção entre elas. O mesmo aconteceu com a avaliação qualitativa.

Fez-se então uma nova seleção que teve como base a:

- Comparação da linguagem de programação de cada uma (a linguagem de ambas é o PHP - inviável a preferência por uma);
- Data da última versão publicada (ambas, na altura desta seleção, do ano de 2019 e com poucos meses de diferença - condição não suficiente);
- Atividade nos fóruns de ajuda (ambas com bastante atividade - inviável a preferência por uma);
- Complexidade das APIs (sem conhecimento necessário para poder optar por uma).

Novamente, esta análise não levou a conclusões.

Perante o insucesso das anteriores tentativas de seleção, utilizaram-se novos parâmetros para a seleção final, que podem ser consultados na tabela 5. Perante estes parâmetros, o Moodle acabou por ser o selecionado por reunir as seguintes vantagens em relação à plataforma oponente.

Tabela 5: Comparação final entre as plataformas Moodle e FormaLMS - parâmetros utilizados para a seleção final

	Moodle	FormaLMS
Representa uma plataforma divulgada e altamente conhecida e utilizada em todo o mundo.	X	
Possui uma ativa comunidade de suporte.	X	X
Possibilita a existência de uma aplicação móvel.	X	
Trata-se de uma plataforma familiar de muitos dos potenciais utilizadores (psicólogos e professores).	X	

Sendo o Moodle selecionado, fica-se então com o cenário apresentado na tabela 6.

### Realização de *Workshop*

Após a seleção do LMS Moodle e com o objetivo desta escolha ser aprovada, foi criado um *workshop* com o cliente, professores e psicólogos, para apresentar a solução encontrada, demonstrar o seu funcionamento e como os conteúdos poderiam ser inseridos. Visto ainda não ter sido instalada uma versão “ofical” da plataforma (por o servidor onde estaria alojada ainda não ter sido cedido), de maneira a ser possível realizar este *workshop*, foi utilizado o sistema de contentores, Docker (“*Docker Toolbox for Windows*” [16][17]).

No final deste *workshop*, o cliente aprovou a escolha efetuada.

### Considerações

De notar que, tanto na tabela 4, como na tabela 6, os requisitos referentes à existência de aplicação móvel (REQ\_MP\_02) e de aplicação de recursos de gamificação (requisitos: REQ\_F\_18, REQ\_F\_18.1, REQ\_F\_18.2, REQ\_F\_18.3, REQ\_F\_18.4, REQ\_F\_19, REQ\_F\_19.1, REQ\_F\_19.2), não foram contabilizados, por ainda não terem sido criados nesta fase.

Contudo, a sua ausência não revela qualquer impacto na análise efetuada visto que:

- O requisito “REQ\_U\_01” referente à vontade de incluir gamificação é suportado de igual forma por ambas as plataformas, sendo ambas favoráveis para a sua integração.
- A existência da aplicação móvel não foi inicialmente contabilizada como um requisito por parte do cliente. No entanto, como apresentado na tabela 5, uma vez que a plataforma Moodle possibilita a existência de uma aplicação móvel, essa foi então incluída nos requisitos.

De notar também que a avaliação efetuada foi realizada através de uma análise pouco aprofundada ao Moodle. Só um estudo detalhado desta plataforma (a sua arquitetura, design, funcionamento, etc.) permitirá fazer o mapeamento correto dos requisitos exigidos com os recursos que o Moodle fornece, como se irá ver nos capítulos 4 e 5.

Tabela 6: Suporte aos requisitos pelo Moodle

ID	Categoria	Requisito	Prioridade (MoSCoW)	Suporte (0 - não suportar o requisito; 3 - suportar o requisito na totalidade)	Situação (Prioridade VS Suporte do Requisito)
1	BIZ	Negócio (BIZ)			
		<b>Gratuidade</b> - a plataforma deverá ser gratuita.	M	3	Muito Favorável
1	U	Usabilidade (U)			
		<b>Gamificação</b> - a plataforma deverá suportar mecanismos de gamificação.	M	2	Favorável
2	U	Usabilidade			
		<b>Garantir Usabilidade</b> - a plataforma deverá ser eficaz, eficiente e garantir a satisfação do seu utilizador.	M	2	Favorável
1	MP	Multiplataforma (MP)			
		<b>Responsividade</b> - a plataforma deverá ser bem renderizada, independentemente do dispositivo/tamanho de ecrã que o utilizador esteja a utilizar.	M	3	Muito Favorável
1	F	Funcional (F)			
		<b>Login</b> - os utilizadores da plataforma poderão efetuar login na plataforma.	M	3	Muito Favorável
2	F	Funcional			
		<b>Anonimato</b> - a identidade dos utilizadores é mantida anónima.	M	3	Muito Favorável
3	F	Funcional			
		<b>Inserção de Conteúdos</b> - os professores/psicólogos têm permissão para inserir conteúdos na plataforma.	M	3	Muito Favorável
4	F	Funcional			
		<b>Inserção de Conteúdos em Qualquer Momento</b> - os professores/psicólogos poderão adicionar conteúdos na plataforma em qualquer momento, mesmo em fases posteriores do projeto.	M	3	Muito Favorável
5	F	Funcional			
		<b>Inserção de Conteúdos Direcionados</b> - os professores/psicólogos, ao inserir os conteúdos na plataforma poderão selecionar se este se destina aos encarregados de educação ou aos alunos.	M	3	Muito Favorável
6	F	Funcional			
		<b>Histórico de Conteúdos</b> - tanto os alunos, como os encarregados de educação têm acesso a todos os conteúdos e tarefas que já tenham realizado/concluído.	M	3	Muito Favorável
7	F	Funcional			
		<b>Completo Sucessiva das Tarefas</b> - os alunos/encarregados de educação só têm acesso aos materiais disponibilizados de forma sucessiva (para se ter permissão para aceder/abrir um determinado conteúdo, o anterior terá de ser sido realizado/concluído).	M	3	Muito Favorável
8	F	Funcional			
		<b>Conceder Acesso a Conteúdo Não Completo</b> - os professores e psicólogos podem conceder acesso ao alunos ou encarregados de educação a sessões seguintes, mesmo que as anteriores não tenham sido realizadas/concluídas.	M	3	Muito Favorável
9	F	Funcional			
		<b>Notificações</b> - a plataforma deverá suportar o envio de notificações aos alunos e encarregados de educação sempre que novo conteúdo seja colocado na plataforma.	M	3	Muito Favorável
10	F	Funcional			
		<b>Gráficos</b> - a plataforma deverá suportar o acesso a gráficos onde os alunos poderão comparar a evolução do seu humor em relação às atividades que realizam.	M	3	Muito Favorável
11	F	Funcional			
		<b>Fórum</b> - a plataforma deverá suportar a existência de um fórum onde os utilizadores poderão submeter testemunhos e opiniões pessoais.	S	3	Muito Favorável
12	F	Funcional			
		<b>Seleção das Submissões a Tornar Públicas Na Plataforma</b> - os professores/psicólogos terão acesso às submissões realizadas no fórum e poderão selecionar as que tencionam ver tornadas públicas na plataforma.	S	3	Muito Favorável
13	F	Funcional			
		<b>Envio de Mensagens</b> - deve ser possibilitado o envio de mensagens de texto aos alunos/encarregados de educação.	C	0	Desfavorável
14	F	Funcional			
		<b>Feedback</b> - os alunos receberão <i>feedback</i> ao concluir uma tarefa da plataforma.	S	3	Muito Favorável
15	F	Funcional			
		<b>Redes Sociais</b> - os professores/psicólogos poderão partilhar a plataforma nas suas redes sociais.	W	2	Favorável
16	F	Funcional			
		<b>Chats</b> - os professores/psicólogos poderão criar <i>chats</i> aos quais os alunos e encarregados de educação poderão aceder.	W	3	Muito Favorável
17	F	Funcional			
		<b>Progresso na Tarefa</b> - os alunos e encarregados de educação terão possibilidade de avaliar a sua evolução, dentro de uma tarefa específica.	C	3	Muito Favorável

*Esta página foi intencionalmente deixada em branco.*

## 4 Arquitetura do Moodle

Sendo o Moodle a plataforma selecionada para servir de “base” da plataforma “SMS-eSaúde”, de maneira a se poder compreender os seus componentes, a forma como estes se relacionam e como novos componentes podem ser adicionados (com vista à fase de implementação/desenvolvimento dos requisitos), considerou-se relevante apresentar a descrição arquitetural da plataforma Moodle.

### 4.1 Design e Descrição Arquitetural do Moodle

O projeto “Moodle” foi iniciado por Martin Dougiamas em 1999. A sua primeira versão (versão 1.0) foi lançada em 2002, sendo a sua arquitetura condicionada pelas tecnologias existentes na altura (PHP 4.2 e MySQL 3.23). Atualmente, o Moodle já conta com a versão 3.9.1 [18]. No entanto, utilizou-se a versão 3.8.3. para o projeto, por ser a versão mais recente no momento de instalação do Moodle no servidor adotado.

#### 4.1.1 Arquitetura 3-Tier

O Moodle segue uma arquitetura clássica 3-Tier, arquitetura definida pela decomposição de sistemas complexos em camadas. Este modelo representa a arquitetura padrão de sistemas web, sendo, assim, um sistema Cliente-Servidor. A arquitetura 3-Tier surgiu dada a necessidade de separar a lógica das aplicações, da interface do utilizador, de maneira a que os utilizadores web pudessem aceder às aplicações sem ter de as instalar nas suas máquinas locais. Assim, a lógica da aplicação, outrora contida no chamado de “*Thick Client*”, deixa de existir na máquina do utilizador. É, deste modo, movida para o servidor, sendo o navegador web (cliente) usado como um “*Thin Client*”.

No caso do Moodle e, de acordo com a figura 5, temos:

- **Cliente** - passa pelo *browser* e *software* específico utilizado para a visualização dos recursos.
- **Servidor** - servidor web com suporte a PHP, como o Apache ou o *Internet Information Services - Serviços de Informação da Internet* (IIS).

O servidor de alojamento da plataforma SMS-eSaúde foi cedido pelo *Helpdesk* do DEI da UC, especificamente uma *Virtual Machine - Máquina Virtual* (VM) com 2 *cores*, 2 GB de RAM, com *Ubuntu Server 16.04 LTS*.

As três camadas da respetiva arquitetura são, então:

#### 1. *User Interfacen Layer* - Camada de Apresentação/ Interface do Utilizador (UI)

Camada que comanda o fluxo da apresentação e a partir do qual o utilizador interage diretamente, pois é através dela que realiza pedidos (consultas, por exemplo).

#### 2. *Business Logic Layer* - Camada de Negócio/Lógica (BLL)

Camada que trata do armazenamento da “lógica” (funções e regras) da aplicação.

#### 3. *Data Access Layer* - Camada de Acesso aos Dados (DAL)

Camada cujo objetivo passa por permitir o acesso aos dados e fazer com que persistam na respetiva base de dados.

Esta divisão resulta em sistemas mais flexíveis e de mais fácil compreensão e manutenção, visto que as partes constituintes podem ser alteradas de forma independente.

Na figura 5 é possível identificar uma camada “*Moodle External API*” (Camada API Externa). Esta camada tem como objetivo aceder a métodos da API externa padrão do Moodle e do seu núcleo (*core*). É também nesta camada que são definidos os servidos para a integração móvel.

Também é possível ver a camada dos conectores de *web services* (“*WS Connectors*”). Esta é utilizada para fornecer protocolos de serviço da web adicionais e métodos de autenticação, como *Oauth*. Os “*WS Connectors*” traduzem os métodos implementados na API Externa para protocolos específicos.

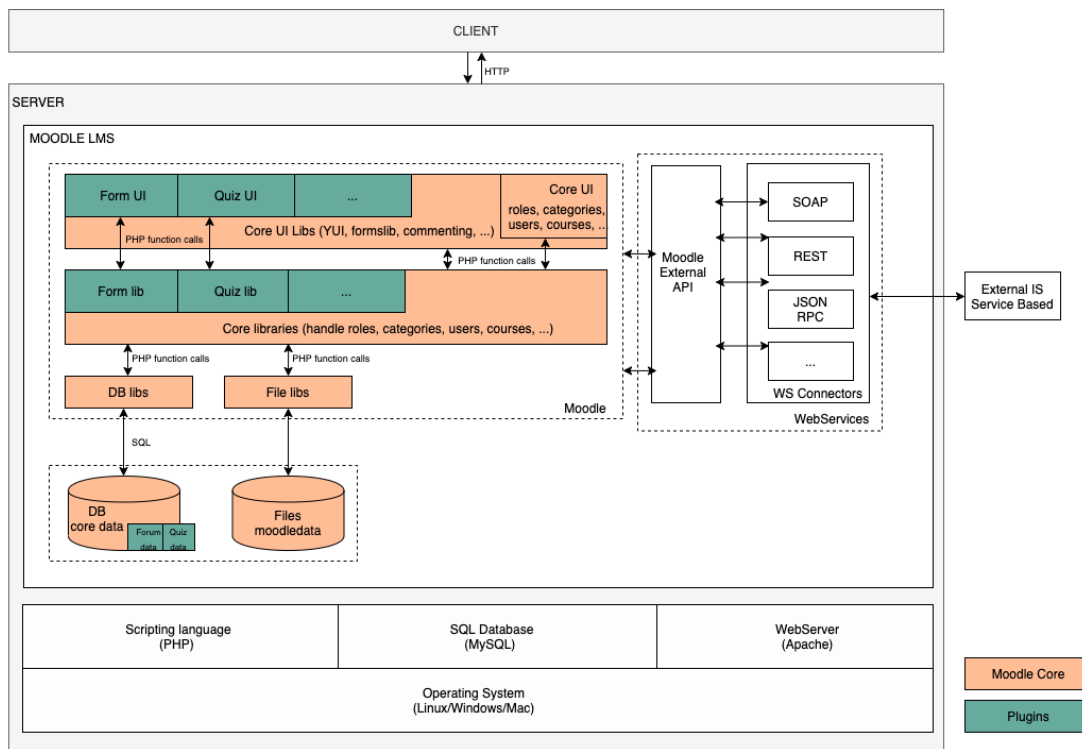


Figura 5: Representação arquitetural do Moodle

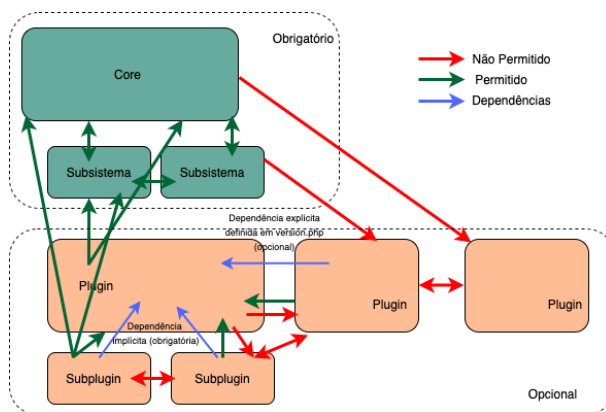


Figura 6: Representação dos componentes do Moodle, suas dependências e possível comunicação (baseado na figura 3 existente na documentação referente à apresentação da comunicação efetuada entre componentes do Moodle [19])

### 4.1.2 Componentes

Como pode ser verificado nas figuras 5 e 6, o Moodle é dividido em diferentes componentes [19]:

- **Core**

Código *core*, encontrado na pasta “lib” ou “/lib/classes/” constituído pelo conjunto de bibliotecas *core* que fornecem a base das funcionalidades nas quais as outras partes do Moodle dependem. O seu código não é opcional e não pode ser removido de forma segura sem colocar o funcionamento do Moodle em causa.

- **Subsistemas**

Conjunto de funções e classes constituintes do Moodle agrupadas logicamente e geralmente vinculadas a recursos específicos, podendo ser desativadas/ativadas.

- **Plugins**

A maior parte do código do Moodle é constituído por *plugins* (daí surge a própria sigla “M” do Moodle cujo significado é “Modular”). Os *plugins* são componentes opcionais no Moodle que têm o propósito de adicionar novas funcionalidades. Existem muitos tipos diferentes de *plugins* sendo que cada um suporta uma diferente forma de fazer “*extend*” às funcionalidades *core*.

- **Subplugins**

Suportado por alguns tipos de *plugins* do Moodle para estender as suas próprias funcionalidades.

- **Canais de Comunicação**

O código do Moodle pode ser chamado de diferentes formas, como por exemplo: chamadas de funções PHP diretas; funções externas; *observers* de eventos; templates (*Mustache*); componentes de *callback*; chamadas `get_string`.

### 4.1.3 Funcionamento

O Moodle, geralmente, é utilizado como uma aplicação independente (*standalone*), podendo ser imediatamente instalado e utilizado na máquina desejada. Contudo, possui implementações básicas de outras funcionalidades, para que também possa ser integrado com outros sistemas [20][21].

Aquando da instalação do Moodle, é de notar que este é então composto por três partes, sendo que as informações de configuração das mesmas se encontram armazenadas num ficheiro designado “`config.php`” (`$CFG->wwwroot`). As três partes referidas são [20]:

1. O **código**, no qual o servidor web não pode escrever, encontrado geralmente nas pastas `/var/www/html/moodle` ou `~/moodle/htdocs`.
2. A **pasta “moodledata”**, onde são armazenados tanto os ficheiros enviados, como os gerados pelo Moodle. Posto isto, o que se encontra nesta pasta já deve poder ser escrito e gravado pelo servidor web. Por razões de segurança não deve estar na *web root*.
3. A **base de dados**, gerida por um dos *Relational Database Management System* - Sistema de Gestão de Base de Dados Relacional (RDBMS) suportados

O Moodle adiciona um prefixo a todos os nomes de tabela de maneira a que a base de dados possa ser partilhada com outras aplicações, se for necessário (“`mdl`”).

Todas estas partes podem estar contidas num único servidor. Na situação alternativa (uma configuração *load-balanced*), cada servidor terá uma cópia do código e existe apenas uma única cópia da base de dados e da pasta “moodledata” que são partilhadas por todos os servidores.

### Despachar Pedidos

Sendo o Moodle uma aplicação web, os seus utilizadores interagem com ela através do *browser* (ou seja, o Moodle tem portanto de responder a pedidos *Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto (HTTP) [20]). Deste modo, relativamente ao *namespace Uniform Resource Locator* - Localizador Uniforme de Recursos (URL), o Moodle utiliza a abordagem PHP padrão para “despachar” os URLs para os diferentes *scripts*.

Por exemplo:

1. Para aceder à página de um curso, o URL seria:

.../course/view.php?id=2 (em que 2 representaria o ID do curso)

2. Para aceder à página de um *quiz*, o URL seria:

.../mod/quiz/view.php?id=14 (em que 14 representaria o ID do *quiz*)

Os *scripts* específicos (course/view.php e mod/quiz/view.php) lidariam então com as solicitações.

Os URLs são permanentes, não se alterando no caso de o professor alterar o nome do curso.

#### 4.1.4 Sistema de Permissões e Papéis

No Moodle, um papel [22] corresponde a uma coleção de permissões definidas e estabelecidas para todo o sistema, que podem ser atribuídas a utilizadores específicos em contextos igualmente específicos (por exemplo, num dado contexto, como um curso, um utilizador pode ter o papel de aluno nesse curso e já ter o papel de professor em outro).

A combinação de papéis, assim como o contexto, definem a capacidade (“*capability*”) de um utilizador específico poder realizar uma determinada ação dentro da página em que se encontra, sendo estas controladas pelo administrador da plataforma.

À semelhança dos papéis referidos no momento em que se identificaram as características padrão de um LMS comum (capítulo 3.1), também o Moodle tem como principais utilizadores do sistema, os seguintes tipos de utilizadores (que desempenham papéis distintos - designados de “*ARCHETYPE*”): Administrador, Gestor, Professor (Editor), Aluno, Professor (não Editor).

#### Contexto

No Moodle, um contexto [23] é um local em que as permissões são atribuídas a um utilizador (através de função).

Tal como acontece, por exemplo, numa hierarquia de pastas e respetivos ficheiros, os contextos do Moodle também são hierarquizados. No nível superior encontra-se o contexto de “Sistema”, sendo que dentro desse contexto existem outros criados de maneira a que os cursos sejam organizados: contexto de “Utilizador”, de “Categoria”, de “Bloco” e de “Módulos de Atividades e Recursos”.

Por exemplo, os contextos predefinidos para o papel de “Professor (Editor)” (accedidos através das configurações de administração e com possibilidade de serem editados) são os existentes na figura 7.



Role archetype ? ARCHETYPE: Teacher (editing) ▾

Context types where this role may be assigned

- System
- User
- Category
- Course
- Activity module
- Block

Figura 7: Contextos atribuídos de forma predefinida a um utilizador de papel “Professor (Editor)” - menu de administração

Do ponto de vista do *developer* a API [24] é bastante simples:

Contexto de Sistema:

```
1 $systemcontext = context_system::instance();
```

Contexto de Utilizador:

```
1 $usercontext = context_user::instance($user->id);
```

Contexto de Categoria:

```
1 $categorycontext = context_coursecat::instance($category->id);
```

Contexto de Curso:

```
1 $coursecontext = context_course::instance($course->id);
```

Contexto de Módulo de Atividades e Recursos:

```
1 $contextmodule = context_module::instance($cm->id);
```

Contexto de Bloco:

```
1 $contextblock = context_block::instance($this->instance->id);
```

### Verificar Permissões - *Capabilities*

Tendo o contexto, as permissões podem ser verificadas. Cada bit de funcionalidade que um utilizador pode ou não ter é chamado de *capability* [25] e a sua verificação fornece um controlo de acesso mais refinado do que apenas uma função “require\_login()”, por exemplo.

```
1 require_capability($capability, context $context, $userid = null)
```

Caso o utilizador em questão não possua a *capability*, um erro é devolvido, isto é, é lançada uma exceção de controlo de acesso, caso o utilizador não tenha a respetiva *capability*.

No caso em que se deseja determinar se um link para o *script* de uma outra página deve ser exibido, a função “has\_capability” pode ser utilizada (retorna um booleano). Essa função verifica se um utilizador tem uma *capability* específica [26] num dado contexto.

```
1 $context = context_module::instance($cm->id);
2 has_capability('mod\folder:managefiles', $context)
```

#### 4.1.5 Gerar *Output*

No Moodle, o *output* [27] é manipulado através de duas variáveis globais:

- **\$PAGE**

Variável global utilizada para rastrear o estado da página a retornar, através do armazenamento da informação da página à qual se pretende fazer *output*, ficando essa informação disponível para o código que gera *HyperText Markup Language* - Linguagem de Marcação de Hipertexto (HTML).

##### Informações Armazenadas Mais Relevantes

Contexto da página:

```
1 $PAGE->set_context( $context );
```

URL da página (ajuda a manipular URLs de forma fácil):

```
1 $PAGE->set_url( $url );
```

Título:

```
1 $PAGE->set_title( get_string( 'welcome', 'local-greet' ) );
```

Cabeçalho:

```
1 $PAGE->set_heading( $title );
```

Uma boa prática da utilização desta variável é quando se pretende lidar e adicionar ficheiros *JavaScript* desejados:

```
1 $PAGE->requires->js( '/local/js/nomedoficheiro.js ' )
```

- **\$OUTPUT**

Algumas páginas utilizam a variável global \$OUTPUT para gerar *output* (trata-se de um *render* genérico).

Um exemplo de utilização da variável \$OUTPUT passa por casos em que se pretende renderizar gráficos no Moodle (tendo em consideração a sua API [28]):

```
1 echo $OUTPUT->render( $chart );
```

Assim, as principais diferenças entre estas duas variáveis são:

1. \$PAGE serve para configurar a página e \$OUTPUT para a exibir.
2. \$PAGE contém a lógica, enquanto que o \$OUTPUT é apenas relacionado com a exibição e produção de HTML.

#### 4.1.6 Abstração da Base de Dados

O Moodle apresenta a sua própria camada de abstração da base de dados traduzida como um invólucro fino em torno das várias bibliotecas da base de dados PHP: classe moodledata. É utilizada a variável global \$DB, que fornece acesso à conexão com a base de dados, lidando com *queries*, simples *updates* e *inserts* [29].

#### 4.1.7 *Logging* e Sistema de Eventos

Os eventos são informações atômicas que descrevem algo que aconteceu no Moodle, sendo, principalmente, resultado de ações realizadas pelo utilizador [30].

Quando uma ação ocorre, um evento é criado por uma API ou *plugin* principal e o sistema de eventos divulga essas informações aos *observers* registrados para esse evento (array com propriedades como o nome do evento, *callback* e prioridade, por exemplo).

Um *observer* (também designado por *Publisher-Subscriber*) representa um padrão que define uma dependência *one-to-many* entre objetos. Assim, sempre que um objeto muda o seu estado, todos os seus dependentes são notificados e atualizados automaticamente, possibilitando que todos os objetos interessados sejam avisados sempre que ocorram mudanças de estado ou eventos acionados noutro objeto.

Assim, o sistema de eventos atua como uma “espinha dorsal” da comunicação em todo o sistema Moodle, contendo informação sobre todas as ações que foram executadas (onde aconteceram; o que aconteceu; quem as realizou), para além do nível educacional da ação (apresentados nesta secção).

Os *observers* de eventos não podem modificar os dados do evento nem interromper o envio dos mesmos, sendo um canal de comunicação unidirecional.

Algo extremamente benéfico, como será demonstrado no capítulo 6, é o facto de o Moodle fornecer e ter os seus próprios eventos já criados, mas possibilitar a criação de novos (API de Eventos [30]).

#### API de Eventos

Cada *plugin* define os seus próprios eventos que podem ser acionados (*triggered*), estendendo uma classe base abstrata para cada evento possível.

Apresentam-se de imediato algumas características dos eventos do Moodle:

##### 1. Objetos Ativos

Todas as ações significativas no Moodle (eventos) são registadas e mantidas em *logs* [31] gravados na base de dados. Quando um evento é acionado (*triggered*) e depois de ser reinstanciado (recuperado de um *log*, por exemplo), um objeto evento é capaz de fornecer funções *callback* para diversos propósitos, sendo um deles para descrever o evento.

##### 2. Inclusão Automática

As definições das classes de eventos são incluídas automaticamente quando necessário, sem a necessidade de manter listas de tipos de eventos conhecidos.

Durante o desenvolvimento, novas definições de eventos e *observers* podem ser adicionados sem a necessidade de atualização.

##### 3. Manutenção

Adicionar novos eventos, assim como migrar eventos existentes é algo relativamente simples, uma vez que toda a informação ou código relacionado com esses eventos está concentrado num ficheiro existente (local fixo).

##### 4. Auto-Documentação

O comportamento dos eventos encontra-se no mesmo local que a sua definição (único ficheiro). Este fator facilita o conhecimento, por parte do *observer* de eventos, de quais

os eventos que podem ser acionados por um determinado *plugin* (incluindo suporte para preenchimento automático e inspeções de código).

Uma lista de todos os eventos está agora disponível como um relatório para administradores e pesquisadores [32].

## 5. Estrutura de Dados Rápida e com Auto Validação

Como os eventos são objetos instanciados, o processador PHP terá de validar a sua estrutura e o tipo de classe. Isso não garante a validade dos dados, mas oferece alguma garantia de consistência. Para além disso, torna possível a deteção de erros de digitação não intencionais.

### Nível Educacional do Evento (Edulevel)

A propriedade “*edulevel*” [30] é uma das propriedades características de eventos que auxilia na definição do valor educacional do evento. Há apenas três opções disponíveis (de forma propositada para facilitar a sua escolha).

- “*Teaching*”

Qualquer evento que é realizado (normalmente por um professor) e tem valor para o ensino, isto é, que afete a experiência e ambiente de aprendizagem dos alunos.

– Exemplos de eventos válidos:

1. Um professor avaliar um aluno;
2. Um professor modificar as configurações do curso;
3. Um professor adicionar uma nova secção à página do curso;
4. Um professor modificar as configurações de um módulo;
5. Um professor adicionar uma página ao curso;
6. Um professor deixar *feedback*.

– Exemplos de eventos inválidos:

1. Um professor fazer uma publicação no fórum, que, tanto pode afetar a experiência de aprendizagem, como não, portanto, nesse caso, o professor está apenas a participar.

- “*Participating*”

Qualquer evento realizado por um utilizador, que esteja ou possa estar relacionado com a experiência de aprendizagem (geralmente aplicado aos alunos).

– Exemplos de eventos válidos:

1. Um utilizador fazer uma publicação no fórum;
2. Um utilizador enviar um ficheiro;
3. Um utilizador fazer um comentário;
4. Um utilizador conversar no *chat*;
5. Um utilizador visualizar a página de um curso.

Exemplos de eventos inválidos:

1. Um utilizador atualizar o seu perfil;

2. Um utilizador visitar o perfil de alguém;
  3. Um utilizador enviar uma mensagem a outro.
- “*Other*”

Qualquer outra ação, relacionada com a administração do site ou específica do utilizador; não tem nenhum valor educacional.

## 4.2 *Plugins*

O Moodle, sendo um OSS e à semelhança de muitos softwares deste tipo, é construído a partir de inúmeros *plugins* independentes, que trabalham em conjunto com o *core* do sistema (através de uma API bem definida) e são partilhados pelas três camadas [20], como atrás referido (figura 5).

Um *plugin*/módulo de extensão representa um programa utilizado para adicionar funcionalidades a um “programa maior”, com o intuito de estender as suas funcionalidades ou adicionar funcionalidades específicas.

O sistema do Moodle é assim formado por um *core* (núcleo), rodeado de *plugins* (“peças” customizáveis): *fat core* e *plugins* fortemente tipados (que usa variáveis e tipos específicos).

Esta divisão em “peças” revela-se extremamente benéfica na medida em que, no caso de o desenvolvimento ser colaborativo, se torna muito mais acessível alterar e acrescentar funcionalidades, consoante necessidades específicas (através dos *plugins*), podendo o sistema *core* ser atualizado durante esse processo, sem que haja qualquer conflito).

### Convenção da Nomenclatura

Um *plugin* no Moodle, assume a forma de uma pasta de ficheiros com um tipo e um nome, formando o componente “nome” do *plugin*, designado “*Frankenstyle*” [33]. Este sistema de nomes (“*Frankenstyle Component Names*”) criado por Petr Škoda é o utilizado como convenção da nomenclatura empregue para identificar exclusivamente *plugins* do Moodle (baseado no teu tipo e nome) presente em todo o código do Moodle.

Os nomes dos componentes *Frankenstyle* têm um prefixo e, de seguida, o nome da pasta, separados por um *underscore*. O prefixo é determinado pelo tipo de *plugin*. Por exemplo, o prefixo para um “Módulo de Atividades e Recursos” é “*mod*”, enquanto que para um *plugin* do tipo “Bloco” é “*block*”.

O nome é o nome da pasta do *plugin*, escrito sempre com letras minúsculas. Por exemplo, o nome de uma atividade do tipo Quiz é “*quiz*”. Assim, o nome do componente *Frankenstyle* para este módulo seria “*mod\_quiz*”.

Na tabela 7 apresentam-se alguns exemplos.

Nos exemplos representados, “*mod*” representa os tipos de *plugin* do tipo “Módulo de Atividades e Recursos”; “*block*”, *plugins* do tipo “Bloco”; “*quiz*” representa *plugins* de relatórios de uma Atividade “Quiz”; “*qtype*” representa *plugins* do tipo Pergunta.

Quanto aos subsistemas do Moodle estes não são *plugins* em si, mas podem ser referidos usando *core\_nome* do subsistema.

Na tabela 8 estão reunidos alguns exemplos.

O tipo e o nome do *plugin* determinam o caminho para a pasta do mesmo, em que o tipo fornece um prefixo e o nome da pasta, o nome do *plugin*.

Tabela 7: Apresentação de exemplos da convenção da nomenclatura adotada (*Frankenstyle*) para identificar *plugins* do Moodle

Tipo do Plugin	Nome do Plugin	Frankenstyle	Pasta
mod	lesson	mod_lesson	mod/lesson
mod	chat	mod_chat	mod/chat
block	badges	block_badges	blocks/badges
block	online_users	block_online_users	blocks/online_users
quiz	statistics	quiz_statistics	mod/quiz/report/statistics
qtype	numerical	qtype_numerical	question/type/numerical

Tabela 8: Apresentação de exemplos da convenção da nomenclatura adotada (*Frankenstyle*) para identificar subsistemas do Moodle

Subsistema Core	Frankenstyle	Pasta
Administração	core_admin	/admin
Registo do Site	core_register	/admin/registration
Backup e Restauro	core_backup	/backup/util/ui
Antivírus	core_antivirus	/lib/antivirus
Editor de Texto	core_editor	/lib/editor

### Plugins Mais Relevantes

Dependendo do tipo de funcionalidade que se deseja implementar, o tipo de *plugin* implementado (assim como a API utilizada no mesmo), será diferente. Por exemplo, um *plugin* de um Módulo de Atividades e Recursos seria muito diferente do que se pretende num *plugin* para definir novos protocolos de comunicação de *web services*.

Atualmente o Moodle conta com a existência de mais de mil setecentos e dezanove *plugins* (um número que se encontra sempre a subir), verificando-se, até ao momento, cinquenta e cinco tipos de *plugins* distintos [34], sendo que diferentes tipos de *plugins* são mais adequados para certos tipos de funcionalidade.

Apresentam-se de seguida, de forma breve, os *plugins* do Moodle considerados mais relevantes (mais utilizados):

- **Módulos de Atividades e Recursos**

Os Módulos de Atividades e Recursos (representados de acordo com a tabela 9) são tipos essenciais de *plugins* do Moodle, senão os mais essenciais, na medida em que é através destes que tanto os cursos/disciplinas, como as respetivas atividades são fornecidas [35]. Incluídos nos Módulos de Atividades e Recursos que os professores/criadores de cursos podem criar estão, por exemplo: *inquéritos*; *chat* e fórum; *quizzes* - ver ANEXO A.

Tabela 9: *Frankenstyle* de *plugins* do tipo “Módulo de Atividades e Recursos”

Nome Frankenstyle	Path
mod	/mod

A lista completa dos Módulos de Atividades e Recursos fornecidos pelo Moodle, sua breve descrição e possível utilização pode ser consultada no ANEXO A.

- **Blocos**

Pequenos blocos utilizados como ferramenta para exibição de informação, podendo ser movidos entre páginas [36]. Estes são representados de acordo com a tabela 10.

Alguns exemplos de *plugins* do tipo “Bloco” já existentes no Moodle são: calendário; utilizadores que se encontram *online*; exibição dos cursos acedidos recentemente.

Tabela 10: *Frankenstyle* de *plugins* do tipo “Bloco”

Nome Frankenstyle	Path
block	/blocks

- **Temas**

*Plugins* que possibilitam alterar a aparência do Moodle (HTML + CSS) [37]. São representados de acordo com a tabela 11.

Tabela 11: *Frankenstyle* de *plugins* do tipo “Tema”

Nome Frankenstyle	Path
theme	/theme

- **Formato de Cursos**

*Plugins* com vista a produzir formas diferentes de apresentar e organizar as atividades e recursos de um curso (representados de acordo com a tabela 12) [38].

Tabela 12: *Frankenstyle* de *plugins* do tipo “Formato de Curso”

Nome Frankenstyle	Path
format	/course/format

- **Inscrição**

*Plugin* para controlar os utilizadores matriculados/inscritos nos cursos (representados de acordo com a tabela 13 [39]).

Tabela 13: *Frankenstyle* de *plugins* do tipo “Inscrição”

Nome Frankenstyle	Path
enrol	/enrol

- **Autenticação**

*Plugins* que viabilizam a conexão com fontes de autenticação externas (representados de acordo com a tabela 14) [40].

Tabela 14: *Frankenstyle* de *plugins* do tipo “Autenticação”

Nome Frankenstyle	Path
auth	/auth

- **Repositório**

*Plugins* que possibilitam a inserção de conteúdos de repositórios externos, para o Moodle (representados de acordo com a tabela 15) [41].

Tabela 15: *Frankenstyle* de *plugins* do tipo “Repositório”

Nome Frankenstyle	Path
repository	/repository

- **Filtros de Texto**

*Plugins* que permitem alterar o texto presente no Moodle (representados de acordo com a tabela 16) [42].

Tabela 16: *Frankenstyle* de *plugins* do tipo “Filtros de Texto”

Nome Frankenstyle	Path
filter	/filter

### Instalação de *Plugins*

A instalação de novos *plugins* no Moodle pode ser realizada de duas formas:

1. Diretamente através do site (Administração do Site > *Plugins* > Instalar *plugins*) pelo *upload* de um ficheiro zip (correspondente ao *plugin*).
2. Acedendo à diretoria respetiva, tendo em conta o tipo de *plugin* que se pretende adicionar (como pode ser visto nas tabelas em cima, um *plugin* do tipo “Bloco” deveria ser incluído na pasta *blocks* e um *plugin* do tipo “Módulos de Atividades e Recursos”, na pasta *mod*).

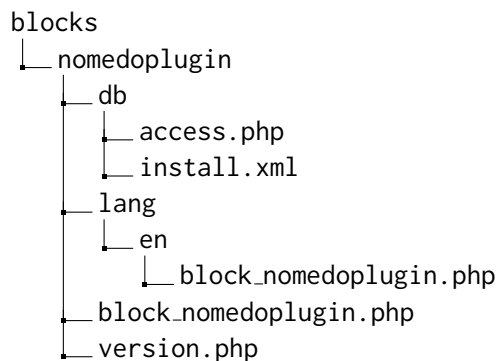
#### 4.2.1 *Plugin* Exemplo - *Plugin* do Tipo “Bloco”

Como referido, um *plugin* no Moodle assume a forma de uma pasta que, consoante o tipo de *plugin*, se encontra numa diretoria específica. A título de exemplo, apresentam-se de seguida os ficheiros predefinidos para criar um *plugin* básico do tipo “Bloco” [36]. Este tipo de *plugin* (como se apresenta na tabela 7 e na tabela 10) localiza-se na pasta *blocks* (/var/www/html/moodle/blocks ou ~/moodle/htdocs/blocks) sendo que o nome atribuído à pasta (nome dado ao *plugin*) irá influenciar o nome dos ficheiros essenciais inseridos na mesma,



uma vez que o seu nome será composto pelo *Frankenstyle* correspondente. O nome é validado pelo método “lib/classes/component.php :: is\_valid\_plugin\_name()” com um regexp: `/^[a-z](?:[a-z0-9_](?!_)) * [a-z0-9] + $/`.

De seguida, apresenta-se a estrutura de diretorias (pastas e respetivos ficheiros) necessária para criar um bloco básico. O conteúdo base dos ficheiros pode ser consultado no ANEXO B.



#### 4.2.2 Pasta “db” (blocks/nomedoplugin/db)

Pasta que contém todas as informações necessárias para instalar ou atualizar o *plugin*. É nela que se encontram dois ficheiros essenciais: “install.xml” e “access.php”.

Os Sistemas de Gestão de Bases de Dados (SGBD) diferem bastante na sintaxe SQL necessária para definir tabelas. De forma a contornar essa questão, cada *plugin* do Moodle (assim como o seu núcleo) define as suas próprias tabelas de base de dados no ficheiro XML “install.xml” (blocks/nomedoplugin/db/install.xml) - ver ANEXO B. O sistema de instalação do Moodle analisa esse ficheiro e usa as informações que contém para criar as tabelas e índices necessários. No próprio Moodle (site) existe também uma ferramenta específica para *developers* (XMLDB) para auxiliar na criação desses ficheiros de instalação.

Se houver a necessidade de alterar a estrutura de dados em versões diferentes tanto do Moodle como de um *plugin*, o *developer* terá de usar um objeto adicional que forneça métodos *Data-driven Learning* - Aprendizagem Orientada a Dados (DDL) para que os dados sejam preservados e atualizados de uma versão para a outra [20].

#### Modelo da Base de Dados

O *Extensible Markup Language Database* - Base de Dados de Linguagem de Marcação Extensível (XMLDB) armazena a definição da base de dados em formato XML, sendo utilizado para instalar o Moodle em qualquer base de dados, assim como utilizado pelo Moodle para definir o modelo da base de dados. Em cada pasta de um *plugin*, como definido em cima, existe uma pasta designada “db” onde a estrutura da base de dados de cada *plugin* é armazenada.

Anteriormente, isso costumava ser feito através de ficheiros separados “mysql.php” e “mysql.sql” (por exemplo), para diferentes bases de dados, o que implicava a criação de ficheiros igualmente diferentes para cada sistema. Caso fosse necessário existir mais bases de dados compatíveis com os *plugins* desenvolvidos, seria necessário criar bastantes ficheiros.

Passar a adotar uma abordagem para XMLDB implica apenas a criação de um ficheiro único, designado de “instal.xml”, podendo mesmo utilizar-se um editor presente das funcionalidades de administração do próprio site (Administração do Site >Desenvolvimento >Editor XMLDB), o que é bastante prático, fácil e economiza tempo. Este editor possibilita criar, editar e atualizar qualquer ficheiro “install.xml” da tabela MySQL para o formato XMLDB. Contudo, para tal, é necessário conceder permissões de escrita (chmod 777 na pasta db correspondente).

É necessário notar que o editor XMLDB não representa uma ferramenta para criar o *plugin*, mas sim, auxiliar na gestão da base de dados (tal como o phpmyadmin possibilita a gestão da base de dados SQL). Tabelas, campos, chaves e índices podem ser criados ou tabelas MySQL podem ser exportadas para o editor XMLDB.

Outra vantagem do XMLDB é possibilitar criar comentários na tabela (caso se queira deixar informação a outros *developers*).

### Definir *Capabilities*

As *capabilities* são definidas no array \$capabilities no ficheiro “access.php” (blocks/nomedoplugin/db/access.php) - ver ANEXO B.

O nome da *capability* consiste em “tipodoplugin/nomedoplugin:nomedacapability” e o *plugin* possibilita a definição de novas *capabilities* relacionadas às funcionalidades que fornece.

### 4.2.3 Pasta “lang” (blocks/nomedoplugin/lang)

O Moodle utiliza o seu próprio sistema para possibilitar a tradução das *strings* utilizadas, para qualquer idioma. O sistema baseia-se na função “get\_string()” onde as *strings* são identificadas por uma chave e pelo nome *Frankenstyle* do *plugin*.

Os diferentes idiomas são definidos por duas letras (representativas do país), por exemplo:

- **en** - inglês  
blocks/nomedoplugin/lang/en
- **en\_us** - inglês americano  
blocks/nomedoplugin/lang/en\_us
- **pt** - português  
blocks/nomedoplugin/lang/pt
- **pt\_br** - português do Brasil  
blocks/nomedoplugin/lang/pt\_br
- **fr** - francês  
blocks/nomedoplugin/lang/fr

Cada uma destas pastas contém um ficheiro (blocks/nomedoplugin/lang/en/block\_nomedoplugin.php, por exemplo) - ver ANEXO B - para possibilitar a correta tradução das *strings* utilizadas no *plugin* (neste caso, em inglês).

Esse ficheiro é composto pelo seguinte:

```
1 <?php
2 $string['pluginname'] = 'Plugin name';
3 $string['title'] = 'Title of the plugin';
4 ...
```

Sendo as *strings* referidas então como:

```
1 get_string('pluginname', 'block_nomedoplugin')
```

#### 4.2.4 Ficheiro “version.php” (blocks/nomedoplugin/version.php)

Cada *plugin* deve possuir um ficheiro “version.php” utilizado pelo *plugin installer* do sistema, para instalar e atualizar o *plugin* - ver ANEXO B.

Define, assim:

- A versão atual do *plugin*;
- A versão necessária do Moodle para poder instalar o *plugin*;
- O nível de maturidade do *plugin*:
  - MATURITY\_ALPHA;
  - MATURITY\_BETA;
  - MATURITY\_RC ou MATURITY\_STABLE (considerado pronto para sites de produção).

#### 4.2.5 Ficheiro “block\_nomedoplugin.php” (blocks/nomedoplugin/block\_nomedoplugin.php)

Ficheiro constituído pela classe “nomedoplugin”, que faz *extend* à classe “block\_base” (classe que descreve um “Bloco do Moodle” e a partir da qual todos os “Blocos” do Moodle derivam [43][44]) - ver ANEXO B.

A classe “nomedoplugin” possui o método “get\_content()” [45] que, quando chamado, popula a variável `$this->content` do bloco [46].

Esta variável contém todo o conteúdo real que é exibido dentro de cada bloco. Os valores válidos que tolera são “NULL” ou um objeto da classe “stdClass”, que deve ter variáveis de membro específicas. Normalmente, ele começa a vida com um valor de NULL e se torna totalmente construído (ou seja, um objeto) quando `get_content()` é chamado.

#### 4.2.6 Outras Convenções

##### • Criação de Eventos

Para criar eventos num *plugin*, uma pasta “classes” tem de ser criada. Dentro dessa pasta, terá também de ser criada uma pasta “event”, que irá conter o ficheiro onde o evento está definido (blocks/nomedoplugin/classes/event/nomedoficheiroevento.php).

##### • Utilização de JavaScript

Para utilizar um único ficheiro *JavaScript*, esse mesmo ficheiro pode ser criado diretamente na pasta principal do *plugin* e ter qualquer nome.

No caso de se desejar utilizar vários ficheiros JavaScript, uma pasta “js” deve ser criada com os ficheiros (.js) - blocks/nomedoplugin/js/nomedoficheiro.js.

##### • Representação Visual

Para utilizar um único ficheiro .css, de maneira a se poder estilizar os *plugins*, esse mesmo ficheiro pode ser criado diretamente na pasta principal do *plugin*, desde que tenha o nome “styles.css” (blocks/nomedoplugin/styles.css).

No caso de se desejar utilizar vários ficheiros .css, uma pasta “styles” deve ser criada na pasta principal do *plugin*. Os ficheiros .css podem então ter qualquer nome (blocks/nomedoplugin/styles/nomedoficheiro.css).

- **Imagens**

Para adicionar imagens ao *plugin*, deve ser criada, na pasta principal do *plugin*, uma pasta “pix” (blocks/nomedoplugin/pix) para as mesmas.

### 4.3 Moodle *Mobile* - Aplicação Móvel

Um aspeto positivo do Moodle, que, como referido na secção 3.2.3, se revelou um dos fatores decisivos da sua escolha (de entre todos os LMSs estudados), é o facto de oferecer uma aplicação móvel bastante completa e atual.

Para tal, o Moodle vem integrado com um *web service* projetado para aplicações móvel. Para que estes sejam ativados de maneira a possibilitar que a aplicação móvel seja executada, é necessário fazê-lo no menu de administração (Administração do Site >Aplicação móvel >Configurações para dispositivos móveis >Ativar *web services* para dispositivos móveis).

O Moodle *Mobile* é uma aplicação HTML5 que utiliza a estrutura de aplicações híbridas (*full stack*) *Ionic*, através da *framework AngularJS*. Por sua vez, a *framework AngularJS* é uma *framework* estrutural extremamente popular, que possibilita o uso de HTML como a linguagem para o *template* adotado (no caso, *mustache*), permitindo estender a sintaxe do HTML para expressar os componentes que se desejam incluir, de forma clara e sucinta.

O Moodle *Mobile* é, portanto, um cliente *web service* baseado em *Representational State Transfer* - Transferência Representacional de Estado (REST) e *JavaScript Object Notation* - Notação de Objetos JavaScript (JSON), sendo a sessão do utilizador emitida através de *tokens* únicos. É recomendado *Hypertext Transfer Protocol Secure* - Protocolo de Transferência de Hipertexto Seguro (HTTPS) e o *upload* e *download* de ficheiros é realizado através de pedidos HTTP padrão. A interação com o local de instalação da aplicação e, assim, o acesso à API do dispositivo móvel, é realizado através da *framework Cordova* (*Phonegap*).

Das características funcionais da aplicação [47], destacam-se:

- Acesso aos conteúdos mesmo quando o utilizador se encontra *offline*.
- Envio de notificações personalizadas.
- *Login* mais rápido e fácil na medida em que, após se efetuar o *login* pela primeira vez, ao aceder à aplicação novamente, o *login* não será exigido.
- Módulos de Atividades e Recursos do Moodle (ANEXO A) são totalmente compatíveis com a aplicação, sem necessidade de o *developer* precisar de desenvolver duas versões.
- Acessibilidade por parte dos professores à análise do envolvimento e participação dos alunos.

É possível modificar as funcionalidades e aspeto padrão do Moodle *Mobile* através de, por exemplo, a adição de um suporte a novos *plugins* desenvolvidos para a aplicação móvel.

#### 4.3.1 Suporte para *Plugins*

Os *plugins* do Moodle (para além dos relativos aos Módulos de Atividades e Recursos já inerentes a uma versão do Moodle não customizada) possibilitam que áreas novas do Moodle possam ser estendidas [48]. Isto é realizado essencialmente pela utilização da *framework Ionic* (*Ionic Angular*) e dos seus componentes [49].

Ademais, novos ficheiros terão de ser adicionados ao *plugin*<sup>7</sup>:

- **db/mobile.php**

Terá de ser adicionado um ficheiro “mobile.php” na pasta “db” do *plugin*, ficheiro este que incluirá as áreas que se pretendem estender, sendo composto por:

- Identificador do *Plugin* - um nome atribuído para identificar o *plugin* (único).
- *Handlers* - diferentes locais onde o *plugin* irá exibir conteúdo (nome único).
- *Display data* - apenas necessário para *plugins* que desejem apresentar o seu título, ou um *icon*, por exemplo, sendo dependente do tipo de “*Delegate*” escolhido.
- *Delegate*<sup>8</sup> - onde exibir o *link* para o *plugin* representado pelo nome do *Delegate* no qual o *plugin* será registado (CoreMainMenuDelegate, CoreCourseModuleDelegate, CoreBlockDelegate, por exemplo).
- *Method* - método na classe `\(component)\output\mobile` do Moodle, executada quando o utilizador clicar na nova opção disponível na aplicação.
- *Lang* - conjunto de IDs do pacote das linguagens utilizados por todos os *handlers*.

- **classes/output/mobile.php**

Novamente, terá de ser adicionado um ficheiro “mobile.php”, agora na diretoria `classes/output/mobile.php` do *plugin*. Representa o ficheiro onde a função *main* será criada (função que exibe o certificado emitido) e no qual serão declaradas e implementadas as funções a utilizar (classe *mobile*).

O nome das funções será o mesmo que o utilizado no ficheiro `db/mobile.php` (no campo dos métodos). Há apenas um argumento “\$args”, que representa um *array* com toda a informação enviada pela aplicação.

No que diz respeito à implementação das função, primeiramente, as permissões e *capabilities* terão de ser verificadas. Posteriormente é devolvida a informação certificada necessária para exibir o *template*.

Por fim, o(s) *template(s)* é/são renderizado(s).

- **templates/ficheiro.mustache**

Criar o *template* representa a parte mais significativa para a apresentação de um *plugin* na aplicação móvel. É o código que será renderizado nessa aplicação através de *Ionic* e dos seus componentes disponibilizados (atributos “ion-”) e componentes padrão da aplicação (atributos “core-”).

- **db/services.php**

Caso se pretenda adicionar um *Web Service* próprio, este terá de ser definido num ficheiro `db/services.php` para possibilitar o acesso a Mobile. Para tal, a seguinte linha deve ser incluída:

Há dois *arrays*: um para as funções e outro para a declaração dos serviços. O *array* das funções é composto pelo nome do *web service* (único); a classe que contém a implementação

---

<sup>7</sup>Suporte para Plugins do Moodle

<sup>8</sup>Opções de *Delegates* a Seleccionar de Acordo com o Tipo de *Plugin*

das funções e o seu *path*; o nome dos métodos (nome da função dentro da classe) e, de modo a possibilitar o acesso a Mobile, a seguinte linha também deve ser incluída:

```
1 'services' => [MOODLE_OFFICIAL_MOBILE_SERVICE, 'local_mobile']
```

O *array* dos serviços deverá incluir o nome da função do *web service*, assim como as *capabilities* necessárias.

- **externallib.php**

Ficheiro “externallib.php”<sup>9</sup> na pasta principal do *plugin*. Este é referenciado como o local para a implementação das funções para o *web service* e dos parâmetros necessários.

---

<sup>9</sup>Descrição para Serviços Externos

## 5 Identificação das Funcionalidades Adicionais

Realizada a análise da arquitetura, design e funcionamento do Moodle, será necessário proceder à identificação das funcionalidades por ele suportadas, de maneira a se poder nomear as que terão de ser adicionalmente implementadas. O presente capítulo tem essa intenção.

### 5.1 Funcionalidades Suportadas

O Moodle é uma aplicação web utilizada com o intuito de possibilitar a criação de ambientes educativos onde professores e alunos se reúnem para, respetivamente, poderem ensinar e aprender. Como referido no capítulo 4, este é constituído essencialmente por peças, os *plugins*, sendo os “Módulos de Atividades e Recursos” (ANEXO A) considerados os mais importantes, pois é através deles que os professores podem criar e gerir cursos e conteúdos, para além de permitirem definir a que utilizadores esses conteúdos se destinam.

Muitos dos requisitos desejados para o projeto são assegurados pela existência desses módulos, descritos no ANEXO A, como por exemplo: o requisito REQ\_F\_11 é assegurado pelo Módulo de Atividades e Recursos “Fórum”; o requisito REQ\_F\_16 é assegurado pelo Módulo de Atividades e Recursos “Chat”. Requisitos como o REQ\_F\_02, REQ\_F\_05, REQ\_F\_06, REQ\_F\_07, REQ\_F\_12, REQ\_F\_14 são assegurados pela manipulação das opções de configuração desses mesmos módulos, assim como por opções existentes no menu de administração do site, podendo todos esses requisitos ser cumpridos através dessas configurações efetuadas.

Dois funcionalidades de extrema importância relacionados com esses módulos (tanto a nível de configuração dos mesmos, como das suas próprias características) são:

- **Acompanhamento do progresso do aluno dentro de cada curso, assim como dentro de cada Módulo de Atividades e Recursos**

O Moodle possibilita o acompanhamento do progresso do aluno em cada curso. Na página principal do perfil do aluno há um bloco que apresenta a visão geral de todos os cursos da plataforma no qual se encontra inscrito. Esse progresso é apresentado através de uma barra de progresso de zero a cem. Para além disso, existem certos Módulos de Atividades e Recursos, como os “Quizzes” [50] que avaliam o progresso internamente. Neste caso o progresso é apresentado num bloco dentro do “Quiz”, representativo da navegação do mesmo (“Bloco de Navegação”): o número total de perguntas é apresentado e à medida que o utilizador vai respondendo às questões, esse progresso é ilustrado no bloco referido.

- **Acompanhamento da conclusão de um Módulo de Atividades e Recursos**

O acompanhamento da conclusão de Módulos de Atividades e Recursos do Moodle [51][52][53] é uma ferramenta, relacionada com o progresso, que possibilita aos professores definir os critérios através dos quais uma atividade é considerada concluída por um aluno e de que forma o aluno pode progredir num curso, sendo desse modo o seu progresso registado.

O professor responsável pelo curso, nas configurações do mesmo, se ativar a opção de acompanhamento da conclusão de atividades do Moodle, é-lhe possível, para cada atividade, seleccionar como deseja que esse registo seja mapeado:

- O professor pode dar liberdade aos alunos de poderem ser eles próprios a marcar de forma manual a *checkbox* (que aparece quando o acompanhamento da conclusão de atividades do Moodle é ativado) relativa à conclusão de uma dada atividade.
- O professor define os critérios e a *checkbox* é assinalada automaticamente quando esses critérios são respondidos.

Com base no conhecimento e configurações relatadas possíveis de atingir com as funcionalidades referidas do Moodle, criaram-se as tabelas 17 e 18, onde estão reunidos todos os requisitos da plataforma SMS-eSaúde e a forma como poderiam ser suportados pelo Moodle, de maneira a identificar os requisitos que ficam em falta.

## 5.2 Análise das Funcionalidades Suportadas e Identificação das que Necessitam ser Adicionadas

Segue-se uma análise dos requisitos ilustrados nas tabelas 17 e 18 relativos às funcionalidades que terão de ser adicionadas.

### REQ\_MP\_01 - Responsividade

A responsividade é uma característica já presente no design do Moodle, contudo, é aqui assinalada por ser um requisito que terá de estar sempre presente em todos os requisitos adicionais. Este requisito terá de ser ainda mais levado em consideração numa fase posterior (num projeto subsequente a este, focado inteiramente no aspeto visual da plataforma).

### REQ\_MP\_02 - Aplicação Móvel

Requisito já suportado pela plataforma. Contudo, embora seja um requisito de baixa prioridade (W), o seu suporte foi um dos fatores que levou à preferência pelo Moodle (por possibilitar a sua utilização).

Apesar de a plataforma poder ser acedida num *browser* (tanto num *Personal Computer* - Computador Pessoal (PC), como num telemóvel) de forma eficiente, a existência de uma aplicação móvel é desejável por se considerar ser algo que adiciona valor ao projeto, para além de, de certa forma, ser um fator que poderia melhorar a relação e adesão pelo público alvo (jovens a quem a plataforma se destina).

A aplicação móvel inerente ao Moodle já vem com inúmeras funcionalidades compreendidas, tanto para alunos, como para professores, como por exemplo: a lista dos cursos e respetivos recursos e atividades criados nos mesmos (os Módulos de Atividades e Recursos padrão apresentados na secção 5.2 e listados no ANEXO A); existência de *chats*; receção e visualização de notificações; calendário; etc).

Todavia, o que for adicionado (para poder ser usado por norma em web), terá de ser igualmente implementado para também poder ser visualizado e utilizado na aplicação móvel, visto que o Moodle não possibilita esse mapeamento direto.

### REQ\_U\_01 - Gamificação

Novamente, outro requisito suportado pelo Moodle por este já acarretar um recurso utilizado na gamificação, como a atribuição de recompensas (as designadas de *Moodle Badges*, adotadas a partir da versão 3.8.3 do Moodle [54][55]). Todavia, espera-se que a plataforma possa vir a atribuir pontos aos alunos, como descrito nos requisitos REQ\_F\_18 (REQ\_F\_18.1, REQ\_F\_18.2, REQ\_F\_18.3, REQ\_F\_18.4) e que as medalhas/recompensas atribuídas sejam dependentes desses pontos, como enunciado nos requisitos REQ\_F\_19 (REQ\_F\_19.1, REQ\_F\_19.2).

Assim, somam-se às funcionalidades adicionais os seguintes requisitos:

- REQ\_F\_18 - Pontuação

- REQ\_F\_18.1 - Receber 2 pontos

1. Após o preenchimento de uma alínea de um “jogo”.



**REQ\_F\_18.2** - Receber 5 pontos

1. Após a visualização de um vídeo (sem pergunta).
2. Após a realização de uma das tarefas diárias (“Diário de Humor” e “Diário de Atividades”).

**REQ\_F\_18.3** - Receber 10 pontos

1. Após o preenchimento de um inquérito.
2. Após visualização de um vídeo com pergunta embutida.
3. Após realização de um *quiz*.
4. Após realização de um registo semanal.

**REQ\_F\_18.4** - Receber 20 pontos

1. Ao ouvir um ficheiro de áudio.
2. Ao participar na sessão presencial.

• **REQ\_F\_19** - Recompensa

**REQ\_F\_19.1** - Receber “crachá pequeno”.

1. Ao atingir 100 pontos.

**REQ\_F\_19.2** - Receber “crachá grande”.

1. Ao atingir 200 pontos.

**REQ\_F\_10 - Existência de um “Gráfico de Humor e Atividades”**

No presente requisito, deseja-se que os alunos tenham um local na plataforma onde possam visualizar um gráfico de linhas onde os seus registos (humor e atividades) sejam apresentados, permitindo ao aluno tirar conclusões. Para tal, terão de se cumprir também os seguintes requisitos:

- **REQ\_F\_10.1** - Submeter diariamente um valor se 0 a 100 que correspondesse ao seu humor (sendo os números de 0 a 100, valores que o quantificavam);
- **REQ\_F\_10.2** - Submeter diariamente atividades extra curriculares que realizaram.

Sendo este um requisito muito específico, não integrado no Moodle, terá de ser adicionado.

**REQ\_F\_13 - Envio de Mensagens**

O Moodle já providencia formas de enviar mensagens através de um *chat*, ou de um fórum. Porém, no presente requisito, deseja-se o envio de mensagens de telemóvel (*short message servise* - SMSs). Terá de se adicionar à plataforma uma API (REST, por exemplo) que possibilite o envio de SMSs.

**REQ\_F\_15 - Redes Sociais**

No presente requisito, deseja-se que os professores/psicólogos possam partilhar a plataforma nas suas páginas de *Facebook* e *Twitter*. Terão de ser adicionados à plataforma botões de partilha disponibilizados pelas ferramentas para *developers* dessas redes sociais.

No capítulo 6 apresenta-se então o desenvolvimento destas funcionalidades.

Tabela 17: Requisitos existentes e mapeamento com as funcionalidades do Moodle

Requisito		Prioridade (MUSCOW)	Suporte pelo Moodle	Considerações Adicionais
ID	Nome			
REQ_BIZ_01	Gratuidade	M	Característica do Moodle	-
REQ_MP_01	Responsividade	M	Característica do Moodle	Apesar de ser uma característica suportada pelo Moodle, é um requisito que terá de ser sempre tido em consideração nos requisitos adicionalmente implementados.
REQ_MP_02	Aplicação Móvel	W	Característica do Moodle	Embora seja possibilitada pelo Moodle, tudo o que for implementado adicionalmente terá de ser também implementado de forma a, para além de funcionar em web, também funcionar na aplicação móvel.
REQ_U_01	Gamificação	M	Compatível com o Moodle	Compatível com o Moodle na medida em que já disponibiliza um recurso de gamificação ("Moodle Badges").
REQ_U_02	Garantir Usabilidade	M	Compatível com o Moodle	Característica do design do Moodle e assegurada pelo servidor onde a plataforma "SMS-eSaúde" se encontra alojada (servidor do DEI que, ao contrário do servidor anterior, da UC, garante o cumprimento deste requisito).
REQ_F_01	Login	M	Característica do Moodle	Possibilitado pela criação de utilizadores e atribuição dos seus respetivos papéis no menu de administração.
REQ_F_02	Anonimato	M	Característica do Moodle	Configurado no menu de administração ou possibilitada a opção de escolha aos próprios utilizadores em situações específicas (como quando pretendem fazer uma submissão no Fórum, por exemplo).
REQ_F_03	Inserção de Conteúdos	M	Característica do Moodle	Cumprido através da criação de cursos e respetivos Módulos de Atividades e Recursos (escolhido a partir do tipo de conteúdo que se pretende inserir)
REQ_F_04	Inserção de Conteúdos em Qualquer Momento	M	Característica do Moodle	Cumprido uma vez que o Moodle nunca proibe a inserção de cursos e respetivos Módulos de Atividades e Recursos (escolhido a partir do tipo de conteúdo que se pretende inserir), apenas é necessário que haja permissões para tal.
REQ_F_05	Inserção de Conteúdos Direcionados	M	Característica do Moodle	Cumprido através do <i>enrolment</i> de utilizadores nos devidos cursos e Módulos de Atividades e Recursos a eles destinados.
REQ_F_06	Histórico de Conteúdos	M	Característica do Moodle	O Moodle possibilita a visualização de todos os conteúdos, desde que esteja associado ao mesmo (através do <i>enrolment</i> de Atividades).
REQ_F_07	Completa Sucessiva das Tarefas	M	Característica do Moodle	Genda e configurada pelo criador dos cursos (e administrador da plataforma).
REQ_F_08	Conceder Acesso a Conteúdo Não Completo	M	Característica do Moodle	Possível de ser genda e configurada pelo criador dos cursos (e administrador da plataforma).
REQ_F_09	Notificações	M	Característica do Moodle	Possível de ser configurada pelo criador dos cursos (e administrador da plataforma).

Tabela 18: Requisitos existentes e mapeamento com as funcionalidades do Moodle - continuação

ID	Requisito		Prioridade (MuSCoW)	Suporte pelo Moodle	Considerações Adicionais
	Nome				
REQ_F_10	Existência de um gráfico de Humor e Atividades	M	-	Necessário Implementar	
REQ_F_10.1	Diário de Humor	M	-	Necessário Implementar	
REQ_F_10.2	Diário de Atividades	M	-	Necessário Implementar	
REQ_F_11	Fórum	S	Característica do Moodle	Cumprido através da criação do Módulo de Atividades "Fórum"	
REQ_F_12	Seleção das Submissões a Tornar Públicas	S	Característica do Moodle	Cumprido através da gestão por parte do criador do Módulo de Atividades "Fórum" (e administrador da plataforma).	
REQ_F_13	Envio de Mensagens	S	-	Necessário Adicionar	
REQ_F_14	Feedback	S	Característica do Moodle	Proporcionado após se completar certos Módulos de Atividades e Recursos como um "Quiz", por exemplo.	
REQ_F_15	Redes Sociais	W	-	Necessário Adicionar	
REQ_F_16	Chats	W	Característica do Moodle	Cumprido através da criação do Módulo de Atividades "Chat".	
REQ_F_17	Progresso na Tarefa	C	Característica do Moodle	O Moodle mantém o registo do progresso dos alunos em cada curso através de percentagem. Para além disso, certos Módulos de Atividades como "Quizzes", também exibem o progresso no mesmo.	
REQ_F_18	Pontuação	M	-	Necessário Adicionar	
REQ_F_18.1	Atribuição de dois pontos	M	-	Necessário Adicionar	
REQ_F_18.2	Atribuição de cinco pontos	M	-	Necessário Adicionar	
REQ_F_18.3	Atribuição de dez pontos	M	-	Necessário Adicionar	
REQ_F_18.4	Atribuição de vinte pontos	M	-	Necessário Adicionar	
REQ_F_19	Recompensa	M	-	Necessário Adicionar	
REQ_F_19.1	Atribuição de "crachá pequeno"	M	-	Necessário Adicionar	
REQ_F_19.2	Atribuição de "crachá grande"	M	-	Necessário Adicionar	

*Esta página foi intencionalmente deixada em branco.*

## 6 Desenvolvimento das Funcionalidades Adicionais

Como referido no capítulo 5, as funcionalidades adicionais foram identificadas por:

1. Ser necessário realizar algum tipo de configuração para que possam ir ao encontro do que se pretende exatamente;
2. Ter de se encontrar formas de as suportar, adicionando ou modificando soluções que já existem para, de novo, ir ao encontro do desejado;
3. Ser necessário desenvolver “de raiz”.

### 6.1 Plano para Integração das Funcionalidades Adicionais

O Moodle representa uma plataforma existente há dezanove anos, sendo este fator um aspeto extremamente benéfico: quer pelos seus anos de contínua evolução e adaptação às necessidades dos seus utilizadores, como pela sua vasta documentação, fundamental essencialmente para utilizadores mais específicos como é o caso de *developers*.

Ao ser desenvolvido em colaboração com uma comunidade virtual formada por utilizadores de todo o mundo (programadores, *developers*, professores, *designers*) e de o próprio Moodle ser uma plataforma extremamente personalizável, com inúmeros recursos padrão, constitui uma plataforma imensamente completa.

De acordo com o enunciado no capítulo 4 e, uma vez que os requisitos enunciados representam funcionalidades muito específicas a ser adicionadas ao “programa maior”, considera-se que a maneira mais acessível e, sobretudo, sustentável de as adicionar ao Moodle é através da sua incorporação sob a forma de *plugins*.

Como referido na secção 4.2, existem atualmente cinquenta e cinco tipos de *plugins* diferentes no Moodle. Será, portanto, fundamental escolher o tipo de *plugin* correto, para que a funcionalidade adicional em questão seja adequadamente incorporada.

### 6.2 Ambiente de Desenvolvimento

Estando a plataforma “oficial” instalada e configurada no servidor do DEI requisitado, encontra-se então disponível para que o cliente, professores e psicólogos possam começar a inserir conteúdos. Com possíveis utilizadores que podem, nesta fase, já utilizar a plataforma, não seria de todo uma opção viável que o processo de desenvolvimento fosse realizado na plataforma “oficial”. Desse modo, de maneira a assegurar a sua integridade e proteção, evitando falhas devido a questões alheias, considerou-se importante arranjar uma maneira que garantisse a máxima eficiência no processo de desenvolvimento.

Assim, foi necessário instalar e configurar uma plataforma que representasse uma versão para desenvolvimento e realização de testes (ambiente de desenvolvimento) íntegra e que se assemelhasse o mais possível com a plataforma “oficial” (emular o ambiente de produção o mais semelhante possível).

A primeira estratégia foi a de utilizar uma versão do *Docker* (tal como se utilizou no *workshop* realizado e já referido na secção 3.2.3, por ainda não ter sido disponibilizado um servidor da UC ou do próprio DEI onde a plataforma “oficial” estivesse instalada), pela sua portabilidade, *deployment* e teste contínuo, segurança e isolamento.

Ao utilizar “*Bitnami Moodle Stack Containers*” [56], nomeadamente através do *Docker*, a ideia seria:

1. Correr o servidor do *Docker* onde se fariam as edições;
2. Após se “obter o pretendido”, utilizar a estratégia de instalação de *plugins* em que se faz *upload* da pasta *zipada* na plataforma “oficial”;
3. Usar links simbólicos para possuir sempre os ficheiros na máquina pessoal.

Contudo, pela ocorrência de alguns problemas na manipulação e edição do código, nesse contexto, abandonou-se essa opção.

Assim, adotou-se uma nova estratégia: utilizar os instaladores nativos do *Bitnami* (“*Bitnami Moodle Stack Installers*” nomeadamente) [57]. Esta solução também se revela benéfica pois:

- Automatiza a configuração da pilha das aplicações oferecidas (tanto em Windows, como em macOS e Linux);
- Inclui todo o software necessário para que possa ser imediatamente utilizado e de forma bastante simples;
- Possibilita o desenvolvimento e teste numa aplicação específica em ambiente local (no contexto deste projeto, utilizou-se macOS) de forma independente, sem interferir com o software do sistema.

Assim, seguem-se os principais passos utilizados no processo de desenvolvimento no ambiente selecionado para esse efeito:

1. Iniciar a pilha do Moodle presente na máquina pessoal (Mac);
2. Efetuar “*mount*” aos volumes de dados da pilha exportados;
3. Criar links simbólicos entre as pastas na máquina pessoal (de maneira a possuir sempre os ficheiros na mesma) e as pastas e ficheiros desejados nos volumes de dados montados, assim como no GitHub (para a correta gestão, manipulação e controlo das versões);
4. Aceder ao *browser* através do endereço IP da aplicação correspondente cedido, de maneira a poder visualizar essas alterações;
5. Após se “obter o pretendido” (*plugin* referente ao requisito desejado), utilizar a estratégia de instalação de *plugins* através do *upload* da pasta *zipada* na plataforma “oficial”.

Após se ter instalado o ambiente de desenvolvimento, e de maneira a ter acesso a mensagens de *debug* na plataforma, foi necessário, no menu de administração, ativar as mensagens de *debug* para *developers* (Administração do Site >Desenvolvimento >*Debugging* >PROGRAMADOR: Mensagens de erro adicionais para programador).

### 6.2.1 Papéis e Permissões Criados

Aquando da instalação da pilha Moodle, o único utilizador existente desempenhava o papel de “Administrador” do sistema.

Uma vez que todas as funcionalidades adicionais envolvem a interação de utilizadores com papel no sistema de “Aluno” e “Professor” (Criador de Cursos), foram criados utilizadores deste tipo para tornar o processo de desenvolvimento e respetivos testes efetuados ao longo desse processo, mais fidedigno. Um utilizador com o papel de “Gestor” também foi criado, embora não tenha sido utilizado. Todos estes utilizadores podem ser vistos na figura 8.

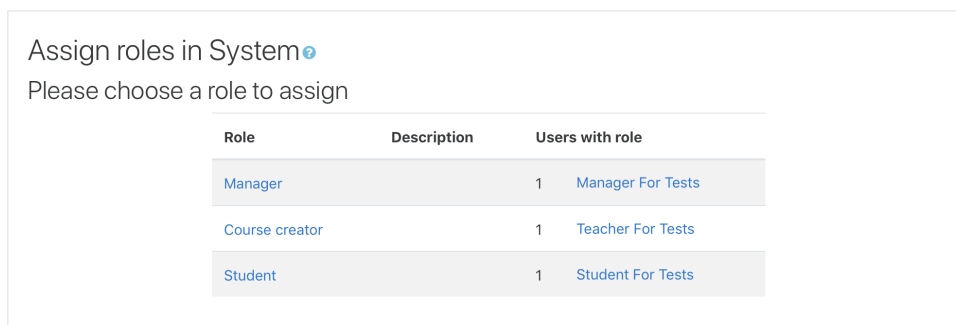


Figura 8: Utilizadores criados no ambiente de desenvolvimento

### 6.3 Processo de Desenvolvimento e Estratégias Adotadas

O processo de desenvolvimento consistiu essencialmente na exploração de formas que possibilitassem a incrementação das funcionalidades adicionais desejadas à plataforma. Como referido anteriormente (secção 6.1), a integração das funcionalidades adicionais seria efetuada pela adição de *plugins*. No entanto, consoante o requisito a implementar, estratégias diferentes foram aplicadas.

#### 1. Estratégia Um

Criação de raiz dos *plugins* que correspondessem às funcionalidades adicionais desejadas.

Neste caso, foi necessário perceber como os *plugins* seriam criados, utilizando-se as APIs disponibilizadas pelo Moodle. Para isso foi necessário o seu estudo e a aprendizagem do seu funcionamento e implementação.

#### 2. Estratégia Dois

Instalação de *plugins* já existentes que:

- Permitissem tirar proveito das suas funcionalidades e, através do estudo do seu código, possibilitassem a inserção, em locais específicos, de novo código para, assim, corresponder na totalidade com as funcionalidades adicionais desejadas.
- Permitissem tirar proveito da sua total capacidade sem que fosse necessário adicionar novo código.

Em ambas as estratégias foi sempre dada a prioridade às funcionalidades e configurações já predefinidas pelo Moodle, se estas representassem soluções viáveis para o cumprimento dos respetivos requisitos.

Tabela 19: Estratégias adotadas nos requisitos a desenvolver

REQ_F_10	Estratégia 1
REQ_F_18	Estratégia 2
REQ_F_19	Estratégia 2
REQ_MP_02	Estratégia 1
REQ_F_13	Estratégia 2
REQ_F_15	Estratégia 2

Na tabela 19, apresentam-se as estratégias utilizadas por cada requisito a desenvolver.

O processo de desenvolvimento teve então início pelo requisito REQ\_F\_10 por influenciar os requisitos REQ\_MP\_02, REQ\_U\_01, REQ\_F\_18 e REQ\_F\_19.

### 6.3.1 Requisitos “REQ\_F\_10”

Os requisitos “REQ\_F\_10”, compreendem um conjunto de requisitos que, tal como indicado na secção 2.6, passam, essencialmente por:

1. Fornecimento de um local onde os alunos, diariamente e, apenas uma vez, indiquem, de forma quantitativa (numa escala de zero a cem), como se sentiam naquele dia - REQ\_F\_10.1;
2. Fornecimento de um local onde os alunos preenchem as tarefas extra-curriculares que realizaram nesse mesmo dia - REQ\_F\_10.2;
3. Criação e exibição de um gráfico formado pelas submissões realizadas pelos alunos nesses locais (valor do humor submetido e número de atividades realizadas) - o próprio REQ\_F\_10.

#### Estratégia Adotada

Seguindo as estratégias apresentadas, em primeiro lugar, achou-se pertinente explorar os *plugins* já existentes de maneira a procurar aqueles que fossem ao encontro das funcionalidades desejadas, ou que, facilmente, pudessem ser adaptados para ir ao encontro das mesmas. No caso de tal não acontecer, implementar-se-ia, então, os *plugins* de “raíz”.

Após essa exploração, compreendeu-se que não havia nenhum *plugin* já existente que pudesse ser aproveitado para o presente requisito e, portanto, tomou-se a decisão de o implementar na totalidade. Isto porque este requisito promove funcionalidades muito específicas. *Plugins* existentes que até pudessem ser adaptados de maneira a alcançar o desejado, iriam ser bastante complexos: perder-se-ia mais tempo a “eliminar” tudo o que fosse desnecessário e, depois, adaptar o que pudesse ser aproveitado, do que iniciar e criar um *plugin* mesmo do princípio. Assim se percebe que para o desenvolvimento deste requisito se abordou pela Estratégia 1.

#### Seleção do Tipo de *Plugin* a Desenvolver

A intenção do cliente que serviu de motivo para a existência deste requisito, seria de que os alunos observassem neste gráfico que o número de atividades que realizavam num determinado dia era diretamente proporcional ao valor atribuído para quantificar o humor nesse mesmo dia (demonstrando idealmente que, quanto mais atividades realizassem, melhor seria o resultado manifestado no seu humor). Seria, portanto, um requisito que diria respeito aos alunos.

Sendo de preenchimento diário, seria benéfico tê-lo na página principal/*dashboard*, de modo a que o acesso fosse mais imediato e fácil (evitando que o aluno se esquecesse de o preencher e/ou a excessiva navegação entre páginas).

Sabe-se que qualquer sistema de software que exija ou possibilite a sua interação / manuseamento pelo utilizador, deve ter mecanismos que otimizem o tempo dispendido pelo mesmo na realização de uma qualquer tarefa que deseje ver cumprida (usabilidade).

Assim, possibilitar o acesso a este *plugin* de forma “imediata” (só exigindo o *login* na plataforma) considerou-se uma decisão acertada.

Uma vez que o cliente desejava que estes três componentes (diário de humor, diário de atividades e gráfico gerado), embora relacionados, fossem independentes e acionados em alturas diferentes no decorrer da intervenção (Projeto SMS) realizada nas escolas, decidiu-se que seriam



desenvolvidos três *plugins*, para esse efeito. Assim, foi necessário selecionar o tipo de *plugin* escolhido.

Dadas as características apresentadas do requisito:

- exigir exibição de informação,
- exigir que essa informação fosse apresentada na página inicial do perfil do utilizador,
- poder ser acionados de modo a só aparecer no momento específico que se deseje,

considerou-se que seria benéfico utilizar o *plugin* de tipo “Bloco”, [36] (secção 4.2.1) para o desenvolvimento deste requisito.

Assim, criaram-se os *plugins*:

1. *Plugin* “daily\_mood” do tipo bloco - REQ\_F\_10.1;
2. *Plugin* “daily\_activities” do tipo bloco - REQ\_F\_10.2;
3. *Plugin* “mood\_chart” do tipo bloco - REQ\_F\_10.

O *plugin* “mood\_chart”, embora referente ao requisito REQ\_F\_10, foi criado após a criação dos *plugins* para os requisitos REQ\_F\_10.1 e REQ\_F\_10.2, por estar dependente dos dados neles submetidos.

### 6.3.1.1 Requisito “REQ\_F\_10.1”

Tabela 20: Identificação *Frankenstyle* - *plugin* “daily\_mood”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	daily_mood	block_daily_mood	blocks/daily_mood

Em primeiro lugar, para criar o respetivo *plugin*, criou-se um bloco básico, seguindo a estrutura enunciada no capítulo 4.2.1 e exemplificada no ANEXO B, designado “daily\_mood” (tabela 20).

Na diretoria blocks/daily\_mood/lang/, para além da pasta e ficheiro /en/block\_daily\_mood.php, criou-se a pasta e ficheiro /pt/block\_daily\_mood.php, para possibilitar a correta tradução das *strings* utilizadas no *plugin* também em português (idioma utilizado no Projeto SMS).

Uma vez que neste *plugin* se pretendia exibir um *range slider* no qual o aluno poderia marcar e submeter um valor (tal como se faz num formulário), utilizou-se a “*Form API*”<sup>10</sup> para possibilitar a definição e manipulação de dados [29] inseridos pelo utilizador através de formulários web.

O *form* é então instanciado no ficheiro da pasta principal, designado “block\_daily\_mood.php” (blocks/daily\_mood/block\_daily\_mood.php) na função get\_content() predefinida, concretamente através de:

```
$mform = new simplehtml_form();
```

O formulário é então exibido através da função “render()”:

```
$this->content->text = $mform->render();
```

<sup>10</sup>API do Moodle para Formulários

## Ficheiro “block\_daily\_mood.php” (blocks/daily\_mood/block\_daily\_mood.php)

```

1 <?php
2 defined( 'MOODLEINTERNAL' ) || die();
3
4 //Include simplehtml_form.php
5 require_once( 'simplehtml_form.php' );
6
7 class block_daily_mood extends block_base {
8
9     public function init() {
10         //Plugin title through get_string() function – The array key is the string
            identifier and the value is the string text defined in the given language
11         $this->title = (string) get_string( 'title', 'block_daily_mood' );
12     }
13
14     public function get_content() {
15         global $USER;
16
17         if ( $this->content !== null ) {
18             return $this->content;
19         }
20
21         $this->content = new stdClass;
22
23         // Check if user already submitted today humor
24         $humor = $this->get_today_humor();
25
26         if( $humor ) {
27             $this->content->text = '<div class="today-humor"><p>' . get_string( '
            your_today_humor', 'block_daily_mood', $humor ) . '</div>';
28         }
29         else {
30             //Instantiate simplehtml_form
31             $mform = new simplehtml_form();
32
33             //Form processing and displaying is done here
34             if ( $fromform = $mform->get_data() ) {
35                 //In this case you process validated data. $mform->get_data()
            returns data posted in form.
36                 $this->content->text = '<div class="humor-saved"><p>' . get_string
            ( 'humor_saved', 'block_block_daily_mood' ) . '</div>';
37
38                 $this->save_humor( $mform->get_data()->humor );
39
40                 $mform->set_data( $mform->get_data() );
41             } else {
42                 $this->content->text = $mform->render(); //displays the form
43             }
44         }
45         return $this->content;
46     }
47 ...

```

O próprio *form* é então criado/estruturado num novo ficheiro designado de “simplehtml\_form.php”.

De maneira a se poder incluir os elementos característicos de um formulário no *plugin*, foi necessário criar uma classe que faria *extend* à classe “moodleform” definida no ficheiro “forms-lib.php” (utilizando a variável global \$CFG, que assegura as definições de configuração comumente usadas):

**\$CFG->libdir/formslib.php**

**Ficheiro “simplehtml\_form.php” (blocks/daily\_mood/simplehtml\_form.php)**

```

1 <?php
2 require_once("$CFG->libdir/formslib.php");
3
4 class simplehtml_form extends moodleform {
5
6     public function definition() {
7         global $CFG;
8
9         $mform = $this->_form; //underscore is mandatory
10
11         MoodleQuickForm::registerElementType('slider',
12                                             "$CFG->dirroot/blocks/daily_mood/slider.
php",
13                                             'SliderSelector');
14
15         //Add elements to form
16         $mform->addElement('slider', 'humor');
17
18         $this->add_action_buttons(false, get_string('button_save_humor', '
block_daily_mood'));
19     }
20     ...
21 }

```

Através da API fornecida, é possível utilizar “*Simple Forms*”, correspondentes a formulários simples para introdução e submissão de texto. Contudo, uma vez que se desejava um *range slider*, utilizou-se sim elementos de formulários customizáveis (*Custom Form*<sup>11</sup>). Para além disso, utilizou-se um método neste ficheiro fornecido pelo “moodleform” para adicionar botões associados a uma ação (no final do formulário), sendo a funcionalidade padrão desse botão a de guardar alterações/submeter o formulário:

**\$this->add\_action\_buttons()**; (um parâmetro booleano possibilita especificar se se deseje incluir um botão de cancelamento, para além de incluir o rótulo para o botão de submissão)

Para poder adicionar e criar então esses elementos de formulário personalizáveis, houve necessidade de se utilizar um “*Quick Form*”, instanciando-se diretamente a classe `HTML_QuickForm_element`. Isto ficou então definido num novo ficheiro designado “slider.php”.

**Ficheiro “slider.php” (blocks/daily\_mood/slider.php)**

```

1 <?php
2 defined('MOODLE_INTERNAL') || die();
3
4 global $CFG;
5 // Load the main classes
6 require_once("$CFG->libdir/pear/HTML/QuickForm/input.php");
7
8
9 class SliderSelector extends HTML_QuickForm_input{
10
11     public function __constructor($elementName=null, $elementLabel=null,
    $attributes=null) {
12
13         HTML_QuickForm_element::__construct($elementName, null, $attributes);
14     }
15
16     function toHtml(){
17         $output = 'output_' . $this->getAttribute('id');

```

<sup>11</sup>API do Moodle para Formulários Customizados

```

18     return $this->_getTabs() . '<div class="diario_slider" ><input type="range
19     " min="0" max="100" ' .
20     'oninput=" ' . $output . '.value = ' . $this->getAttribute('id') . ' ' .
21     'value" ' .
22     $this->_getAttrString($this->_attributes) . ' /><br /><output class="
    dh_output" id=" ' . $output . '">50</output></div>';
    }
}

```

Uma vez que no respetivo requisito se pretendia que houvesse apenas uma submissão por dia, houve necessidade de, ao se efetuar uma submissão, para além do valor do humor submetido, guardar informação sobre o dia, mês e ano da submissão, assim como do *timestamp* (por questões de histórico caso, nos testes, houvesse necessidade de apagar vários registos no mesmo dia), sendo tudo isto obtido através da classe `DateTime`<sup>12</sup> do PHP (e referida na “*Time API do Moodle*” [58]) e do respetivo construtor “`get_server_timezone_object()`” dessa API:

```
1 $now = new DateTime("now", core_date::get_server_timezone_object());
```

Os dados seriam guardados e acedidos através das funções disponíveis para aceder a dados na base de dados do Moodle, funções essas que fornecem um alto nível de abstração e garantem que a manipulação da base de dados funcionará em diferentes RDBMSs. Isso é possível através do objeto `$DB` (capítulo 4.1.6) utilizado em novas funções criadas, `get_today_humor()` e `save_humor($humor)` - ficheiro `blocks/daily_mood/block_daily_mood.php`:

```
1 $result = $DB->get_record('block_daily_mood', $data);
1 $result = $DB->insert_record('block_daily_mood', $data);
```

Também se utilizou a variável global `$USER` para identificar o id do utilizador, de maneira a se associar o humor submetido ao respetivo utilizador.

```
1 $data['user_id'] = $USER->id;
```

O ficheiro com o modelo de dados foi definido na seguinte diretoria: `blocks/daily_mood/db/install.xml`.

Assim, a tabela da base de dados foi criada com o mesmo nome dado ao *plugin* (`block_daily_mood`), contendo as colunas: `id` (chave primária), `user_id` (id do utilizador), `humor` (valor do humor submetido), `year` (ano da submissão), `month` (mês da submissão), `day` (dia da submissão), `timestamp` (*timestamp* da submissão).

Em anexo é possível verificar a estrutura final do *plugin* em questão (ANEXO C).

### 6.3.1.2 Requisito “REQ\_F\_10.2”

Tabela 21: Identificação *Frankenstyle* - *plugin* “daily\_activities”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	daily_activities	block_daily_activities	blocks/daily_activities

Inicialmente o cliente deu liberdade para decidir de que forma seria desenvolvido o *plugin* para o requisito em questão. O requisito REQ\_F\_10.2 inicial tinha apenas a especificação de haver um local no perfil do aluno que lhe permitisse inserir as atividades extra curriculares que

<sup>12</sup>Classe `DateTime` do PHP

realizou no dia corrente, sendo que o número de atividades realizadas apareceria no *plugin* do gráfico (REQ\_F\_10 - *plugin* “*mood\_chart*”). Assim, a solução encontrada foi a de possibilitar a inserção das atividades pelo aluno utilizando as APIs centrais apresentadas na documentação de desenvolvimento do Moodle, em concreto, novamente, a “*Form API*” [59] (desta vez com um “*Simple Form*”).

Assim, o *plugin* desenvolvido (“*daily\_activities*”) iniciou-se com a estrutura padrão para a criação de *plugins* do tipo bloco (tabela 21), à qual foi acrescentado um ficheiro php (classe *simplehtmlform* que herdava os métodos da classe pai, nomeadamente e, de novo, a classe *moodleform*). Nesse ficheiro foi criada a estrutura do formulário constituído por entradas de texto, onde cada entrada corresponderia a uma atividade. Em conjunto com o cliente, considerou-se que um número razoável de entradas a preencher deveriam ser dez e, portanto, esse foi o número estipulado para o máximo de atividades possíveis de ser submetidas.

```

1 ...
2     $mform->addElement('text', 'atividade1', get_string('label_atividade', '
3     block_atividades', 1)); //Add element to form
4     $mform->setType('atividade1', PARAMTEXT); //Set type of element -
PARAMTEXT is used for cleaning data that is expected to contain multi-lang
content.
5
6     for ($i=2; $i <= 10; $i++) {
7         $name_input = 'atividade' . $i;
8         $previous_input = 'atividade' . ($i - 1);
9         $mform->addElement('text', $name_input, get_string('label_atividade',
'block_atividades', $i));
10        $mform->setType($name_input, PARAMTEXT);
11    }

```

O texto colocado, num dado dia, em cada entrada do formulário pelo aluno corresponde a uma atividade. Após serem submetidas, as atividades apareceriam sempre de novo no formulário, mesmo que se fizesse *refresh* ao *browser*, mas apenas se tivessem sido inseridas e submetidas nesse dia (novamente através do uso e manipulação da classe *DateTime* do PHP e do construtor “*get\_server\_timezone\_object()*” da “*Time API* do Moodle”). Após o dia, o formulário ficaria de novo limpo para poder ser utilizado no dia seguinte (e assim sucessivamente). Isto para permitir que o aluno tivesse a liberdade de, ao longo de cada dia, ir acrescentando novas atividades e manter visíveis as submissões já efetuadas de modo a poder ter acesso a elas (para não as duplicar) e as editar, se tal pretendesse (sendo as atualizações/edições realizadas expressas no gráfico).

Para além disso, um ficheiro *javascript* foi criado para facilitar a experiência do utilizador: o formulário continha apenas uma entrada visível. Novas entradas só seriam adicionadas (máximo de dez, como referido), caso o utilizador aluno tivesse escrito algo na anterior.

```

1 $PAGE->requires->js(new moodle_url('/blocks/atividades/atividades.js'));

```

Assim, havendo uma entrada visível, uma segunda ficaria visível caso o aluno tivesse escrito algo na primeira; uma terceira ficaria visível caso o aluno tivesse escrito algo na segunda; e assim sucessivamente.

Após a criação do presente *plugin* (ANEXO C), procedeu-se ao processo de validação por parte do cliente com uma demo. Embora, do ponto de vista de experiência do utilizador, a solução apresentada fosse bastante simples, o cliente desejou efetuar alterações, ou seja, modificar o *plugin* desenvolvido. Este fator levou à necessidade de refazer trabalho. No entanto, deu possibilidade ao cliente de visualizar exatamente aquilo que pretendia. Esta modificação levou a que os critérios de aceitação do presente requisito fossem alterados, para dar início ao desenvolvimento de um novo *plugin*.

Estas foram:

- Apresentar o *form* sob a forma de uma tabela em que cada linha da tabela corresponderia a uma atividade e as colunas aos dias da semana.
- Os dias da semana apresentados seriam os dias da semana atual (de segunda a domingo), sendo alterados quando uma nova semana se iniciava.
- O aluno marcaria com um *check* em cada dia da semana as atividades que realizaram.
- Haveria um número maior do máximo de atividades a inserir (cinquenta).
- A tabela iniciar-se-ia com um número base de atividades (entradas), mas o aluno teria a possibilidade de adicionar linhas à tabela e também de as apagar.

### Segunda Versão do *Plugin* “*daily\_activities*”

A estrutura da tabela foi então criada, a forma como se adicionou os formulários foi mantida, acrescentando os elementos do tipo *checkbox* e um novo botão (ficheiro “*table\_structure.php*”).  $\$max$  representa o número máximo de atividades que aparecem na tabela, embora seja possível adicionar mais linhas, como se verá mais adiante.

#### Ficheiro “*table\_structure.php*” (*blocks/daily\_activities/table\_structure.php*)

```

1 ...
2 $max = ($number > 5) ? $number : 5;
3 ...
4 //TABLE CREATION
5 for ($i=0; $i < 50; $i++){
6     if($i >= $max) {
7         $mform->addElement('html', '<tr id = the_lines style="display:none"><td>');
8         //Add html element
9     }
10    else {
11        $mform->addElement('html', '<tr id = the_lines><td id= the_lines_first>');
12        //Add html element
13    }
14    $name_input = 'activity[' . $i . ']';
15
16    //CREATE TEXT INPUT
17    $mform->addElement('text', $name_input); //Add text element to form
18    $mform->setType($name_input, PARAMTEXT); //Set the type
19    $mform->addElement('html', '</td>');
20
21    //CREATE CHECKBOXES
22    for ($j=0; $j < 7; $j++){
23        $mform->addElement('html', '<td id = "check_cells">'); //Add html element
24        $checkbox_name = 'day_of_week[' . $i . '][' . $j . ']';
25        $mform->addElement('checkbox', $checkbox_name, '', '', "onclick="
26        getRealistic()"'); //Add checkbox element
27        $mform->addElement('html', '</td>'); //Add html element
28    }
29 ...

```

A primeira linha da tabela é então formada pelos dias da semana corrente, sendo os elementos de  $\$dates$  determinados na função *seven\_days*( $\$day$ ) e o elemento  $\$day$  determinado na função *week\_begin*().

```

1 ...
2 $day = $this->week_begin();
3
4 $mform->addElement('hidden', 'week_begin', $day->getTimestamp()); //Add hidden

```

```

    element
5 $mform->setType('week_begin', PARAMRAW); //Set element type - PARAMRAW (no
    cleaning whatsoever)
6 ...
7 $dates = $this->seven_days($day);
8
9 $mform->addElement('html', '<table id="myTable" class="responstable"><tr>
10 <th class = "activities">'. 'List of Activities' . '</th>
11 <th class = "check_days" id="lblWeekDay1">Seg, ' . $dates[0] . '</th>
12 <th class = "check_days" id="lblWeekDay2">Ter, ' . $dates[1] . '</th>
13 ...
14 </tr>');
15 ...
16
17 public static function week_begin() {
18     $now = new DateTime("now", core_date::get_server_timezone_object());
19     $now->setTime(0, 0, 0);
20     $week_day = ((int)$now->format('N')) - 1; // Number between 0 and 6
21     $now->sub(new DateInterval('P'. $week_day . 'D'));
22     return $now;
23 }
24
25 public static function next_day($date) {
26     $date->add(new DateInterval('PID'));
27     return $date;
28 }
29
30 public static function seven_days($day) {
31     $tmp = clone $day;
32     $dates = [$tmp->format('j M')];
33     for ($i=0; $i < 6; $i++) {
34         $tmp = table_structure::next_day($tmp);
35         $dates[] = $tmp->format('j M');
36     }
37     return $dates;
38 }
39 ...

```

A última linha da tabela é constituída pelo número de atividades selecionados em cada dia, exibidos através do ficheiro *javascript* e da função `countCheckboxes()`, que conta o número de atividades *checkadas* em cada coluna (cada dia):

```

1 $PAGE->requires->js(new moodle_url('/blocks/daily_activities/js/daily_activities.
    js'));

```

O respetivo ficheiro *javacript* também contém a função para, através do *click* num botão html "+", adicionar novas linhas (função `addRow()`).

```

1 $mform->addElement('html', '<button class = "btn-2" type="button" onclick="addRow()
    ">+</button>');

```

A estrutura da base de dados da primeira versão do *plugin* desenvolvido para o cumprimento do requisito REQ\_F.10.2, era apenas composta por uma tabela com as dez atividades (atividade1, atividade2...), o id do aluno correspondente e o *timestamp* correspondente à submissão.

Visto que o número de entradas aumentou cinco vezes, manter a estrutura de uma coluna por atividade na base de dados (cinquenta colunas) não seria de todo uma solução prática. Assim, eliminou-se a tabela da base de dados anterior, para dar lugar a duas novas tabelas na base de dados:

- Tabela “*daily\_activities*”

Formada por quatro colunas: o id (chave primária), o id do aluno (*user\_id*), o número (*number*) e o nome (*name*).

- Tabela “*daily\_records*”

Formada, igualmente, por quatro colunas: id (chave primária), o id da atividade (*activity\_id*), o id do aluno (*user\_id*) e o *timestamp*.

Assim, as atividades ficaram estruturadas da seguinte forma:

```

1 stdClass Object
2 (
3     [activity] => Array
4     (
5         [0] => Listen to a Podcast
6         [1] => Workout
7         [2] =>
8         [3] =>
9         [4] =>
10        [5] =>
11    )
12    [day_of_week] => Array
13    (
14        [0] => Array
15        (
16            [1] => 1
17        )
18        [1] => Array
19        (
20            [2] => 1
21            [3] => 1
22        )
23    )

```

A estrutura e apresentação final do *plugin* pode ser verificada no ANEXO C.

### 6.3.1.3 Requisito “REQ\_F\_10”

Tabela 22: Identificação *Frankenstyle* - *plugin* “mood\_chart”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	mood_chart	block_mood_chart	blocks/mood_chart

Em primeiro lugar, para o desenvolvimento do respetivo *plugin*, estudaram,-se *plugins* do Moodle já existentes que integrassem gráficos [60][61][62], o que não se revelou muito útil pois cada um utilizava estratégias distintas<sup>13</sup> <sup>14</sup>, não integrando soluções já providenciadas pelo Moodle (o que se decidiu que seria sempre a preferência, como referido no capítulo 5).

Ao se pesquisar por métodos de o próprio Moodle viabilizar a exibição de gráficos, o que seria uma estratégia benéfica por se utilizar funcionalidades do próprio Moodle (sem necessidade de integrar ou utilizar ferramentas externas), encontrou-se a “*Chart APP*” [28].

<sup>13</sup>Libreria “*HighchartsJS*”

<sup>14</sup>*YUI Charts*



A API de gráficos (*Core Chart API*) é um conjunto de métodos destinados a fornecer uma interface simples e moderna para gerar gráficos dinâmicos.

A primeira etapa para criar um novo gráfico passa por se criar uma instância do tipo de gráfico desejado, sendo que a presente interface do Moodle possibilita a criação de gráficos do mais diverso tipo: gráficos de barras; gráficos de linhas; gráficos *pie*; gráficos *doughnut*.

Para este requisito desejava-se que o gráfico criado fosse um gráfico de linhas suaves <sup>15</sup>.

```

1 ...
2 //CHART
3 $chart = new \core\chart_line();
4 $chart->set_smooth(true);
5 ...

```

O gráfico criado foi então formado por duas linhas distintas: valores submetidos no “Diário do Humor”; número de atividades submetido no “Diário de Atividades”. Os valores submetidos no “Diário do Humor”, assim como o número de atividades submetido no “Diário de Atividades” são obtidos diretamente da base de dados das tabelas “block\_daily\_mood” (do *plugin* “daily\_mood” - REQ\_F.10.1) e “block\_daily\_records” (*plugin* “daily\_activities” - REQ\_F.10.2), respetivamente.

```

1 ...
2 $data1_array = $this->get_humor_series();
3 $data2_array = $this->get_activities_series();
4
5 $data1 = new core\chart_series('Humor', $data1_array);
6 $data2 = new core\chart_series('Atividades', $data2_array);
7 $data2->set_yaxis(1);
8 $data2->set_type(\core\chart_series::TYPELINE);
9
10 ...
11
12 $chart->add_series($data2);
13 $chart->add_series($data1);
14 ...

```

Assim, no presente gráfico, os valores existentes no eixo dos *yy* foram representados, do lado esquerdo, com valores entre zero e cem (número mínimo e máximo definidos para a escala de humor), enquanto que, no lado direito, se definiram os valores de zero a cinquenta (número mínimo e máximo de atividades realizadas diariamente).

```

1 ...
2 $yaxis = $chart->get_yaxis(0, true);
3 $yaxis->set_min(0);
4 $yaxis->set_max(100);
5
6 $yaxis2 = $chart->get_yaxis(1, true);
7 $yaxis2->set_min(0);
8 $yaxis2->set_max(50);
9 $yaxis2->set_position($yaxis2::POS_RIGHT);
10 ...

```

Em relação ao eixo dos *xx*, definiu-se que este seria formado pelas datas da semana corrente. O valor mais à direita do eixo, representa o dia atual, sendo que haveria seis dias à sua esquerda que representam os seis dias passados.

```

1 ...
2 $chart->set_labels($this->get_labels());
3 ...

```

---

<sup>15</sup>API de Gráficos de Linhas - Moodle

Uma vez que todos os gráficos são renderizados através do método `render()` localizado no ficheiro “`outputrenderers.php`” (`moodle/htdocs/lib/outputrenderers.php`), para se exibir o gráfico (objeto), esse método tem de ser chamado.

```
1 ...
2 $this->content->text = $OUTPUT->render($chart, false);
3 ...
```

A estrutura final do respetivo *plugin* pode ser consultada no ANEXO C.

### 6.3.2 Requisitos “REQ\_F\_18”

No requisito REQ\_F\_18 tencionava-se que o utilizador possuísse um local no seu perfil onde pudesse ver a sua pontuação, sendo que esta aumentava sob determinadas condições.

#### Estratégia Adotada

Tendo em conta a natureza do requisito REQ\_F\_18 (e seus sub-requisitos, nomeadamente, os requisitos REQ\_F\_18.1, REQ\_F\_18.2, REQ\_F\_18.3 e REQ\_F\_18.4) e o facto de se ter encontrado *plugins* já existentes na diretoria de *plugins* do Moodle que, dadas as suas características, serviriam de uma boa base para o seu cumprimento, foi utilizada a Estratégia 2.

Recorreu-se então a quatro *plugins*:

1. *Plugin* do tipo “Bloco” - `block_xp` (“*Level up! - Gamification*”) [63]

Tabela 23: Identificação *Frankenstyle* - *plugin* “`block_xp`”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	xp	block_xp	blocks/xp

2. *Plugin* do tipo “Módulo de Atividades e Recursos” - `mod_videotime` (“*Video Time*”) [64]

Tabela 24: Identificação *Frankenstyle* - *plugin* “`mod_videotime`”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>mod</i>	videotime	mod_mod_videotime	mod/videotime

3. *Plugin* do tipo “Módulo de Atividades e Recursos” - `mod_hvp` (“*Interactive Content - H5P*”) [65]

Tabela 25: Identificação *Frankenstyle* - *plugin* “`mood_hvp`”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>mod</i>	hvp	mod_hvp	mod/hvp

4. *Plugin* do tipo “Módulo de Atividades e Recursos” - `mod_attendance` (“*Attendance*”) [66]

Tabela 26: Identificação *Frankenstyle* - *plugin* “`mood_attendance`”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>mod</i>	attendance	mod_attendance	mod/attendance

Os quatro são *plugins* independentes sendo que no presente capítulo são descritos de maneira a justificar o motivo tanto da sua escolha, como das alterações efetuadas com vista ao cumprimento dos requisitos.

Por o “*Level up! - Gamification*” se considerar o *plugin* central do requisito “REQ\_F.18” (onde estará assente, assim como todos os seus sub-requisitos), segue-se de imediato a sua exposição:

### Plugin “*Level up! - Gamification*”

Pode dizer-se que este *plugin* do Moodle representa um dos mais utilizados (terceiro *plugin* do tipo Bloco com mais *downloads* - atualmente com cerca de 13000), estando presente na lista dos vinte e um *plugins* com resultados mais benéficos quando se trata de *e-learning*. Isto porque representa o *plugin* do Moodle mais completo no que torna à aplicação dos recursos aplicados na gamificação (figura 9).

Por ser um *plugin* tão apelativo e ao se estudar com mais profundidade as suas características e o seu funcionamento, decidiu-se que a sua adoção seria vantajosa.

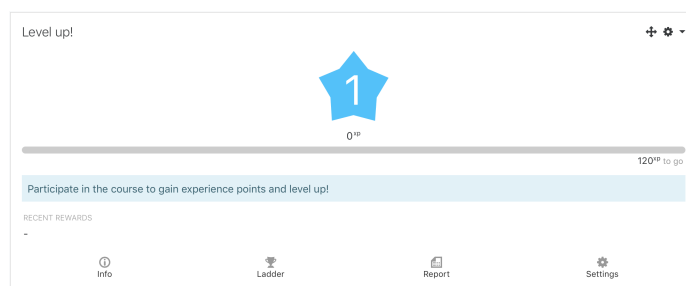


Figura 9: *Plugin* “*Level up! - Gamification*” (block\_xp) após ser adicionado à página inicial do administrador

- Características

“*Level up! - Gamification*” é então um *plugin* do tipo bloco cujas funcionalidades passam por:

1. Possibilitar a adição de gamificação à plataforma Moodle pela atribuição de pontos aos alunos (utilizadores “aluno”) de forma automática, tendo em conta as suas ações;
2. Exibir o nível atual (em função da pontuação) em que o utilizador se encontra, assim como quantos pontos são necessários para se atingir o nível seguinte;
3. Fornecer relatórios com a visão geral dos níveis em que os alunos se encontram (benéfico para os professores);
4. Enviar notificações para felicitar os alunos à medida que sobem de nível;
5. Fornecer uma tabela de classificação apresentando a classificação dos alunos;
6. Possibilitar o anonimato dos alunos nas tabelas de classificação (em conformidade com o REQ\_F.02);
7. Possibilitar a personalização do número de níveis, dos pontos necessários para subir um nível e da sua aparência;
8. Fornecer total controlo sobre os pontos ganhos por ação;
9. Desbloquear conteúdos quando um determinado nível é atingido [67];

10. Permitir que os pontos ganhos possam ser relativos aos cursos (e módulos de atividades e recursos neles existentes) ou para todo o site;
11. Suportar inclusão de *shortcodes* (que podem ser utilizados nos editores de texto do próprio Moodle para incluir ou modificar conteúdos, contudo, é necessário a instalação de um *plugin* “Shortcodes” do tipo Filtro [68] para os ativar).
12. Ser compatível com *General Data Protection Regulation* - Regulamento Geral sobre a Proteção de Dados (GDPR) (através dos próprios requisitos de privacidade do Moodle [69][70]).
13. Ser compatível com a versão do Moodle utilizada.

É possível perceber, de facto, que se trata de um *plugin* extremamente completo que facilmente pode ser manipulado para ir ao encontro do que se pretendia ver cumprido com os requisitos em questão.

- Como se Processa a Atribuição de Pontos

Como se enunciou nos pontos 1, 6, 7 e 9 da lista de características apresentada em cima, uma das funcionalidades do *plugin* em estudo é a possibilidade de definir as regras para a atribuição de pontos aos alunos.

Devendo a atribuição de pontos ser configurada ou pelo administrador, gestor, ou professores da plataforma, a forma como as regras de atribuição de pontos são definidas e estruturadas, é realizada no próprio site do Moodle, na opção de configuração do *plugin* em questão.

Para definir essas regras há dois elementos que necessitam de atenção:

- Condições
- Ações

O *plugin* observa todas as ações desempenhadas pelos alunos, atribuindo-lhes determinados pontos quando essas ações coincidem com aquelas que se encontram na lista de regras e estão em conformidade com as condições definidas. Essas condições são então (figura 10):

**Todas** as condições inseridas são verdadeiras.

**Qualquer uma** das condições inserida é verdadeira.

**Nenhuma** das condições inseridas é verdadeira.

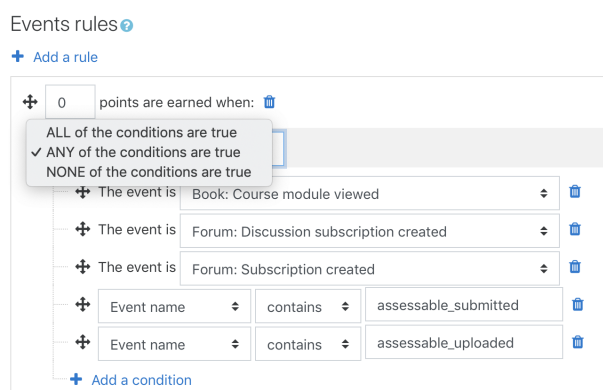


Figura 10: *Plugin* “Level up! - Gamification” (block\_xp) - níveis das condições para atribuição de pontos

Quanto às ações, tal como foi explicado no capítulo 4.1.7, a forma como o Moodle sabe e mantém o histórico das ações que foram desempenhadas por todos os utilizadores, é através de eventos [30] e da sua deteção quanto são acionados. Todos os eventos podem ser consultados no próprio site pelo administrador em Administração do Site >Relatórios >Lista de Eventos (*Log*).

- Eventos Predefinidos Reconhecidos pelo *Plugin* - Condição de Utilização

O *plugin* “*Level up!*” só reconhece eventos do contexto de módulo de atividades (constituindo esses eventos as ações para a definição das regras de atribuição de pontos). A única exceção encontrada é apenas um evento do contexto do sistema (*core*), relacionado com cursos/disciplinas (neste caso, a visualização dos mesmos). Todos estes eventos são apresentados no ANEXO D. Por outro lado, este *plugin* ignora todos os eventos que:

- São de um contexto diferente de “curso” ou “módulo”;
- Têm um nível educacional (*edulevel*) diferente de “*Participating*”;
- Estão sinalizados como anónimos (realização de um inquérito em modo anónimo, por exemplo);
- São iguais e foram acionados num curto espaço de tempo. Isto é possível através de um tempo mínimo definido para que ações iguais possam ser identificadas. O “*Level up! - Gamification*” considera idênticos os eventos que ocorrem no mesmo contexto e objeto.

- Como Adicionar Novos Eventos

Como explicado, o *plugin* “*Level up!*” apenas reconhece eventos associados aos cursos e seus módulos de atividades e recursos. Isto representa uma limitação para o nosso projeto, na medida em que se identifica (de acordo com os requisitos do projeto) a necessidade de reconhecimento de outro tipo de eventos.

Todavia, como se referiu, existe um tipo de evento distinto que o *plugin* reconhece: um evento ao nível do sistema.

Pensou-se então que, uma vez que o *plugin* “*Level up!*” tem, na sua pré definição, capacidade de reconhecer este evento, também será possível que reconheça outros eventos diferentes.

Deste modo, estudou-se o código do “*Level up!*” de maneira a encontrar o local no seu código onde essa exceção era tornada possível. Ao fazê-lo, encontrou-se o ficheiro “*event\_lister.php*” (*blocks/xp/classes/local/rule/event\_lister.php*), onde na função “*get\_core\_events()*” é possível adicionar novos eventos.

Esta função retorna um *array* onde as chaves são constituídas pelos nomes dos eventos e os valores pelas classes dos eventos.

#### Ficheiro “*event\_lister.php*” (*blocks/xp/classes/local/rule/event\_lister.php*)

```

1 protected function get_core_events () {
2     global $CFG;
3     $eventclasses = [ get_string ('course') => [
4         '\\core\\event\\course_viewed' ]
5     ];
6
7     return $eventclasses;
8 }
```

### 6.3.2.1 Requisito “REQ\_F\_18.1 - Atribuição de dois pontos”

No presente requisito pretende-se que sejam atribuídos dois pontos ao aluno sempre que preencha uma pergunta de um “jogo”. Contudo, o “jogo”, como é chamado pelo cliente, não passa de um pequeno *quiz*. Assim, avaliando os eventos detetados pelo “*Level up!*”, os eventos considerados mais adequados são os referentes ao módulo de atividades “*Quiz/Teste*” (ANEXO D). Ao se fazer uma análise aos “jogos” (*quizzes*) criados pelo cliente, já devidamente pensados e estruturados, verificou-se que todos eram formados por sete perguntas.

Uma vez que:

- não existe nenhum evento (nem eventos *core*) relativo ao preenchimento de alíneas num *quiz*,
- no projeto SMS se tencionar recompensar os alunos mesmo que não respondam bem às questões (com as respostas consideradas corretas), mas sim que participem,
- todos os “jogos” (*quizzes*) criados pelo cliente são formados por sete perguntas,
- ser possível definir que um módulo de Atividade “*Quiz/Teste*” apenas pode ser submetido quando todas as perguntas foram existentes no mesmo forem respondidas (secção 5.1),

decidiu-se que, utilizando o evento “Tentativa de teste submetida” (ANEXO D) e associando a este evento a pontuação de 14 pontos (dois pontos por sete perguntas existentes em cada “jogo”), o requisito REQ\_F\_18.1 seria cumprido, pois seria possível ter a garantia de que todas as perguntas tinham sido respondidas. O evento selecionado foi, então (figura 11):

```
\mod_quiz\event\attempt_submitted
```

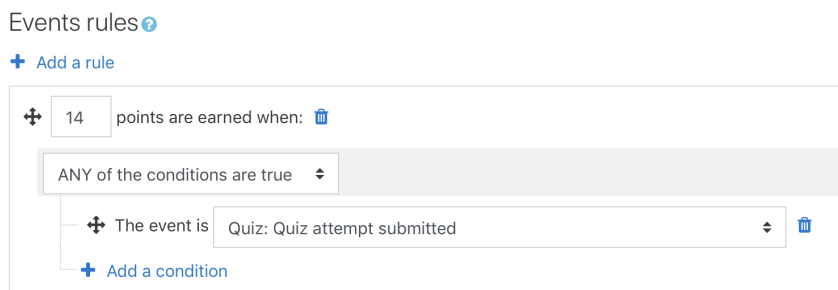


Figura 11: *Plugin “Level up! - Gamification” (block\_xp)* - opção de regra para o cumprimento do REQ\_F\_18.1

### 6.3.2.2 Requisito “REQ\_F\_18.2 - Atribuição de cinco pontos”

#### 1º Caso - Aluno Efetua *Login* Na Plataforma

Facilmente se consegue perceber que o evento correspondente à ação de um utilizador efetuar *login* na plataforma é representado por um evento *core* do Moodle (em vez de eventos relacionados com cursos e seus módulos de atividades e recursos como até a este ponto).

Ao se explorar então os eventos *core*, encontrou-se o evento pretendido:

```
\core\event\user_loggedin
```

Assim, este evento foi adicionado à função “*get\_core\_events()*” do ficheiro “*event\_lister.php*”,

para dessa forma poder ser reconhecido pelo *plugin*.

#### Ficheiro “event\_lister.php” (blocks/xp/classes/local/rule/event\_lister.php)

```

1 protected function get_core_events() {
2     global $CFG;
3     $eventclasses = [ get_string('course') => [
4         '\\core\\event\\course_viewed' ]
5     ],
6     'Login' => [ '\\core\\event\\user_loggedin' ]
7 ];
8     return $eventclasses;
9 }

```

De modo a ser reconhecido pelo “*Level up! - Gamification*”, foi necessário tornar este evento (user\_loggedin) num evento de nível educacional (edulevel) “*Participating*” (uma vez que se tratava de um evento de edulevel “Other” e, como referido, esse tipo de eventos não é reconhecido pelo *plugin*).

#### Ficheiro “user\_loggedin.php” (lib/classes/event/user\_loggedin.php)

```

1 <?php
2 namespace core\event;
3
4 defined('MOODLE_INTERNAL') || die();
5
6 //User login event class
7 class user_loggedin extends base {
8     ...
9     protected function init() {
10         $this->context = \context_system::instance(); //Define context
11         $this->data['crud'] = 'r';
12         $this->data['edulevel'] = self::LEVEL_PARTICIPATING; //Define
13         edulevel
14         $this->data['objecttable'] = 'user';
15     }

```

## 2º Caso - Aluno Vê Vídeo Sem Pergunta

O Moodle fornece formas de adicionar ficheiros de vídeo (.mov e .mp4), sendo o “VideoJS” *player* o *media player* padrão.

Um vídeo pode ser acrescentado a qualquer módulo de atividades e recursos do Moodle. Dessa forma, se um vídeo fosse colocado num módulo de atividades e recursos qualquer, como por exemplo, num “Trabalho” e se se definisse o evento associado à visualização de um vídeo como “Módulo de disciplina visualizado” (\mod\_assign\event\course\_module\_viewed), não havia uma maneira imediata de garantir que o conteúdo lá inserido, e posteriormente visualizado, seria de facto um vídeo.

Desse modo, decidiu-se adicionar ao Moodle um módulo de atividades e recursos exclusivo para a inserção de vídeos.

O *plugin* selecionado foi então o “*Video Time*” [64], cujas características estão em conformidade com os requisitos:

- Assegura que o conteúdo visualizado é mesmo um vídeo, pois só possibilita a inserção de conteúdo desse tipo.
- O vídeo fica visível na própria plataforma sem necessidade de haver um reencaminhamento para outra página.

- Extremamente fácil de utilizar pois ao criar o módulo, basta colocar o URL do vídeo (URL de um vídeo “Vimeo”, o que não vai contra os requisitos da plataforma).

O evento selecionado para o cumprimento deste requisito, adicionado e detetado automaticamente pelo *plugin* “*Level up!*” por ser um *plugin* do contexto de módulo de atividades, foi o evento “Módulo de curso visualizado”:

```
\mod_videotime\event\course_module_viewed
```

### Concluir Vídeo para Atribuir Pontos

Algo que se ponderou na fase de desenvolvimento deste requisito, embora não esteja especificado nos critérios de aceitação dos mesmos, foi se se devia detetar se um vídeo teria sido visto até ao fim para, só aí, atribuir os pontos.

Considera-se que não há uma maneira efetiva de controlar se um aluno viu ou não viu um vídeo até ao fim. O aluno pode passar o vídeo à frente, ganhando pontos na mesma ou até deixar o vídeo a “correr” e não estar a prestar atenção ou até mesmo a fazer outra atividade. A única forma fidedigna de se verificar se um aluno tinha visto o vídeo, seria o de fazer, no final do vídeo, uma ou várias perguntas sobre o mesmo. Só no caso de o aluno acertar é que se atribuiria pontuação (algo que não seria viável para este requisito pois não se pretende adicionar questões).

No entanto, para o caso de se vir a desejar integrar esta monitorização, pensou-se em mecanismos que pudessem ser utilizados.

Através do “*Video Time*” e fazendo-se esse *plugin* valer do *Vimeo*, o acompanhamento da conclusão de visualização poderia ser feito pelo facto de o *Vimeo* fornecer uma API para *developers* muito rica (*JavaScript API* sendo que a biblioteca é designada *Froogaloop*)[71][72] que fornece o necessário para avaliar o progresso de um vídeo. A biblioteca é semelhante à do *jQuery* por permitir agrupar objetos DOM em objetos *Froogaloop* às quais se aplicará as funções da API a esse objeto.

Todavia, esse plano foi abandonado por o cliente considerar que a marcação numa *checkbox* que indicasse que o vídeo tinha sido visto, seria suficiente. Isso seria possível recorrendo à funcionalidade de acompanhamento da conclusão de uma atividade do Moodle (secção 5.1), através de duas opções:

- Primeira opção:

O professor definiria que o módulo de atividade “*Video Time*” só estaria concluído se o aluno o tivesse visto (evento `\mod_videotime\event\course_module_viewed`). Desse modo, uma *checkbox* seria preenchida automaticamente, indicando a conclusão da atividade, o que faria com que fossem atribuídos os pontos.

- Segunda opção:

Seria dada a liberdade ao aluno de, quando tivesse visto o vídeo, marcar a atividade como completa. Para tal, seria apenas necessário incluir o evento `\core\event\course_completion_updated` que indicaria (que a atividade tinha sido concluída) ao *plugin* “*Video Time*” e conjugá-lo com o evento referido na opção anterior.

Uma vez que o “*Level up! - Gamification*” não atribui pontos quando um evento igual a outro é acionado num curto espaço de tempo, os alunos não poderiam fazer batota, nem receber pontos a mais, se tivessem a ideia de, de forma consecutiva, atualizar o estado da *checkbox* referente à conclusão da atividade (marcar e desmarcar a *checkbox*).



Desse modo adicionou-se o novo evento no respetivo *plugin*, para dar a possibilidade de, em conjunto com o evento identificado anteriormente, caso o cliente pretenda, utilizar as funcionalidades disponíveis e fornecidas pelo Moodle para assinalar a conclusão de uma tarefa (capítulo 5.1), sem necessidade de utilizar ferramentas ou configurações externas. Este evento faria *extend* à sua real diretoria, uma vez que se trata de um evento *core*:

```
\mod_videotime\event\course_completion_updated
```

**Ficheiro “course\_completion\_updated.php”**  
(mod/videotime/classes/event/course\_completion\_updated.php)

```
1 <?php
2 namespace mod_videotime\event;
3
4 defined('MOODLEINTERNAL') || die();
5
6 class course_completion_updated extends \core\event\course_completion_updated {
7 ...
8 }
```

Esse evento foi então adicionado ao *plugin* “Level up! - Gamification”.

**Ficheiro “event\_lister.php” (blocks/xp/classes/local/rule/event\_lister.php)**

```
1 protected function get_core_events() {
2     global $CFG;
3     $eventclasses = [get_string('course') => [
4         '\\core\\event\\course_viewed '
5     ],
6     'Course Completion Updated' => ['mod\\videotime\\event\\
course_completion_updated '
7     ],
8     'Login' => ['\\core\\event\\user_loggedin '
9     ];
10    return $eventclasses;
11 }
```

### 3º Caso - Aluno Realiza uma Tarefa Diária

As tarefas diárias identificadas passam pelas submissões realizadas no “Diário de Humor” e “Diário de Atividades”, dois dos três *plugins* criados de raíz.

Sendo que estes são *plugins* a ser adicionados ao sistema, para além de não serem *plugins* do tipo Atividade, teve de se adicionar dois eventos novos (um em cada *plugin*) que são acionados, em ambos os casos, quando uma submissão é efetuada.

Para tal, teve de se adicionar em ambas as pastas dos *plugins* uma pasta “classes”, contendo outra pasta “event” onde os eventos seriam definidos, tal que:

- Tivessem o nível educacional (edulevel) “Participating”.
- Não fossem eventos anónimos (algo que na interação com ambos os *plugins* nunca aconteceria).

Os novos eventos são então:

```
\block_daily_mood\classes\event\humor_set
```

### Ficheiro “humor\_set.php” (blocks/daily\_mood/classes/event/humor\_set.php)

```

1 <?php
2 namespace block_daily_mood\event;
3 defined('MOODLEINTERNAL') || die();
4
5 class humor_set extends \core\event\base {
6     public function init() {
7         $this->data['crud'] = 'u';
8         $this->data['edulevel'] = self::LEVELPARTICIPATING; //Define Edulevel
9         $this->data['objecttable'] = 'block_daily_mood';
10    }
11
12    public static function get_name() {
13        return get_string('eventhumor_set', 'block_daily_mood');
14    }
15
16    public function get_url() {
17        return null;
18    }
19
20    public function get_description() {
21        return "Humor set by student with id '$this->userid'";
22    }
23 }
```

```
\block_daily_activities\classes\event\activity_set
```

### Ficheiro “activity\_set.php”

(blocks/daily\_activities/classes/event/activity\_set.php)

```

1 <?php
2 namespace block_daily_activities\event;
3 defined('MOODLEINTERNAL') || die();
4
5 class activity_set extends \core\event\base {
6
7     public function init() {
8         $this->data['crud'] = 'u';
9         $this->data['edulevel'] = self::LEVELPARTICIPATING; //Define Edulevel
10        $this->data['objecttable'] = 'block_daily_activities';
11    }
12
13    public static function get_name() {
14        return get_string('eventactivity_set', 'block_daily_activities');
15    }
16
17    public function get_url() {
18        return null;
19    }
20    public function get_description() {
21        return "Activities set by student with id '$this->userid'";
22    }
23
24 }
```

O *trigger* de ambos os eventos foi adicionado nos ficheiros blocks/daily\_mood/block\_daily\_mood.php e blocks/daily\_activities/block\_daily\_activities.php para ser efetuado após a submissão dos dados.

**Ficheiro “block\_daily\_mood.php” (blocks/daily\_mood/block\_daily\_mood.php)**

```

1 else {
2     $mform = new simplehtml_form();
3     //Form processing and displaying is done here
4     if ($fromform = $mform->get_data()) {
5         ...
6         $this->save_humor($mform->get_data()->humor);
7
8         $mform->set_data($mform->get_data());
9
10        //TRIGGER DAILY MOOD EVENT
11        $context = context_system::instance(); //Define Context (System)
12        $event = \block_daily_mood\event\humor_set::create(array('context' =>
13        $context));
14        $event->trigger();
15        ...

```

**Ficheiro “block\_daily\_activities.php”  
(blocks/daily\_activities/block\_daily\_activities.php)**

```

1 ...
2 else {
3     $mform = new table_structure();
4     //Form processing and displaying is done here
5     //Form Processing and Displaying
6     if ($fromform = $mform->get_data()) {
7
8         $this->save_activities($mform->get_data());
9
10        // Clean invalid checkboxes
11        $mform = new table_structure();
12        $mform->set_data($mform->get_data());
13
14        //TRIGGER DAILY ACTIVITIES
15        $context = context_system::instance(); //Define Context (System)
16        $event = \block_daily_activities\event\activity_set::create(array('
17        context' => $context));
18        $event->trigger();
19
20        $this->content->text = $mform->render();
21        ...

```

Posteriormente, adicionou-se ambos os eventos à função respetiva do *plugin* “Level up!”.

**Ficheiro “event\_lister.php” (blocks/xp/classes/local/rule/event\_lister.php)**

```

1 protected function get_core_events(){
2     global $CFG;
3     $eventclasses = [get_string('course') => [
4         '\\core\\event\\course_viewed' ]
5     ],
6     'Humor' => ['block_daily_mood\\event\\humor_set'
7     ],
8     'Daily Activities' => ['block_daily_activities\\event\\activity_set'
9     ],
10    'Video Completion Updated' => ['mod\\videotime\\event\\
11    course_completion_updated'
12    ],
13    'Login' => ['\\core\\event\\user_loggedin' ]
14 ];
15 return $eventclasses;

```

Assim, de acordo com os eventos adicionados e selecionados, as opções de regras escolhidas, representantes desses eventos, podem ser consultadas nas figuras 12 e 13.

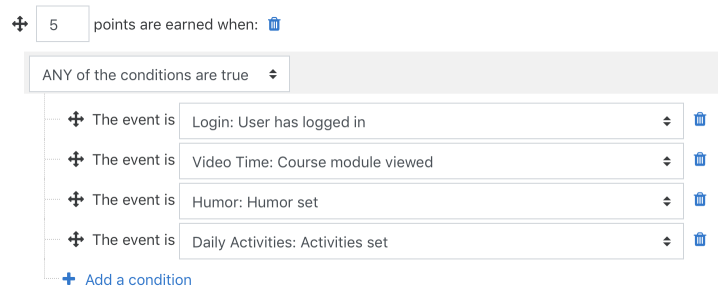


Figura 12: *Plugin “Level up! - Gamification”* (block\_xp) - primeira opção de regras para o cumprimento do REQ\_F\_18.2

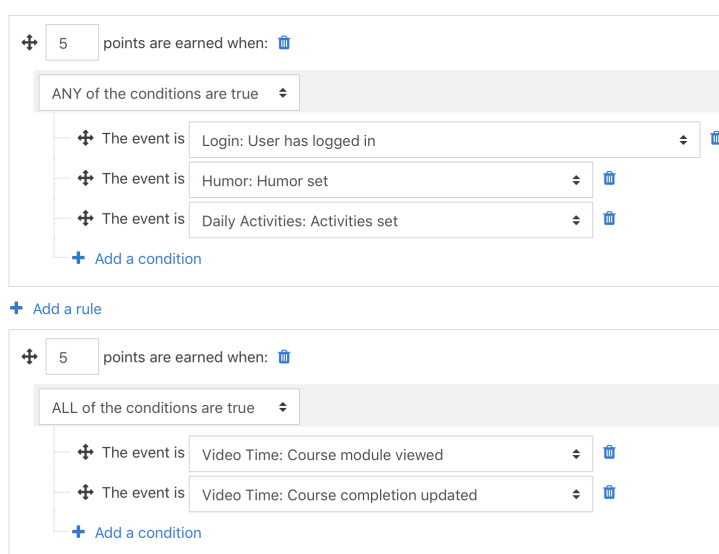


Figura 13: *Plugin “Level up! - Gamification”* (block\_xp) - segunda opção de regras para o cumprimento do REQ\_F\_18.2 - segunda opção

### 6.3.2.3 Requisito “REQ\_F\_18.3 - Atribuição de dez pontos”

#### 1º Caso - Aluno Preenche um Inquérito

O presente requisito poderia ser mapeado de forma direta através dos eventos já oferecidos pelo “Level up!”, nomeadamente eventos pertencentes às atividades “Inquérito” ou “Inquérito Predefinido”. Assim, o requisito seria cumprido, com a atribuição de dez pontos, se qualquer um dos seguintes eventos se verificasse:

- Inquérito (evento: “Resposta submetida”):

```
\mod_feedback\event\response_submitted
```

- Inquérito Predefinido (evento: “Resposta submetida”)

`\mod_survey\event\response_submitted`

Contudo, de modo a que os pontos sejam atribuídos, nenhum dos Inquéritos pode ser submetido de forma anónima.

## 2º Caso - Aluno Vê um Vídeo Com Pergunta

Anteriormente, falou-se de como poderiam ser integrados vídeos no Moodle. Não obstante, o seguinte requisito refere-se a “vídeos com pergunta”. Assim, um novo mecanismo teve de ser encontrado.

Aquando da exploração inicial das funcionalidades do Moodle, identificou-se um tipo de conteúdo que poderia ser adicionado aos editores de texto do mesmo: *HTML 5 Package - Pacote HTML 5 (H5P)*.

O H5P<sup>16</sup> é uma estrutura de colaboração de conteúdo gratuita e de código aberto baseada em *JavaScript* que visa facilitar a criação, partilha e reutilização de conteúdo em HTML5. Assim, imensos conteúdos interativos podem ser facilmente criados num editor baseado na web, sendo um deles, vídeos interativos (vídeos com perguntas embutidas).

Para além da possibilidade de criar conteúdo H5P e de o colocar posteriormente no Moodle através do editor de texto, encontrou-se o *plugin* “*Interactive Content – H5P*” (`mod_hvp`) [65] que permite inserir conteúdo H5P como um novo módulo de atividades (podendo, deste modo, os seus eventos ser detetados pelo “*Level up!*”).

Assim, o evento selecionado para se adicionar a regra no *plugin* “*Level up!*” com vista ao cumprimento deste requisito foi “Módulo de curso visualizado”:

`\mod_hvp\event\course_module_viewed`

## Concluir Vídeo com Pergunta para Atribuir Pontos

Relativamente ao tema abordado no 2º Caso do requisito REQ\_F\_18.2, que refere se a monitorização do tempo de visualização de um vídeo dever ser, ou não ser tida em consideração para o cumprimento do requisito, no caso de se usar H5P na criação de um vídeo interativo, essa questão seria facilmente contornada recorrendo aos seguintes passos:

1º - Inserir as perguntas em qualquer momento do vídeo e atribuir notas às perguntas realizadas, independentemente da resposta (estas notas ficariam disponíveis no “Livro de Notas” (“*Gradebook*”) de cada curso onde esse módulo de atividades estaria inserido);

2º - Exigir que os alunos não possam avançar no vídeo caso as perguntas anteriores não tenham sido respondidas (secção “*behavior*”);

3º - Utilizando a ferramenta do Moodle que permite rastrear a conclusão de uma atividade (secção 5.1) em conjunto com relatórios fornecidos pelo Moodle que permitem ao administrador e professor terem conhecimento da pontuação obtida por cada aluno, o professor ao ter acesso a essas notas preencheria na plataforma a atividade como “completa” aos alunos que tinham tido a nota máxima (pois significava que tinham respondido a todas as perguntas e, portanto, tinham visto o vídeo até ao fim).

4º - Colocar a atividade dos alunos que tiveram a nota máxima como completa, ou, dada essa condição, a tarefa ser *checkada* como completa de forma

---

<sup>16</sup>H5P

automática (o que permite ativar, de uma forma ou de outra, o evento `\mod_hvp\event\course_module_completion_updated`).

Esse evento foi então criado e adicionado ao *plugin* em questão, para possibilitar essa monitorização:

```
\mod_hvp\event\course_completion_updated
```

### 3º Caso - Aluno Realiza um *Quiz*

Uma vez que:

- se utilizou o Módulo de Atividades e Recursos “*Quiz*” no requisito REQ\_F\_18.1;
- ambos os requisitos, embora com Módulos de Atividades e Recursos iguais, estão relacionados a pontuações diferentes;
- o cumprimento do presente requisito atribui ao aluno a mesma pontuação atribuída no requisito relacionado com o vídeo interativo, ou seja, com a utilização de H5P,

para este requisito utilizou-se também os eventos fornecidos pelo Módulo de Atividades e Recursos H5P através da criação de *quizzes* H5P.

O evento selecionado foi, então “Tentativa de resposta submetida”:

```
\mod_hvp\event\attempt_submitted
```

### 4º Caso - Aluno Preenche um Registo Semanal

Segundo o cliente, os registos semanais passam por colocar à disposição dos alunos algumas fichas/trabalhos para os alunos preencherem e submeterem.

Dadas as características desta atividade, considerou-se que aquela que seria mais adequada para representar este requisito seria o módulo de atividade “Trabalho”.

O evento selecionado foi então “Enviada submissão de um trabalho”:

```
\mod_assign\event\assessable_submitted
```

De acordo com os eventos adicionados e selecionados, as opções de regras selecionadas, representantes desses eventos, podem ser consultadas na figura 14.

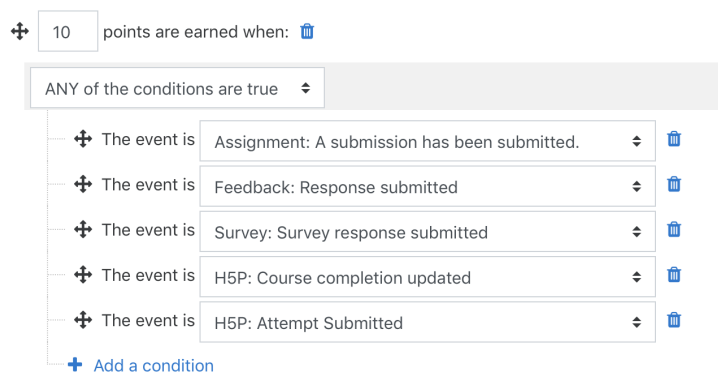


Figura 14: *Plugin “Level up! - Gamification”* (block\_xp) - opção de regras para o cumprimento do REQ\_F\_18.3

### 6.3.2.4 Requisito “REQ\_F\_18.4 - Atribuição de vinte pontos”

#### 1º Caso - Aluno Ouve um Ficheiro de Áudio

No presente requisito surgiram as mesmas questões levantadas no 2º Caso do requisito REQ\_F\_18.2, ou seja, encontrar formas de adicionar ficheiros de áudio, especificamente como Módulo de Atividades e Recursos (uma vez que é possível fazê-lo em qualquer Módulo de Atividades e Recursos que o Moodle fornece). Todavia, não se encontrou nenhum *plugin* funcional para tal por as soluções existentes serem extremamente antigas.

Dadas as suas características os dois Módulos de Atividades e Recursos do Moodle que seriam mais adequados para inserir ficheiros de áudio, possibilitando então a inserção de ficheiros .mp3, .aac, .flac, .m4a, .oga, .ogg e .wav, seriam:

- Páginas
- Ficheiros

Dado que a experiência de utilizador para um professor ao utilizar um “Ficheiro” para fazer *upload* de um áudio é mais intuitiva para esse Módulo de Atividades e Recursos em comparação com o Módulo de Atividades e Recursos “Página”, para além do *feedback* dado nos fóruns de discussão da comunidade Moodle, serão neste caso utilizados os eventos relativos ao Módulo de Atividades e Recursos “Ficheiro”.

Embora este Módulo de Atividades e Recursos também possibilite a inserção de outros tipos de ficheiros sem ser áudio, isto não constituiu um problema uma vez que os conteúdos e programa do Projeto SMS já se encontram todos criados e nenhum necessitou ser um “Ficheiro” obrigatoriamente. Este módulo ficaria reservado então para ficheiros de áudio.

Assim, seleccionou-se o evento “Módulo de disciplina visualizado”:

```
\mod_resource\event\course_module_viewed
```

#### Concluir Áudio para Atribuir Pontos

Novamente, neste caso também poderá ser ativado o rastreamento de conclusão de atividades. Assim, a mesma estratégia foi utilizada (criar um evento `course_completion_updated` no *plugin* e adicioná-lo ao “*Level up!*”).

```
\mod_resource\event\course_completion_updated
```

De acordo com os eventos adicionados e seleccionados, as opções de regras seleccionadas, representantes desses eventos, podem ser consultadas nas figuras 15 e 16.

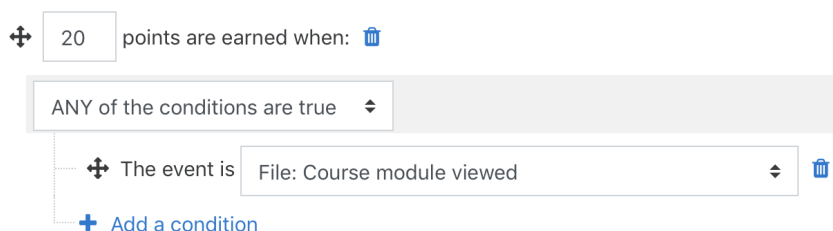


Figura 15: *Plugin “Level up! - Gamification”* (block\_xp) - primeira opção de regra para o cumprimento do REQ\_F\_18.4

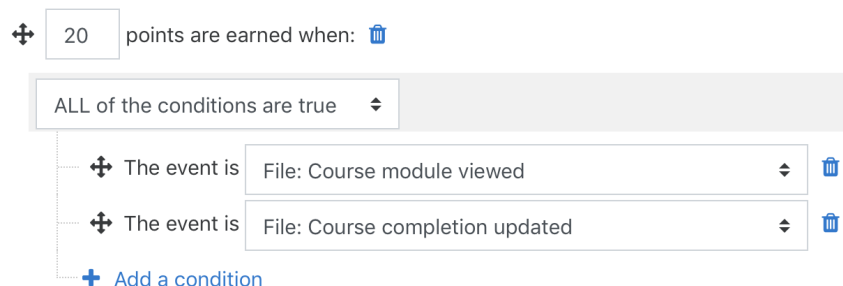


Figura 16: *Plugin “Level up! - Gamification” (block\_xp)* - segunda opção de regras para o cumprimento do REQ\_F\_18.4

## 2º Caso - Aluno Participou na Sessão Presencial

Neste caso, a condição para a atribuição de vinte pontos aos alunos passaria pelo facto de o aluno ter participado na sessão presencial.

O método que se escolheu para dar a possibilidade ao professor de assinalar as presenças foi o *plugin* “Attendance” [66] do tipo Módulo de Atividades e Recursos. Este permite que os professores mantenham um registo de presença na plataforma como substituto ou complemento de um registo de presença em papel, onde se podem definir todas as sessões, os seus horários (dia e hora), a frequência das sessões (número de dias por semana e duração) ou criar apenas sessões esporádicas (configuração realizada no próprio site).

Para poder atualizar as presenças, o professor tem então a lista das sessões, recebendo em cada uma a lista dos respetivos alunos inscritos no curso. As opções padrão para atualização das presenças fornecidas são: Presente (P); Ausente (A); Atrasado (L); Dispensado (E).

Embora as sessões também possam ser configuradas para permitir aos alunos registar a sua própria presença, essa situação nunca será aplicada.

Houve necessidade de arranjar um método que possibilitasse ao professor criar na plataforma um registo das datas das sessões realizadas e, no momento da sessão, ou posteriormente, assinalar a presença ou ausência de um aluno. Caso o aluno tivesse comparecido, um evento seria acionado para que a esse mesmo aluno fossem atribuídos vinte pontos pela participação.

Os eventos no Moodle estão predefinidos para serem atribuídos ao próprio utilizador que realiza uma ação e não a um utilizador afetado pela realização de uma ação de outro utilizador. Assim, após se explorar o funcionamento do *plugin* e o seu código, percebeu-se que as atualizações de presença que o *plugin* possibilitava, representavam os “*statusid*”. Estes encontravam-se associados a uma determinada classificação (*grade*) atribuída nas notas no “*Gradebook*” (figura 17).

Assumindo que essas classificações são igualmente os pontos (“xp” da tabela da base de dados “*block\_xp*”) que serão atribuídos aos alunos, encontrou-se uma solução bastante flexível (figura 18).

Assim, o professor/psicólogo assinala uma presença e é verificado se o “*statusid*” é um id válido (P ou L). Isto é realizado no ficheiro “*structure.php*” (mod/*attendance/classes/structure.php*) ao qual se adicionou novo código.

Em primeiro lugar, na função *get\_statuses* verifica-se o tipo de *status*. Assumindo que as letras “P”, “A”, “L” e “E” não são alteradas, vai-se então à base de dados buscar o “*statusid*”.



## Attendance for the course :: New Site



Sessions Add session Report Export Status set Temporary users

Changes to status sets will affect existing attendance sessions and may affect grading.

Status set 1 (P L E A) ▾

#	Acronym	Description	Points	Available for students (minutes)	Automatically set when not marked	Action
1	P	Present	2.00	<input type="text"/>	<input type="radio"/>	
2	L	Late	1.00	<input type="text"/>	<input type="radio"/>	
3	E	Excused	1.00	<input type="text"/>	<input type="radio"/>	
4	A	Absent	0.00	<input type="text"/>	<input type="radio"/>	
*	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>	

Add

Update

Figura 17: *Plugin “Attendance”* (mod\_attendance) - predefinição da atribuição de “grades”

## Attendance for the course :: New Site



Sessions Add session Report Export Status set Temporary users

Changes to status sets will affect existing attendance sessions and may affect grading.

Status set 1 (P L A E) ▾

#	Acronym	Description	Points	Available for students (minutes)	Automatically set when not marked	Action
1	P	Present	20	<input type="text"/>	<input type="radio"/>	
2	L	Late	20	<input type="text"/>	<input type="radio"/>	
3	E	Excused	0	<input type="text"/>	<input type="radio"/>	
4	A	Absent	0	<input type="text"/>	<input type="radio"/>	
*	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>	

Add

Update

Figura 18: *Plugin “Attendance”* (mod\_attendance) - alteração da atribuição de “grades” para adicionar pontuação ao *plugin “Level up! - Gamification”* (block\_xp)

### Ficheiro “structure.php” (mod/attendance/classes/structure.php)

```

1  /* Get the grade for a specific $statusid*/
2  public function get_status_grade($statusid){
3      global $DB;
4      $filter = array('id' => $statusid);
5      $status = $DB->get_record('attendance_statuses', $filter);
6      if($status) {
7          return $status->grade;
8      }
9      return 0;
10 }
11
12
13 /*Add points to the user with id $studentid*/
14 public function update_xp($studentid, $points){
15     global $DB;
16     $filter = array(
17         'userid' => $studentid
18     );
19     $xp = $DB->get_record('block_xp', $filter);
20     if($xp) {
21         $xp->xp += $points;
22         $DB->update_record('block_xp', $xp);
23     }
24     else {
25         // student didn't score yeat, first time
26         $xp = array(
27             'courseid' => 1,
28             'userid' => $studentid,
29             'xp' => $points,
30             'lvl' => 1,
31         );
32         $DB->insert_record('block_xp', $xp);
33     }
34 }

```

O “statusid” é atualizado se for diferente (se o professor alterar o *status* da presença). Assim, é necessário ir buscar a *old grade* e a *new grade*: ao *old grade* temos de subtrair e ao *new grade* temos de somar.

```

1  public function take_from_form_data($formdata) {
2      ...
3          // CODE CHANGE
4          if($dbsesslog[$log->studentid]->statusid <> $log->statusid) {
5              $old_grade = $this->get_status_grade($dbsesslog[$log->studentid]->
6              statusid);
7              $new_grade = $this->get_status_grade($log->statusid);
8              $this->update_xp($log->studentid, $new_grade - $old_grade);
9          }
10     } else {
11         $DB->insert_record('attendance_log', $log, false);
12         // CODE CHANGE
13         $new_grade = $this->get_status_grade($log->statusid);
14         $this->update_xp($log->studentid, $new_grade);
15     }
16     ...
17 }

```

### 6.3.3 Requisitos “REQ\_F\_19”

No que diz respeito ao requisito REQ\_F\_19 (REQ\_F\_19.1 - Receber “crachá pequeno”; REQ\_F\_19.2 - Receber “crachá grande”), apesar de o próprio Moodle permitir incluir e atribuir “*Moodle Badges*” [54][55], este será também cumprido através de configurações realizadas no *plugin* “*Level up!*” (Estratégia 2) pelas seguintes razões:

1. Ser um requisito que se encontra diretamente relacionado com a pontuação atribuída pelo *plugin* “*Level up!*”;
2. Saber-se que o “*Level up!*” possibilita a personalização do número de níveis, dos pontos necessários para subir de níveis e a sua aparência.

Da forma como o “*Level up!*” está configurado atualmente, a exibição dos níveis é representada por uma imagem e a subida de nível é representada pela alteração dessa imagem, para outra. Assim:

- As imagens correspondentes aos níveis do “*Level up!*” irão ser substituídas pelos crachás criadas para o projeto;
- O padrão de pontos predefinido para subir de nível irá ser substituído por um padrão de 100 em 100 pontos. As imagens atribuídas a cada 100 pontos correspondem aos crachás criados para o projeto, intercalando entre “crachás pequenos” (REQ\_F\_19.1) e “crachás grandes” (REQ\_F\_19.2).

3. Manter tudo no mesmo “sistema” e assim ser menos confuso e dispersar menos o aluno.

O *plugin* foi então modificado, podendo a sua apresentação final ser consultada em anexo (ANEXO C).

### 6.3.4 Requisito “REQ\_MP\_02”

No caso dos Módulos de Atividades e Recursos, esses *plugins*, como referido na secção 4.3, são mapeados de forma direta para a aplicação móvel. No entanto, uma vez que os restantes *plugins* implementados no presente projeto, passam por *plugins* do tipo Bloco, que exigem que o seu suporte pela aplicação móvel seja adicionado, a estratégia utilizada para o cumprimento do presente requisito foi a Estratégia 1.

De maneira a se poder desenvolver as versões para a aplicação móvel dos *plugins* criados de “raíz” (nomeadamente, requisitos REQ\_F\_10, REQ\_F\_10.1 e REQ\_F\_10.2) e dos adaptados (REQ\_F\_18, REQ\_F\_18.2, REQ\_F\_18.3, REQ\_F\_18.4, REQ\_F\_19, REQ\_F\_19.1, REQ\_F\_19.2) utilizou-se a documentação do Moodle relativo ao suporte de *plugins* pela aplicação móvel [48].

Uma vez que é através dos *templates* criados que os *plugins* são exibidos na aplicação móvel, os tipos de *templates* gerados (que, no fundo, retratam os tipos de *plugins*) serão selecionados. Assim, os tipos de *templates* utilizados e gerados, tendo em consideração os *plugins* implementados, passarão por *templates* gerados e descarregados quando o utilizador abre o *plugin* (figura 19). Significa isto que a função receberá parâmetros relativos ao contexto do *plugin*.

Para tal, é necessário selecionar o *Delegate*<sup>17</sup> correto, classificado pelo tipo de *plugin*. Para o desenvolvimento das versões móvel dos *plugins* já desenvolvidos, utilizou-se o *delegate* “*Core-BlockDelegate*” de maneira a dar suporte aos blocos.

Os *templates* adotados pelo Moodle passam por *templates mustache*, chamados de *templates “logic-less”* por apenas utilizarem *tags* (quando existentes) e não incluírem intruções “if”, cláusulas “else” ou ciclos. São desenvolvidos em *Ionic* através da *Framework AngularJS* do

<sup>17</sup>Delegates Existentes

Google, considerada uma das melhores e mais populares *frameworks* existentes. A aplicação é incorporada no *container* nativo utilizando *Cordova*.

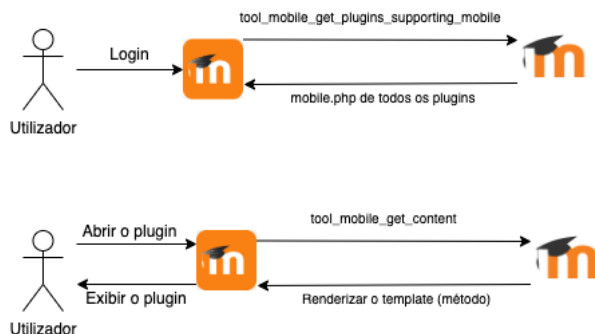


Figura 19: Templates para o tipo de *plugins* selecionado (baseado na figura existente na documentação referente ao suporte de *plugins* pela aplicação móvel do Moodle [48])

No contexto do presente projeto, as versões móvel dos *plugins* poderão ser implementadas da forma que o *developer* entender, desde que, não comprometa a natureza do requisito em questão.

#### 6.3.4.1 Requisito “REQ\_F\_10.1” (Versão Móvel)

O primeiro *plugin* onde se criou a versão móvel foi o *plugin* “daily\_mood”. Os ficheiros e pastas predefinidos (secção 4.3) foram criados, assim como um ficheiro “common.php” que continha as funções comuns tanto para o *plugin web* (anteriormente existentes no ficheiro “block\_daily\_mood.php” da pasta principal), como para móvel.

Web e *mobile* funcionam em conformidade, na medida em que, se o valor do humor for submetido em web, não é possível em *mobile* submetê-lo novamente (no mesmo dia), assim como se o valor do humor for submetido em *mobile*, não é possível voltar a submeter (no mesmo dia) em web.

O *web service* para a possibilidade de submissão do botão para guardar o valor do *range slider* também foi incluído seguindo todas as predefinições.

No caso do presente *plugin* dois *templates* foram criados: um para a exibição do *range slider* inicial e botão de submissão, e outro, exibido após a submissão com a informação de que o humor teria sido submetido e qual o valor submetido.

##### Ficheiro “mobile.php” (blocks/daily\_mood/classes/event/mobile.php)

Ficheiro onde se encontram declaradas e implementadas as funções a utilizar.

##### Ficheiro “mobile.php” (blocks/daily\_mood/db/mobile.php)

Tal como descrito na secção 4.3, o presente ficheiro será construído da seguinte forma:

```

1 <?php
2 defined( 'MOODLE_INTERNAL' ) || die();
3
4 $addons = [
5     'block_daily_mood' => [ // Plugin identifier
6         'handlers' => [ // Different places where the plugin will display
           content.
7             'view' => [ // Handler unique name (alphanumeric).
8                 'displaydata' => [

```

```

9             'title' => 'pluginname',
10            'class' => 'humor_journal',
11            'type' => ''
12        ],
13
14            'delegate' => 'CoreBlockDelegate', // Delegate (where to
display the link to the plugin)
15            'method' => 'humor_initial', // Main function in \daily_mood\
output\mobile
16        ],
17    ],
18    'lang' => [ // Language strings that are used in all the handlers.
19        ['pluginname', 'block_daily_mood'],
20        ['button_save_humor', 'block_daily_mood'],
21    ],
22 ],
23 ];

```

### Ficheiro “services.php” (blocks/daily\_mood/db/services.php)

Ficheiro para adicionar o próprio *web service*:

```

1 <?php
2 defined('MOODLEINTERNAL') || die();
3
4 //Functions array
5 $functions = array(
6     'block_daily_mood_set_humor' => array(
7         'classname' => 'block_daily_mood_external', //class that contains
the implementation of the functions
8         'classpath' => 'blocks/daily_mood/externallib.php', //path of
the class
9         'methodname' => 'set_humor', //name of the function inside the
class
10        'description' => 'Change Humor',
11        'type' => 'write',
12        'ajax' => true,
13        'services' => [MOODLE_OFFICIAL_MOBILE_SERVICE, 'local_mobile'] //
enable the mobile access
14    ),
15 );
16
17 //Declaration of services
18 $services = array(
19     'Diario Humor' => array( //the name of the web service
20         'functions' => array (
21             'block_daily_mood_set_humor'), //web service functions of
this service
22         'restrictedusers' => 0, // if enabled, the Moodle administrator
must link some user to this service
23         // into the administration
24         'enabled' => 1, // if enabled, the service can be reachable on a
default installation
25         'shortname' => '', // optional - but needed if restrictedusers
is set so as to allow logins.
26     )
27 );

```

**Ficheiro “externallib.php” (blocks/daily\_mood/externallib.php)**

Este ficheiro representa o local para a implementação das funções para o *web service*. O nome das funções é arbitrário, contudo, deve seguir uma certa convenção. A descrição de parâmetros e a descrição de valores retornados são obtidos através da mesma classe, por chamadas estáticas a métodos, ao se adicionar “\_parameters” e “\_returns” ao valor do nome do método.

Para além disso, no presente ficheiro, também se efetua novamente o *trigger* do evento criado para o respetivo *plugin*, para possibilitar a atribuição de pontos ao se submeter o humor na aplicação móvel.

```

1 <?php
2 defined('MOODLE_INTERNAL') || die();
3
4 require_once($CFG->libdir . "/externallib.php");
5 require_once(dirname(__FILE__) . "/common.php");
6
7 class block_daily_mood_external extends external_api {
8
9     public static function set_humor_parameters() {
10         return new external_function_parameters(
11             array(
12                 'humor' => new external_value(PARAM_INT, 'Daily Humor Value',
13 VALUE_REQUIRED, '', NULL_NOT_ALLOWED),
14             ));
15     }
16
17     public static function set_humor(int $humor) {
18         global $USER;
19         global $DB;
20
21         $params = self::validate_parameters(self::set_humor_parameters(),
22             array(
23                 'humor' => $humor
24             ));
25         $humor = $params['humor'];
26
27         $result = save_humor($humor);
28
29         ...
30
31         $result = array(
32             "humor" => $humor
33         );
34         return $result;
35     }
36
37     public static function set_humor_returns() {
38         return new external_single_structure(array(
39             'humor' => new external_value(PARAM_INT, 'Humor Set'),
40         ));
41     }
42 }

```

**Ficheiro “mobile\_daily\_mood.mustache”  
(blocks/daily\_mood/templates/mobile\_daily\_mood.mustache)**

Ficheiro para criar a apresentação do *plugin* (um *range slider* de zero a cem e um botão para submeter o valor, botão esse que chama a função do *web service* no servidor).

```

1     ...
2     <ion-range min="0" max="100" value="50" pin="true" ticks="false" [(
ngModel)]="humor" name="ddd" color="secondary">
3         <ion-label range-left color="secondary">0</ion-label>
4         <ion-label range-right color="secondary">100</ion-label>
5         <ion-icon size="small" slot="start" name="thermometer" item-start></
ion-icon>
6         <ion-icon slot="end" name="thermometer"></ion-icon>
7     </ion-range>
8
9     <button color="tertiary" ion-button core-site-plugins-call-ws-new-content
10        name="block_daily_mood_set_humor"
11        form="humor_form"
12        [params]="{humor: humor}"
13        [preSets]="{getFromCache: 0, saveToCache: 0}"
14        samePage="true"
15        component="block_daily_mood" method="humor_result" >
16     <ion-icon name="thermometer" item-start></ion-icon>&nbsp;  
17         {{ 'plugin.block_daily_mood.button_save_humor' | translate }}
18     </button>
19     ...

```

### Ficheiro “daily\_mood\_set.mustache” (blocks/daily\_mood/templates/daily\_mood\_set.mustache)

Ficheiro para criar a apresentação do *plugin* após a submissão do formulário.

```

1  {{=<% %>=}}
2  <ion-card>
3  <div class="hero-unit">
4    <h1>Daily Mood</h1>
5
6    <ion-icon size="small" name="thermometer" item-start></ion-icon>
7    <ion-text color="secondary">
8    <p>Humor already defined: <% humor %></p>
9    </ion-text>
10
11 </div>
12 </ion-card>

```

Assim, o requisito REQ\_F.10.1 é assegurado na aplicação móvel, podendo a sua estrutura final ser consultada no ANEXO C.

#### 6.3.4.2 Requisito “REQ\_F\_10.2” (Versão Móvel)

De seguida, implementou-se e versão móvel do *plugin* “Diário de Atividades” (REQ\_F.10.2) que, só diferia do “Diário do Humor” na forma como se carregam e exibem os dados e como estes são enviados. A particularidade está no facto de se enviar um *array* dinâmico (uma vez que os dados e o seu número de parâmetros variam, consoante o número de atividades que um aluno preenche). Assim utilizam-se “*Dynamic Names*”<sup>18</sup> de maneira a que o *web service* possa aceitar dados dinâmicos.

Isso foi efetuado no template criado para o *plugin*, no ficheiro blocks/daily\_activities/templates/daily\_activities.mustache, através da diretiva *Angular* “ngModel” que permite armazenar o valor de uma determinada entrada ou selecionar uma variável *javascript*, ou informar/selecionar o valor a definir.

<sup>18</sup> Aceitar Nomes Dinâmicos num Web Service

## Ficheiro “daily\_activities.mustache” (blocks/daily\_activities/templates/daily\_activities.mustache)

```

1   ...
2   <%#activities%>
3   <ion-item button="false" lines="full">
4     <ion-input color="light" value="<% name %>" placeholder="Escreva Nova
Atividade" [(ngModel)]="CONTENT.OTHERDATA.data['activity<% number %>']"></ion-
input>
5     <ion-checkbox color="light" checked="<% checked %>" name="day<% number
%>" [(ngModel)]="CONTENT.OTHERDATA.data['day<% number %>']"></ion-checkbox>
6   </ion-item>
7   <%/activities%>
8   ...
9

```

Uma vez que se poderia criar a versão *mobile* do *plugin* da forma que se desejava (e visto que se utilizam *web services* na submissão do formulário), estruturou-se a sua exibição da seguinte forma:

- A versão móvel do *plugin* é constituída por *inputs* (para introduzir as atividades diárias) e respetivas *checkboxes* para marcar as atividades realizadas.
- As atividades que aparecem, ao se abrir a aplicação, são do dia corrente sendo que existem dois botões, “previous” e “next”, para “andar” para os dias anteriores ou para os dias seguintes, respetivamente (é apenas dado um *timestamp*, o do dia atual, e a partir daí, avança-se, ou recua-se nos dias).

```

1   <button color="tertiary" ion-button core-site-plugins-new-content
2     [args]="{timestamp: <% day %>}"
3     [preSets]="{getFromCache: 0, saveToCache: 0}"
4     samePage="true"
5     component="block_daily_activities" method="activities_previous">
6     {{ 'plugin.block_daily_activities.button_previous' | translate }}
7   </button>
8
9   <button color="tertiary" ion-button core-site-plugins-new-content
10    [args]="{timestamp: <% day %>}"
11    [preSets]="{getFromCache: 0, saveToCache: 0}"
12    samePage="true"
13    component="block_daily_activities" method="activities_next">
14    {{ 'plugin.block_daily_activities.button_next' | translate }}
15  </button>
16

```

Assim, o requisito REQ\_F.10.2 é assegurado para a aplicação móvel (ver ANEXO C).

Após a criação destes novos botões na versão *mobile*, pensou-se em adicionar estas funcionalidades na versão web (avançando ou recuando nas semanas exibidas na tabela). Contudo, após inúmeras tentativas, nesse caso, não foi possível a adição dessas funcionalidades.

Isto porque os *web services* necessitam de autenticação. Algo que se fez, foi analisar o código do *Moodle*, para perceber como esta poderia ser efetuada, uma vez que a documentação não continha qualquer informação e não era clara. Percebeu-se que essa autenticação pode ser efetuada através de *username* e *password* ou através de um *token*.

Para se obter o *username* e *password*, estes dados tinham de ser pedidos no momento, ao utilizador, o que não faria sentido.

Para se obter o *token*, tinha de se gerar um *token* [73] para cada utilizador, sendo que sempre que um utilizador fosse adicionado, um novo *token* também teria de o ser. Uma vez



gerados e guardados, iam-se chamando esses *tokens*. No entanto, considerou-se que esta solução também não faria sentido.

Outra solução que também foi encontrada (outra maneira de chamar funções pela versão *web* do Moodle), foi através de um *service* (ficheiro lib/ajax/service.php):

#### Ficheiro “daily\_activities.js” (blocks/daily\_activities/js/daily\_activities.js)

```

1 function previousWeek() {
2     let token = getCookie("MoodleSession");
3     let sesskey = document.getElementsByName('sesskey')[0].value;
4     let current_week = document.getElementsByName('week_begin')[0].value;
5     let previous_week = current_week - 604800; // Minus 1 week
6
7     let params = {
8         weekbegin: previous_week,
9     }
10
11     fetch('/moodle/lib/ajax/service.php?sesskey=' + sesskey +
12         '&info=block-daily-activities-previous-week',
13         {
14             method: 'POST',
15             headers: {
16                 'Content-Type': 'application/json; charset=utf-8'
17             },
18             body: JSON.stringify(params)
19         })
20     .then(response => response.json())
21     .then(result => alert(result));
22 }
23

```

No entanto, não foi possível, também assim, obter o que se pretendia (para além de não ser uma solução que estivesse documentada).

#### 6.3.4.3 Requisito “REQ\_F\_18” (Versão Móvel)

Tratando-se de um *plugin* apenas para exibição de dados, não foi necessário utilizar um *web service* nem criar os ficheiros relacionados (db/services.php; externallib.php).

Os ficheiros mais relevantes para a criação do presente *plugin* na versão móvel foram então:

#### Ficheiro “mobile.php” (blocks/xp/db/mobile.php)

Semelhante ao criado para o *plugin* “daily\_mood”

```

1 <?php
2 defined('MOODLE_INTERNAL') || die();
3
4 $addons = [
5     'xp_journal' => [ // Plugin identifier
6         'handlers' => [ // Different places where the plugin will display
7             content.
8             'xp_journal' => [ // Handler unique name (alphanumeric).
9                 'displaydata' => [
10                     'title' => 'pluginname',
11                     'class' => 'xp-journal',
12                     'type' => ''
13                 ],
14                 'delegate' => 'CoreBlockDelegate', // Delegate (where to
15                 display the link to the plugin)
16                 'method' => 'mobile-xp-journal', // Main function in \xp\
17                 classes\output\mobile

```

```

15         ],
16     ],
17     'lang' => [ // Language strings that are used in all the handlers.
18         ['pluginname', 'block_xp'],
19     ],
20 ],
21 ];
22

```

### Ficheiro “mobile.php” (blocks/xp/classes/output/mobile.php)

Ficheiro com as funções utilizadas para aceder aos dados das tabelas do *plugin* “Level up!”, entre elas, a função principal “mobile\_xp\_journal”, assinalada no ficheiro blocks/xp/db/mobile.php (responsável também pela exibição do *plugin* através da renderização do *template* criado):

```

1     ...
2     public static function mobile_xp_journal($args) {
3         global $OUTPUT, $DB;
4
5         $data = array();
6         $data["xp"] = mobile::get_xp($args['userid']);
7         $data["lvl"] = mobile::get_lvl($args['userid']);
8         return [
9             'templates' => [
10                [
11                    'id' => 'main',
12                    'html' => $OUTPUT->render_from_template('
13                block_xp/xp', $data),
14                ],
15            ],
16            'javascript' => '',
17            'otherdata' => '',
18        ];
19     }
20     ...

```

### Ficheiro “xp.mustache” (blocks/xp/templates/xp.mustache)

```

1     ...
2     <ion-text color="secondary">
3     <h1>Level: <% lvl %></h1>
4     </ion-text>
5
6     <ion-text color="secondary">
7     <h1>Points: <% xp %></h1>
8     </ion-text>
9     ...

```

Uma vez que grande parte das exigências do requisito REQ\_F\_18 foram assegurados através de Módulo de Atividades e Recursos e de esses serem automaticamente mapeados para a aplicação móvel, os únicos requisitos que terão de ser também manipulados para que, aquando do seu uso na aplicação móvel, sejam cumpridos, são os dos novos *plugins* criados (referentes aos requisitos REQ\_F\_10.1 e REQ\_F\_10.2).

Assim, no ficheiro blocks/daily\_mood/externallib.php, também se efetuou *trigger* ao evento criado (\block\_daily\_mood\event\humor\_set), após a submissão do valor do humor.

O mesmo se efetuou no ficheiro blocks/daily\_activities/externallib.php (evento \block\_daily\_activities\event\activities\_set), após a submissão no “Diário de Atividades”.

Algo que não se conseguiu efetuar na aplicação móvel em tempo útil de desenvolvimento do projeto, foi a exibição do “Gráfico de Humor e Atividades” (REQ\_F\_10) e dos crachás relativos ao requisito REQ\_F\_19.

Para o primeiro caso, algo que se pensou foi utilizar a biblioteca “*ChartJS*” do Moodle para exibição do gráfico na aplicação móvel, sendo apenas necessário fornecer: o tipo de gráfico; um *array* com cada valor a exibir no gráfico; um *array* com as *labels* para cada valor.

No segundo, a estratégia seria exibir a imagem correspondente ao crachá atual no respetivo *plugin*. Contudo, não se conseguiu adicionar imagens à aplicação *Ionic* do Moodle.

No entanto, não se considerou que estas questões representassem um grande impacto no sucesso do projeto. A existência da aplicação móvel consiste, na verdade, num bónus adicionado ao projeto, e é, como referido, inerente a um requisito de prioridade relativamente baixa.

Porém, tenciona-se futuramente adicionar estes *plugins*/requisitos em falta.

### 6.3.5 Requisito “REQ\_F\_13”

O presente requisito surgiu, essencialmente, da vontade de o cliente enviar SMSs em alturas específicas e com conteúdo específico. Contudo, visto que é um requisito que acaba por ter sempre custos associados, e o cliente ainda precisar de elaborar o plano que deseja para o envio de SMSs, acabou por se decidir que os clientes da plataforma deveriam ter sempre uma opção que pudessem acionar, no futuro, caso decidissem tomar essa decisão e passar a poder enviar SMSs.

Novamente, a primeira abordagem selecionada, que acabou por ser a utilizada, foi a de pesquisar por soluções já existentes na diretoria de *plugins* do Moodle (Estratégia 2), na medida em que utilizasse a capacidade de centros de contacto remoto que serviriam como meio para o envio das mensagens.

Tabela 27: Identificação *Frankenstyle* - *plugin* “block\_sms”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	sms	block_sms	blocks/sms

Encontrou-se um *plugin*, também de tipo Bloco (tabela 27), designado “*Blocks: SMS Notifier*” [74]. Este permite que os administradores, gestores e professores da plataforma, enviem mensagens de texto aos restantes utilizadores (alunos e encarregados de educação). Para tal, as APIs utilizadas são as seguintes (podendo o utilizador deste *plugin*, neste caso, o *developer* e o cliente, escolher por qual irão optar):

- *Nexmo* [75]
- *Twilio* [76]
- *Clickatell* [77]

Por ser das três a mais reconhecida, essencialmente por ser utilizado por grandes empresas de todo o mundo para potenciar a sua comunicação (como é o caso da *Netflix* e *Airbnb*) e cobrar apenas pelo que usa, sem a necessidade de grandes contratos, a API REST do *Twilio* foi a selecionada.

Em suma, o *Twilio* é uma plataforma utilizada por *developers* para agilizar a comunicação através das APIs fornecidas (quer seja para adicionar recursos como voz, vídeo, ou envio de mensagens) em aplicações [78]. Por trás das APIs do *Twilio* há uma *Super Network*, que serve de camada de software para conectar e otimizar as redes de comunicação em todo o mundo.

De uma forma geral, a aplicação comunica com o *Twilio* utilizando o *Twiml* (um conjunto de instruções baseado em XML) e com as APIs REST.

O REST constitui um padrão arquitetural que descreve a forma como Sistemas Distribuídos podem expor uma interface consistente, sendo uma API acessada através do Protocolo HTTP num conjunto de URLs predefinidos (representando diversos recursos, ou seja, qualquer informação/conteúdo acessado nesse local, que pode ser retornado como JSON, HTML, ficheiros de áudio ou imagens, sendo que geralmente esses recursos têm um ou mais métodos que podem ser executados por HTTP, como GET, POST, PUT e DELETE).

O *Twilio*, por exemplo, fornece diversas APIs REST independentes para o envio de mensagens de texto, entre outras. No ecossistema do *Twilio*, cada produto é a sua própria API. Todos os produtos são utilizados da mesma maneira, seja por HTTP ou usando as bibliotecas auxiliares do *Twilio* para várias linguagens de programação diferentes.

### Envio de SMSs:

Para enviar SMSs (figura 20) com a API *Twilio* [79], a nossa aplicação terá de enviar um “API request” para enviar uma mensagem de saída. O nosso servidor web envia um pedido *POST* (mensagens) para o *Twilio* e o *Twilio* envia um SMS ao utilizador destino. Isto é realizado através do Cliente REST do *Twilio*, que é utilizado para criar uma mensagem a partir (*From*) de um Número de Telefone da conta *Twilio* para um número de telemóvel destino, com um determinado texto (presente no *Body text*).

O pedido *POST* para a API pode ser feito diretamente com cURL ou utilizando uma biblioteca auxiliar suportada para o envio de mensagens com PHP, Python, entre outras linguagens.



Figura 20: Envio de SMSs através do Twilio

### Receção de SMSs:

Para lidar com mensagens de entrada (embora esse caso não faça parte do projeto), isto funciona de forma reversa (figura 21). Quando uma mensagem de chegada é recebida, o *Twilio* realiza um pedido web (HTTP) ao servidor web à procura de instruções *Twiml* para saber o que efetuar.

O utilizador envia um SMS para o Número de Telefone *Twilio* escolhido tendo em consideração as capacidades do projeto (o *Twilio* conta com números de telefone disponíveis em mais de trinta países, permitindo escolher dentro deste inventário os números certos, com os recursos certos para um dado projeto).

De seguida, o *Twilio* realiza um pedido HTTP para o nosso servidor (aplicação web). Esse devolve algum *Twiml* e se esse indicar que deve enviar uma mensagem, então envia uma resposta de volta ao utilizador (“*Hello World*”).

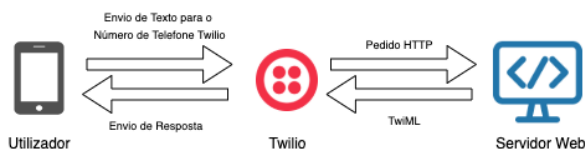


Figura 21: Receção de SMSs através do Twilio

Notar que tudo isto só é possível quando se tem uma conta criada de maneira a se poder ter acesso aos Números de Telefone disponibilizados pelo *Twilio* e aos ID's Alfanuméricos de Envio (*Account SID*), sendo que, os preços das mensagens de texto são cobradas por segmento e, no caso de Portugal, os preços praticados são os encontrados na tabela 28.

Tabela 28: Tabela de preços praticados em Portugal para envio e receção de mensagens através do *Twilio*

	Enviar	Receber
Números de Telefone	\$0.0450	\$0.0450
ID's Alfanuméricos de Envio	\$0.0450	N/A

Posto isto, o papel então do *plugin* selecionado (“*Blocks: SMS Notifier*”) é o de agilizar e promover a integração da API do *Twilio* (*sms/classes/twilio\_api.php*) para poder ser utilizado na plataforma Moodle por um utilizador que não seja *developer* (o cliente) e facilmente passar a conseguir enviar mensagens caso venha a sentir essa necessidade, como é o caso do cliente do presente projeto.

Para tal, os únicos passos que terão de ser seguidos, após a instalação do *plugin*, serão os apresentados nos pontos seguintes:

1. Colocar no formulário concedido o ID Alfanumérico de Envio (*Account SID*), o *Token Auth* e o Número de Telefone do *Twilio* (*From Number*) - figura 22;

Figura 22: Primeiro passo da configuração do *plugin*

2. Selecionar a opção para enviar SMSs - figura 23;

Figura 23: Segundo passo da configuração do *plugin*

## 3. Enviar SMS - figura 24.

Figura 24: Terceiro passo da configuração do *plugin*

Considera-se também pertinente referir que o facto de, no futuro, poderem existir custos associados ao uso deste mecanismo, não interfere com o requisito REQ\_BIZ\_01 referente à gratuitidade da plataforma. Qualquer custo nunca será imputado ao utilizador (uma vez que são da responsabilidade do cliente).

## 6.3.6 Requisito “REQ\_F\_15”

Face às características do presente requisito e o atual conhecimento do Moodle, decidiu-se utilizar o “Bloco HTML” [80] já fornecido pelo Moodle. Utilizou-se, portanto, a Estratégia 2.

Tabela 29: Identificação *Frankenstyle* - *plugin* “block.html”

Tipo do <i>Plugin</i>	Nome do <i>Plugin</i>	<i>Frankenstyle</i>	Pasta
<i>block</i>	html	block.html	blocks/html

O “Bloco HTML” (tabela 29) é um bloco padrão usado para adicionar texto e multimédia (por exemplo) à página inicial do site, podendo o seu título ser alterado para o que se deseja (no caso, em vez de “Bloco HTML”, modificou-se para “Botões de Partilha nas Redes Sociais” e, adicionou-se então, os botões desejados (figura 25).

O cumprimento do requisito seria então assegurado pela utilização, no caso do *Facebook*, de “*Social Plugins*”<sup>19</sup> disponibilizados pelo “*Facebook for Developers*” e, no caso do *Twitter*, pelo “*Twitter Developer*” (*Tweet button*<sup>20</sup>). No entanto, outra solução que se encontrou, que acabou por ser a utilizada, passou pela integração de botões de partilha “*Share This*”<sup>21</sup> que, para além de possibilitarem a integração de botões de partilha tanto para o *Facebook*, como para o *Twitter*, permitem também a partilha em muitas outras plataformas (*Pinterest*, *Mail*, *Whatsapp*, por exemplo), o que o cliente considerou benéfico.

<sup>19</sup> Botão de Partilha do Facebook

<sup>20</sup> Botão de Partilha do Twitter

<sup>21</sup> Botões de Partilha “Share This”

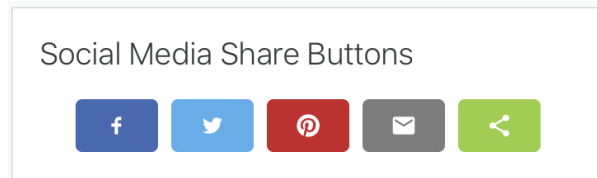


Figura 25: *Plugin* do tipo Bloco “HTML” após as configurações efetuadas - perfil de um professor/psicólogo

Algo que foi necessário realizar neste *plugin* foi definir que esse bloco só poderia ser acessado por um utilizador com o papel de “Professor”. Uma das formas de se averiguar essa questão, foi de descobrir se o utilizador atual possuía uma das principais *capabilities* associadas ao papel de “Professor”, como por exemplo:

```
1 if (!has_capability('moodle/course:manageactivities', $context)) {  
2     return;  
3 }
```

*Esta página foi intencionalmente deixada em branco.*



## 7 Testes

De acordo com a norma IEEE Std 1059-1993 [81], em Engenharia de Software, um teste define-se como:

**Um método de análise de um item de software para verificar diferenças entre condições atuais e condições desejadas. Envolve também a avaliação de funcionalidades do software.**

A fase de realização de testes de software representa então a investigação realizada com vista a fornecer aos *stakeholders* informações sobre a qualidade do produto ou serviço de software em questão. Tendo em consideração os requisitos do produto desejado, são realizados testes ao mesmo, para verificar se o produto suporta os requisitos pretendidos.

Algumas vantagens da realização de testes são:

- **Redução de Erros**

Os testes são realizados para que se possam encontrar erros, de modo a poderem ser corrigidos. Quando mais erros se encontrarem, mais erros se corrigem e menos erros existirão.

- **Custo-benefício**

A realização de testes dentro do tempo estipulado auxilia na economia de recursos a longo prazo. Ao se detetar erros num estado inicial, eventuais erros custam menos a corrigir.

- **Redução de Manutenção**

Ao se realizarem testes e se corrigirem os erros, é evitada a necessidade de corrigir situações no futuro.

- **Qualidade do Produto**

O teste assegura a entrega de um produto de qualidade ao cliente.

- **Satisfação do Cliente**

O principal objetivo de qualquer produto de software (uma vez que o cliente é que é o proprietário do produto).

- **Segurança**

Benefício mais vulnerável da realização de um teste de software visto que a segurança representa um dos principais objetivos de qualquer utilizador.

Assim, o presente capítulo cumpre este propósito.

Existem inúmeros tipos de testes de software, sendo que, para o presente projeto foram realizados testes manuais.

### 7.1 Testes Manuais

Os testes manuais de um produto de software representam testes executados de forma manual através da verificação de cenários de teste, sem necessidade de nenhuma ferramenta de automação. Representam 70% de todas as atividades de teste.

Através da realização de testes manuais, um *tester* desempenha o papel do utilizador final. Não é utilizada, como referido, nenhuma ferramenta de automação, embora sejam executados vários cenários de teste de forma manual para identificar comportamentos ou defeitos inesperados.

A realização de testes manuais é, assim, de vital importância, pois permite encontrar erros que apenas o utilizador final poderia encontrar ao interagir com o programa.

Uma vez que os requisitos funcionais do projeto (capítulo 2) foram documentados através de *User Stories*, os critérios de aceitação dos mesmos foram utilizados para a criação dos cenários de teste.

## 7.2 Realização de Testes

Foram realizados testes manuais aos trinta e três cenários de teste criados a partir dos requisitos implementados no capítulo 6 (resultando num total de trinta e três testes). Os testes podem ser consultados nas tabelas E1 a E33, reunidas no ANEXO E. Como pode ser verificado, todos os cenários passaram nos testes efetuados.

A título de exemplo, apresenta-se na tabela 30 o primeiro teste efetuado (teste web).

### Teste *Web*

Tabela 30: Teste T\_01

<b>ID do Teste</b>	T_01
<b>Requisito Referente</b>	REQ_F_10.1
<b>Cenário do Teste</b>	Submeter Diário do Humor
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao diário onde preenche diariamente o valor do seu humor sob a forma de um range slider.</li> <li>3. O valor do humor varia entre zero e cem.</li> <li>4. O aluno seleciona um valor.</li> <li>5. O aluno submete o valor que selecionou.</li> </ol>
<b>Dados do Teste</b>	humor = 30
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno consegue submeter o valor selecionado.</li> <li>2. O aluno deve ficar com o registo do valor que submeteu.</li> <li>3. O aluno só tem possibilidade de realizar uma submissão (por dia).</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

## 8 Planejamento e Gestão do Projeto

O planejamento e a gestão representam duas etapas fulcrais de qualquer projeto: dão direção ao mesmo; garantem o que será, efetivamente, entregue, assegurando que as expectativas são correspondidas; oferecem o valor real relativamente à oportunidade de negócios.

Uma boa gestão de um projeto é de extrema importância para o seu sucesso, devendo ser estipulado tanto no início do mesmo, como ao longo do processo de desenvolvimento, para que, caso venha a ser preciso, seja possível efetuar as alterações necessárias atempadamente. Para tal, identificar os riscos de um projeto é extremamente importante de maneira a criar medidas para poder mitigá-los e evitar quaisquer problemas.

No presente capítulo, os riscos inerentes a este projeto serão apresentados, assim como a metodologia para a sua gestão.

### 8.1 Avaliação dos Riscos

É inevitável que durante o desenvolvimento de um projeto, contratempos aconteçam. Portanto, torna-se imperativo que estes possíveis obstáculos sejam identificados, pois desempenham potenciais riscos que poderão pôr em causa o sucesso do projeto. Reconhecer estes riscos é fundamental para elaborar atitudes e medidas para, aquando do seu acontecimento, o impacto seja minimizado.

Estes riscos podem ser classificados tendo em conta o seu impacto e a sua probabilidade:

- **Impacto:**

- Baixo;
- Elevado;
- Catastrófico.

- **Probabilidade:**

- Pouco Provável;
- Provável;
- Muito Provável.

De seguida, é realizada uma breve descrição dos riscos levantados no âmbito do projeto corrente.

- **R\_01:** Dificuldades no processo de desenvolvimento devido à falta de experiência nas principais linguagens de programação e *frameworks* utilizadas (PHP e *Ionic Angular*), assim como a trabalhar com LMSs, nomeadamente, com o Moodle.
- **R\_02:** Possibilidade de alterar os requisitos numa fase tardia do projeto.
- **R\_03:** Início tardio da fase de implementação.
- **R\_04:** Identificação tardia do verdadeiro âmbito do projeto e onde incidiria.
- **R\_05:** Estimativa extremamente generosa do tempo para realizar as tarefas.

Na tabela 31 apresenta-se o impacto, probabilidade e estratégia de mitigação identificados para cada risco.

Tabela 31: Tabela de riscos e respetiva estratégia de mitigação

ID	Impacto	Probabilidade	Estratégia de Mitigação
R_01	Catastráfico	Provável	Ler documentação; realizar tutoriais; utilizar a assistência da documentação do Moodle, <i>plugins</i> já desenvolvidos e exemplos existentes.
R_02	Baixo	Provável	Só alterar os requisitos que fossem facilmente modificados e que não implicassem grandes alterações.
R_03	Elevado	Provável	Tentar reduzir as <i>sprints</i> de cada tarefa definida para a fase de implementação.
R_04	Elevado	Muito Provável	Redefinir a estratégia do projeto tendo em consideração a sua natureza.
R_05	Elevado	Muito Provável	Redefinir constantemente as <i>sprints</i> definidas, tentando compensar o tempo a mais despendido, na sprint seguinte.

### Risco R\_01

O risco R\_01, encontra-se associado à falta de experiência nas tecnologias utilizadas neste projeto.

Em primeiro lugar, no que diz respeito às principais linguagens de programação e *frameworks* utilizadas para o desenvolvimento de *plugins* no Moodle (PHP) e sua exibição na aplicação móvel (*Ionic Angular*), não havia qualquer tipo de experiência nessas linguagens. Contudo, esta componente do risco foi rapidamente mitigada, não acabando por revelar nenhum impacto no desenvolvimento do projeto.

Quanto à falta de experiência em trabalhar com o Moodle, a estratégia de mitigação passou pela documentação do Moodle, *plugins* já desenvolvidos e exemplos encontrados, na medida em que estes seriam utilizados como assistência de modo a se encontrar respostas a questões de desenvolvimento que fossem surgindo. Acontece que, a maior parte dos *plugins* já existentes tinham muitos anos de desenvolvimento investidos, o que levou a que a interpretação do seu código fosse desafiante. Quanto à documentação do Moodle, esta, apesar de vasta, também se encontrava desatualizada, para além de ser pobre em exemplos. A junção destes acontecimentos levou a que esta questão não fosse facilmente amenizada, representando mesmo a maior dificuldade encontrada neste projeto. Para aprender a desenvolver, com a integração com o Moodle, o determinado para cada requisito, muitas vezes, por muita documentação existente, só foi possível, literalmente, pela aprendizagem “tentativa - erro”. Se fossem tarefas desintegradas do Moodle era algo extremamente simples de fazer, mas, uma vez que este tem “regras” muito próprias, houve necessidade de as aprender e de as aplicar. Assim, tudo o que seriam tarefas relativamente fáceis de desenvolver, revelaram-se difíceis.

### Risco R\_02

O risco R\_02 está relacionado com a necessidade de alterar os requisitos em fases tardias do projeto. A solução mais radical seria a de não admitir a introdução de novos requisitos em fases tardias, ou não permitir a alteração de requisitos anteriores. Contudo, para este projeto, foi dada a preferência por modificar ou alterar requisitos se implicassem alterações extremamente simples e que não pusessem em causa o tempo de desenvolvimento. Porém verificou-se que o tempo gasto para a execução dessas alterações foi subestimado pelos motivos expostos no risco R\_01: novos pedidos do cliente, que seriam fáceis de implementar, acabavam por levar o triplo

do tempo, pela questão da integração com a plataforma Moodle.

Devido à vontade de ir ao encontro das expectativas do cliente, muitas vezes tornou-se difícil, enquanto se desenvolviam essas alterações, tomar a decisão de abdicar das mesmas, ou não as incluir, para que no fim houvesse mais funcionalidades desenvolvidas e o resultado do trabalho apresentado fosse melhor.

### **Risco R\_03**

Este risco passa pelo facto de o processo de desenvolvimento dos requisitos se ter iniciado tardiamente, limitando o tempo disponível para tal. Isto porque, uma parte do tempo existente para o projeto, foi dispendido na escolha da plataforma selecionada e no estudo e adaptação das funcionalidades da plataforma a usar (o LMS Moodle). Esta fase mereceu o tempo dedicado, pois acabou por condicionar todo o desenvolvimento do projeto (e a sua verdadeira essência).

### **Risco R\_04**

Uma característica do presente projeto foi o facto de não ter sido iniciado com um objetivo e requisitos predefinidos. O projeto iniciou-se apenas com a necessidade de o cliente de possuir uma plataforma web.

Assim, antes de passar à fase da sua implementação, houve necessidade de:

- Explorar soluções que pudessem ser adotadas (começando com OSSs e terminando em LMSs);
- Explorar LMSs, efetuar a sua comparação e selecionar a plataforma a adotar;
- Instalar e configurar versões (definitivas e de desenvolvimento) do Moodle nos mais diversos ambientes: Docker, Windows, macOS, CentOS, Linux;
- Passar finalmente à fase de estudo do Moodle e análise da forma como as funcionalidades adicionais seriam integradas.

Por esse motivo, os requisitos e todas as decisões que foram tomadas de maneira a entregar a melhor solução possível ao cliente, acabaram por interferir com a identificação do âmagdo do projeto, para além de influenciar o seu verdadeiro *scope*.

### **Risco R\_05**

Os riscos acima referidos, acabaram por levar à identificação de um novo risco: estimativa extremamente generosa do tempo necessário para realizar as tarefas.

Por se ter iniciado a fase de desenvolvimento tarde e o âmagdo do projeto ter sido identificado igualmente tarde, o tempo disponível ficou reduzido. Para além disso, julgou-se que as tarefas de alguns requisitos seriam relativamente fáceis de cumprir num curto espaço de tempo, o que acabou por não ser verdade, devido às falhas identificadas na documentação do Moodle. Perdeu-se muito tempo a aprender por meio de “tentativa - erro” e a solucionar as dificuldades que iam surgindo.

## **8.2 Gestão do Projeto**

Tendo em conta as datas apresentadas para o projeto, foram estabelecidos prazos para concluir as tarefas (prazos *versus* entregas). Assim, o planeamento do mesmo foi dividido em dois semestres (Primeiro e Segundo Semestres do ano letivo em questão), sendo definido no início de cada um e adaptado ao longo do seu desenvolvimento, aquando da sua necessidade.

No **Primeiro Semestre**, o trabalho foi estruturado em *time-boxes*, técnica de gestão que prioriza os prazos em relação às entregas [2], sendo que para cada uma foram definidos objetivos a cumprir (tarefas e iterações individuais). Utilizando esta técnica, parte-se do pressuposto de que todas as entregas solicitadas só serão concluídas até à data de conclusão definida, acabando por incentivar a flexibilidade e promover uma melhoria contínua. Diariamente também era realizada uma introspectiva e análise do trabalho realizado e do trabalho ainda por realizar e apontadas manualmente num *log* pessoal. Sendo o trabalho desenvolvido apenas por uma única pessoa (a autora da Tese de Mestrado), não houve a necessidade de reuniões de equipa, mantendo-se, contudo, o contacto frequente com o cliente.

No **Segundo Semestre** utilizou-se um modo iterativo inspirado no *Scrum* (não se podendo dizer que a *framework* utilizada passa mesmo pelo *Scrum*, visto que, no caso deste projeto, os papéis de “*Product Owner*”, “*Scrum Team*” e “*Scrum Master*” inerentes ao *Scrum*, foram todos desempenhados por uma única pessoa). Contudo, esta foi a metodologia escolhida por facilitar a gestão das tarefas; permitir um bom registo e monitorização do trabalho; possibilitar uma rápida resposta a mudanças e viabilizar uma colaboração direta com o cliente. Cada tarefa necessária para o cumprimento dos objetivos deste projeto, foi representada por uma *sprint*. A aplicação utilizada, devido à familiaridade resultante da sua utilização em projetos passados, foi o “*Trello*”. As tarefas foram organizadas num “*Scrum board*” e integrou-se a ferramenta “*Team Gantt*” para a criação e monitorização do “*Gantt Chart*”.

### 8.2.1 Calendarização Planeada

Primeiramente, de maneira a ser possível definir o planeamento do projeto, foi necessário ter em consideração as datas e o plano de trabalhos previamente estabelecidos.

As data estipuladas foram as seguintes:

- **16 de setembro de 2019** - Início do Estágio;
- **20 de janeiro de 2020** - Submissão do Relatório Intermédio;
- **27 de janeiro de 2020** - Defesa pública da 1<sup>a</sup> parte do Estágio;
- **29 de junho de 2020** - Entrega do Relatório Final;
- **6 a 10 de julho de 2020** - Apresentação e Defesa Final.

No início do projeto, também foram apresentados os planos de trabalhos estipulados tanto para o Primeiro, como para o Segundo Semestre.

#### Primeiro Semestre

No caso do Primeiro Semestre:

1. Especificação da solução a implementar junto dos utilizadores finais;
2. Análise do estado da arte em plataformas de gestão de conteúdos;
3. *Gap analysis* entre as funcionalidades disponíveis e desejadas;
4. Seleção da plataforma a adotar;
5. Identificação das funcionalidades extra a desenvolver;
6. Escrita do relatório intermédio.

Considerando este plano de trabalhos, e as datas estipuladas, foi construído o planeamento inicial para realizar os objetivos do Primeiro Semestre. Este encontra-se na figura 26, onde estão

apresentadas as tarefas a realizar e as metas temporais para estarem concluídas.

Tencionou-se que o tempo para a realização de cada tarefa durasse catorze dias, para que a escrita do relatório pudesse ser realizada em tempo útil.

Relativamente à figura 26, a azul encontram-se representadas as *deadlines* a ter em atenção neste semestre, de acordo com as datas apresentadas no início deste capítulo.

Primeiro Semestre		
16/09/2019 - 27/01/2020		
Datas estipuladas:	Nº de dias:	
16/9/2019 - 30/9/2019	14 dias	Identificação, Estruturação e Priorização dos Requisitos da Plataforma
1/10/2019 - 15/10/2019	14 dias	Estado de Arte em Plataformas de Gestão de Conteúdos
16/10/2019 - 30/10/2019	14 dias	Pesquisa e Comparação de Soluções Open-Source Apropriadas
31/10/2019 - 14/11/2019	14 dias	Gap Analysis entre as Funcionalidades Disponíveis e as Desejadas
15/11/2019 - 29/11/2019	14 dias	Seleção da Plataforma a Adotar
30/11/2019 - 19/1/2020	51 dias	Escrita e Revisão do Relatório Intermédio
<b>20/1/2020</b>	-	<b>Entrega do Relatório Intermédio</b>
21/1/2020 - 26/1/2020	5 dias	Preparação da Defesa do Relatório Intermédio
<b>27/1/2020</b>	-	<b>Defesa do Relatório Intermédio</b>

Figura 26: Primeiro Semestre - planeamento estipulado

## Segundo Semestre

No caso do Segundo Semestre, o plano de trabalhos que tinha sido inicialmente estipulado no início do estágio passava pelos seguintes pontos:

1. Instalação, configuração e lançamento da plataforma;
2. Inserção dos conteúdos de base;
3. Formação dos utilizadores;
4. Criação de recursos de treino;
5. *Beta-testing*;
6. Evolução da plataforma de acordo com as necessidades de utilização;
7. Desenvolvimento dos restantes componentes do sistema;
8. Escrita do relatório final.

Contudo, o planeamento sob a forma de *Gantt Chart*, utilizando o “Trello” (*Scrum board*) em conjunto com o “Team Gantt”, dada a natureza deste projeto e ao *feedback* recebido na defesa do Relatório Intermédio, não seguiu a linha deste plano de trabalhos, havendo a necessidade de este último ser reformulado. Assim, o plano de trabalhos inicial estipulado para o *Segundo Semestre* (figura 27) passou a ser:

1. Instalação, configuração e lançamento da plataforma (configuração do seu domínio);
2. Refazer requisitos sob a forma de *User Stories*;
3. Afinar *User Stories*;
4. Recolher e documentar os requisitos relativos à gamificação;
5. Nova identificação das funcionalidades adicionais a implementar;
6. Fase de desenvolvimento;
7. Fase de testes;

8. Escrita do Relatório Final;
9. Formação dos utilizadores.

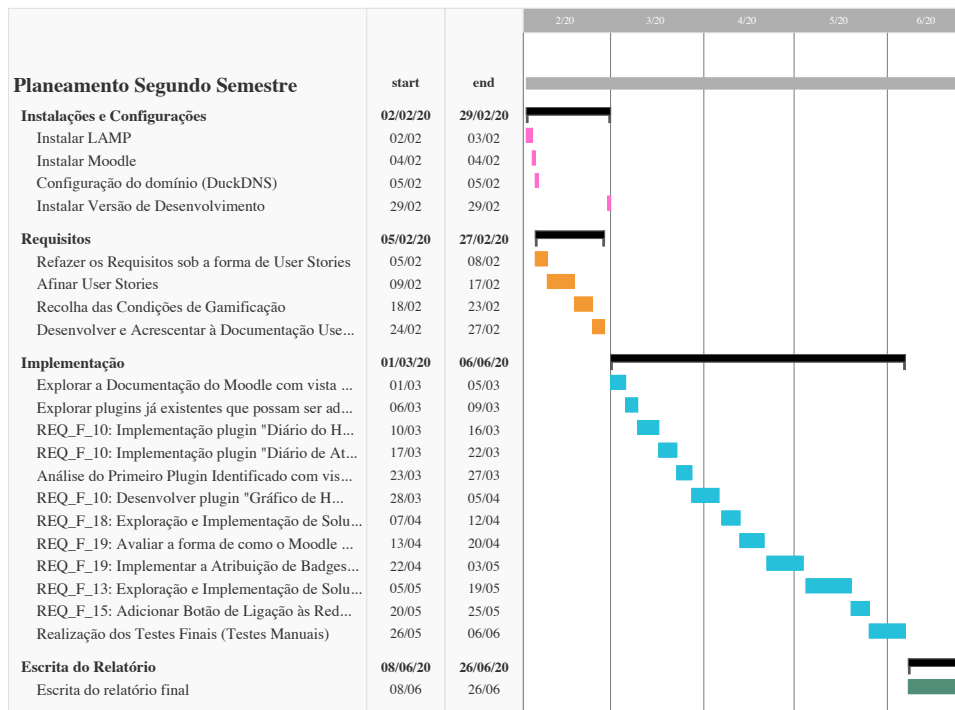


Figura 27: Segundo Semestre - planeamento estipulado

### 8.2.2 Modificações no Planeamento

Ao longo do processo de desenvolvimento deste projeto houve necessidade de fazer algumas alterações no planeamento criado inicialmente devido, tanto a modificações inevitáveis, inerentes à natureza do projeto, como à existência de alguns contratemplos.

#### Primeiro Semestre

No caso do Primeiro Semestre, foi necessário fazer alguns ajustes, como: a criação de uma fase de retificação e validação dos requisitos; a existência de um dia de *workshop*/formação aos futuros professores/psicólogos que irão utilizar a plataforma, de maneira a ensinar como é que estes poderiam inserir conteúdos na plataforma de forma independente. Desta forma, foram acrescentadas ao plano de trabalhos, as tarefas:

- Validação “final” dos requisitos e sua priorização;
- Instalação de uma versão do OSS adotado a ser utilizado no *workshop*;
- Realização do *workshop* (*milestone*).

Na tarefa “Instalação de uma versão do OSS adotado a ser utilizado no *workshop*”, uma vez que o servidor solicitado para alojar o *website* ainda não tinha sido disponibilizado, foi necessário preparar um *website* do Moodle temporário (apenas com o intuito de ser utilizado no *workshop*). A solução temporária arranjada foi a utilização do sistema de contentores, *Docker* (“*Docker Toolbox for Windows*”) para virtualização de uma imagem do Moodle [16] e configurar de maneira a que os professores/psicólogos pudessem aceder ao *container* através das suas máquinas na rede local.



O surgimento deste *workshop* manifestou-se pela necessidade de criar, logo de início, uma maior familiaridade dos potenciais utilizadores com a plataforma e obter a sua aprovação, avaliando-se, nesta primeira fase, a sua interação com a plataforma.

Na figura 28 apresenta-se o planeamento real, por *time-boxes*, do Primeiro Semestre.

Primeiro Semestre		
16/9/2019 - 27/1/2020		
Datas estipuladas:	Nº de dias:	
16/9/2019 - 30/9/2019	14 dias	Identificação, Estruturação e Priorização dos Requisitos da Plataforma
1/10/2019 - 15/10/2019	14 dias	Estado de Arte em Plataformas de Gestão de Conteúdos
16/10/2019 - 09/11/2019	24 dias	Pesquisa e Comparação de Soluções Open-Source Apropriadas
10/11/2019 - 11/11/2019	1 dia	Validação "Final" dos Requisitos e sua Priorização
12/11/2019 - 22/11/2019	10 dias	Gap Analysis entre as Funcionalidades Disponíveis e as Desejadas
23/11/2019 - 29/11/2019	6 dias	Seleção da Plataforma a Adotar
30/11/2019 - 16/12/2019	16 dias	Preparação Workshop (Instalar Plataforma e Preparar Formação)
<b>17/12/2019</b>	-	<b>Realização da Formação</b>
18/12/2019 - 19/1/2020	32 dias	Escrita e Revisão do Relatório Intermédio
<b>20/1/2020</b>	-	<b>Entrega do Relatório Intermédio</b>
21/1/2020 - 26/1/2020	5 dias	Preparação da Defesa do Relatório Intermédio
<b>27/1/2020</b>	-	<b>Defesa do Relatório Intermédio</b>

Figura 28: Primeiro Semestre - planeamento real

## Segundo Semestre

Relativamente às modificações realizadas no planeamento do Segundo Semestre, estas foram marcadas pela existência de contratemplos.

- Necessidade de Aquisição de uma Nova Máquina de Trabalho (PC)

Em primeiro lugar, um contratempo existente foi o facto de se ter de substituir a própria máquina pessoal de desenvolvimento deste projeto (PC da autora da Tese de Mestrado). Este acontecimento, para além da espera associada à aquisição de uma máquina nova, levou a que, para o projeto não ficar “parado”, tivesse de se utilizar máquinas alheias, o que limitou e dificultou o tempo de utilização das mesmas, assim como a qualidade do trabalho realizado durante este período.

- Más Decisões no Planeamento do Projeto por Falta de Conhecimento no Moodle

Em determinados momentos, estipulou-se passos para a concretização de certos objetivos, por se julgar que seriam os corretos e eventualmente se percebeu que seria necessário seguir por outras vias. Muito disso se deveu ao facto de o Moodle acabar por se revelar o objeto de estudo e de trabalho deste projeto, com o qual não havia qualquer experiência ou conhecimento - além do superficial. Como tem vindo a ser referido, o Moodle é um *software* com regras e convenções extremamente próprias que exigiram um estudo aprofundado antes de se passar à fase de desenvolvimento propriamente dita.

Um exemplo desta dificuldade passou pelo momento em que se pretendeu seguir para o desenvolvimento dos requisitos (funcionalidades adicionais). Sendo necessário estudar a forma como estas funcionalidades seriam adicionadas e percebendo que esta integração passaria pelo desenvolvimento dos requisitos sob a forma de *plugins*, a estratégia que se adotou foi de procurar e selecionar *plugins* já criados e disponíveis na diretoria de *plugins* do Moodle, que fossem ao encontro, ou que fossem semelhantes, às funcionalidades dos requisitos estipulados. Isto para que apenas fosse necessário realizar alguns ajustes ou alterações nos seus códigos de modo a se obter o pretendido. Contudo, a seleção destes *plugins* levou mais tempo do que o esperado, visto que, em alguns casos, dificilmente se encontrou *plugins* semelhantes aos requisitos do projeto (nomeadamente referentes à

existência de um “Diário de Humor”, “Diário de Atividades” e “Gráfico de Humor e Atividades”). No entanto, após se reunir alguns *plugins* considerados pertinentes no contexto do projeto, o seu estudo revelou-se inútil nesta fase inicial, na medida em que estes *plugins* tinham demasiadas informações e funcionalidades que não eram necessárias. Isto dificultou bastante o seu estudo e a forma como as funcionalidades desejadas seriam implementadas. Dado estes fatores, decidiu-se, tardiamente, optar por desenvolver os *plugins* referidos de “raíz”. Todavia, os *plugins* acabaram por ser úteis para, solucionar outras questões que surgiram.

- Documentação Pobre

Ao se criar *plugins* de “raíz”, maior foi a necessidade de estudar a documentação do Moodle, nomeadamente, os tipos de *plugins* que o Moodle disponibilizava e selecionar aquele que melhor se adequasse aos requisitos. Após se selecionar o tipo de *plugins*, foi também necessário estudar as tais convenções para o desenvolvimento de *plugins* deste tipo e, para cada requisito, procurar na documentação do Moodle a existência de APIs que facilitassem a integração de certos componentes (como foi o caso do “*Form API*”, “*Chart API*”, ou “*Web Service API*” por exemplo).

No entanto, como já referido na secção 1.3, para além de a documentação do Moodle se encontrar desatualizada, era pobre em exemplos, levando a que muitos problemas fossem solucionados apenas através da aprendizagem “tentativa - erro”, o que foi um fator que atrasou bastante o processo de desenvolvimento. Pormenores como pastas e ficheiros constituintes de um *plugin*, que necessitavam de ter um determinado nome e se localizar em diretorias específicas; passos que deviam ser incluídos, que não se encontravam especificados na documentação; pormenores como, na estruturação e definição dos *web services* e suas funções, da versão *mobile* dos *plugins* desenvolvidos, ser necessário definir o *array* de funções antes do *array* de serviços, revelavam-se motivos suficientes para um *plugin* não funcionar. Aquando destas dificuldades, os *plugins* que anteriormente se tinham reunido e estudado revelaram-se importantes pois, pela sua análise e seu uso como exemplo, foi possível por vezes vezes solucionar estas questões.

- Necessidade de Repetição de Tarefas

Outro contratempo passou pelo facto de a instalação e configuração da pilha Linux, Apache, MySQL/MariaDB, PHP (LAMP) no servidor onde a plataforma SMS-eSaúde estaria alojada, (e assim, o próprio Moodle), ter sido uma tarefa que se efetuou várias vezes (sem considerar as versões para criação do ambiente de desenvolvimento, tanto em Windows, como em macOS e da versão apresentada no *workshop*).

O *website* inicial, instalado num servidor remoto da UC, era extremamente lento, impossibilitando completamente a sua utilização (no mínimo dez segundos para obter uma resposta do sistema). Uma vez que este problema não conseguiu ser resolvido pelas entidades responsáveis, uma nova máquina teve de ser solicitada, desta vez, ao *Helpdesk* do DEI. Este processo exigiu, novamente, uma nova espera para que esta fosse concedida, para, de seguida e, por fim, se instalar e configurar a versão final da plataforma SMS-eSaúde (instalação e configuração da pilha LAMP, assim como do Moodle).

Para além disso, o requisito REQ\_F.10.2 teve de ser desenvolvido novamente por não ir ao encontro do que o cliente desejava (como apresentado na secção 6.3.1.2).

- Cache do Moodle

Outra questão que dificultou o processo de desenvolvimento passou pelas “Caches do Moodle” [82]. O facto de o Moodle armazenar temas, *strings* de linguagem, filtros e outras informações calculadas e armazenadas pelo *browser*, fazia com que sempre que se efetuasse qualquer alteração no código do Moodle, durante o processo de desenvolvimento, fosse necessário limpar as caches do Moodle (Administração do Site >Desenvolvimento >Limpar caches), processo que demorava sempre entre 10 a 20 segundos. Esta questão para além de atrapalhar todo o processo de desenvolvimento, também o tornou fastidioso e cansativo, quebrando o ritmo do mesmo. Mesmo quando, no caso do Google *Chrome*, se desativava as caches nas opções de desenvolvimento, este problema não era resolvido. Para além disso, muitas vezes, a própria limpeza das caches não era suficiente, levando a que, se tivesse de modificar a versão do *plugin* (para obrigar que o Moodle realizasse a sua atualização). Em casos mais extremos, houve mesmo a necessidade de apagar o *plugin* e instalá-lo novamente.

Este fator ainda piorou aquando do desenvolvimento na aplicação móvel. Muitas vezes, nenhuma das soluções relatadas acima funcionava, levando a que, em certas situações, não se soubesse se os erros que iam surgindo ficavam ou não resolvidos visto que, à medida que se tentava ter essa confirmação, o resultado que se obtinha na aplicação móvel nunca estava de acordo com o código e as últimas atualizações feitas no mesmo.

Todas estas situações levaram a que o projeto demorasse mais tempo do que o estipulado, mais tarefas tivessem de ser acrescentadas e os tempos estimados para executar certas tarefas fossem constantemente modificados, o que, na prática, levou a que o projeto se estendesse no tempo.

O planeamento real do Segundo Semestre é caracterizado na figura 29, podendo-se observar que as datas de entrega e defesa do projeto, foram alteradas:

- **8 de setembro de 2020** - Entrega do Relatório Final;
- **14 de setembro de 2020** - Apresentação e Defesa Final.

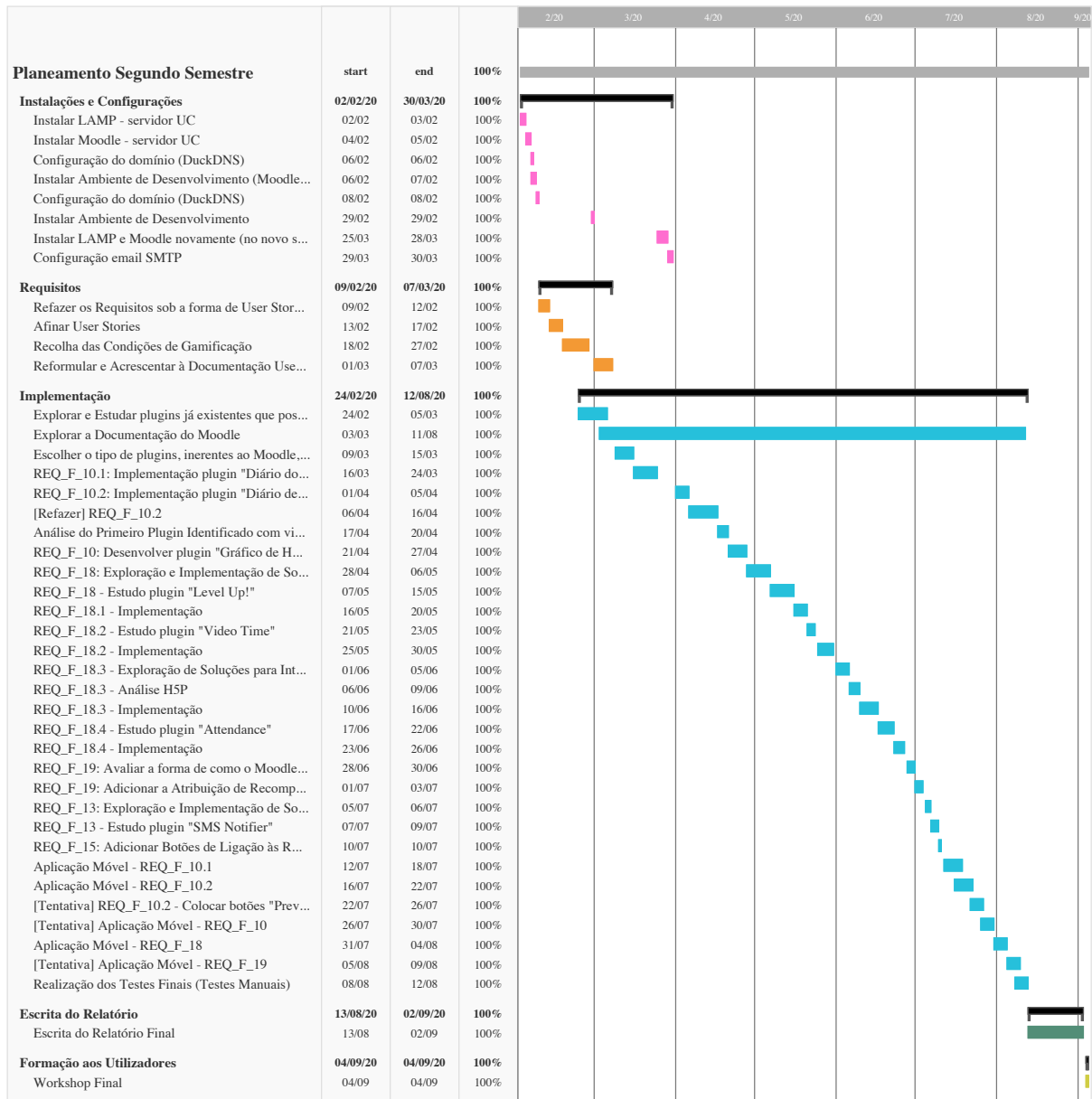


Figura 29: Segundo Semestre - planeamento real

## 9 Conclusão

Com o objetivo de apresentar uma plataforma digital (plataforma SMS-eSaúde) que auxiliasse investigadores da FPCE da UC na implementação e sustentabilidade do Projeto SMS, desenvolveu-se o presente projeto a partir da adaptação do LMS Moodle (criação e adaptação de *plugins*).

Vários foram os contratempos surgidos que provocaram atrasos na execução do projeto: dificuldades relativas à documentação do Moodle; questões relacionadas com a cache do Moodle; necessidade de repetir tarefas já realizadas; inexperiência no Moodle.

É possível que se a fase de desenvolvimento tivesse sido iniciada mais cedo, estes contratempos não tivessem tido expressão no decorrer do trabalho. No entanto, o tempo dispendido na procura de uma plataforma e a decidir qual a plataforma a adotar, que atrasou o início do processo de desenvolvimento, não foi em vão, já que estes foram procedimentos vitais para definir o rumo e o verdadeiro âmagio do projeto.

Contudo, independentemente dos contratempos ocorridos, considera-se que o trabalho realizado foi bem sucedido e gratificante. Todos os requisitos foram implementados e, para além disso, ainda se possibilitou, como um “bónus”, a existência de uma aplicação móvel, compatível com alguns dos novos recursos criados.

Aquilo que começou por ser um projeto que tinha única e exclusivamente o objetivo de apresentar uma plataforma digital ao cliente, tornou-se mais do que isso. Proporcionou um enriquecimento quer a nível científico, quer a nível pessoal, ao permitir:

- Lidar com um cliente real;
- Explorar a definição e conceitos de gamificação e como poderiam ser aplicados;
- Estudar OSSs existentes no mercado que pudessem ser mais benéficos para o cliente, percebendo que se deveria utilizar um CMSs e, posteriormente, um LMS;
- Estudar e comparar LMSs existentes;
- Selecionar o Moodle, o principal objeto de estudo e de trabalho deste projeto;
- Instalar e configurar a plataforma Moodle em diversos ambientes: *Docker*, *Windows*, *CentOS*, *Ubuntu* (e pilha LAMP);
- Estudar a arquitetura do Moodle, a sua documentação e qual seria a melhor forma de integrar código adicional no mesmo;
- Estudar os Módulos de Atividades e Recursos do Moodle;
- Estudar e integrar H5P;
- Seguir e integrar uma série de APIs do Moodle;
- Estudar os *plugins* do Moodle e selecionar quais os tipos de *plugins* que faziam mais sentido, de acordo com os requisitos;
- Desenvolver *plugins* de raiz e adaptar *plugins* já existentes em versões *web* e *mobile*, utilizando diversas tecnologias e linguagens de programação (PHP, HTML, JavaScript, CSS, *Ionic Angular*, por exemplo);
- Realizar o *design* e aparência dos respetivos *plugins* de acordo com *mockups* criados;
- Estudar *plugins* já desenvolvidos com alguma complexidade e customizar esses mesmos *plugins* para irem ao encontro dos requisitos desejados.

*Esta página foi intencionalmente deixada em branco.*

## Referências

- [1] 1233-1998 - *IEEE Guide for Developing System Requirements Specifications*. IEEE
- [2] Miranda E, 'Time Boxing Planning', ACM SIGSOFT Software Engineering Notes, 2011, vol: 36 (6), pp: 1
- [3] Huotari K, Hamari J, 'Defining gamification: a service marketing perspective', Proceeding of the 16th International Academic MindTrek Conference, MindTrek, 2012, p: 17
- [4] Paharia R, 'Who coined the term "gamification"?', Quora, 2010, <http://goo.gl/CvcMs>
- [5] Muriel D, Crawford G 'Video Games as Culture: Considering the Role and Importance of Video Games in Contemporary Society', Taylor and Francis, 2018, pp: 1-194
- [6] Zichermann G, Cunningham C, 'Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps.', O'Reilly, Sebastopol, 2011
- [7] Deterding S, Dixon D, Khaled R, Nacke L, 'From Game Design Elements to Gamefulness: Defining "Gamification"', Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek, 2011, pp: 9-15
- [8] Hamari J, Koivisto J, Sarsa H, 'Does gamification work? - A literature review of empirical studies on gamification', Proceedings of the Annual Hawaii International Conference on System Sciences, IEEE Computer Society, 2014, pp: 3025-3034
- [9] Cheong C, Cheong F, Filippou J, 'Quick Quiz: A Gamified Approach for Enhancing Learning', 2013, pp: 206
- [10] Hamari J, Koivisto J, Motivations S, 'Social Motivations To Use Gamification: An Empirical Study Of Gamifying Exercise', 2013, pp: 1-12
- [11] Li W, Grossman T, Fitzmaurice G, 'GamiCAD: A gamified tutorial system for first time AutoCAD users', Association for Computing Machinery (ACM), 2012, pp: 103
- [12] Turnbull D, Chugh R, Luck J, 'Learning Management Systems: An Overview', Encyclopedia of Education and Information Technologies, Springer International, 2019, pp: 1-7
- [13] De Smet C, Valcke M, Schellens T, De Wever B, Vanderlinde R, 'A qualitative study on learning and teaching with learning paths in a learning management system', Journal of Social Science Education, 2016, vol: 15 (1), pp: 27-37
- [14] Kulshrestha T, Kant A, 'Benefits of Learning Management System (LMS) in Indian Education', International Journal of Computer Science & Engineering Technology (IJCSET), 2013, vol: 4 (8), pp: 1153-1164
- [15] Requisitos de um LMS  
<https://www.ispringsolutions.com/blog/lmsRequirements/> [Online; Acedido: Set 28, 2019]
- [16] Imagem Bitnami do Moodle <https://hub.docker.com/r/bitnami/moodle/> [Online; Acedido: Nov 30, 2019]
- [17] Docker  
<https://docs.docker.com/install/> [Online; Acedido: Nov 30, 2019]
- [18] Últimos lançamentos do Moodle  
<https://download.moodle.org/releases/latest/> [Online; Acedido: Nov 21, 2019]

- [19] Comunicação entre componentes do Moodle  
[https://docs.moodle.org/dev/Communication\\_Between\\_Components](https://docs.moodle.org/dev/Communication_Between_Components) [Online; Acedido: Mar 2, 2020]
- [20] Arquitetura de Aplicações *Open- Source*  
<http://aosabook.org/en/moodle.html> [Online; Acedido: Mar 3, 2020]
- [21] Arquitetura do Moodle  
[https://docs.moodle.org/dev/Moodle\\_architecture](https://docs.moodle.org/dev/Moodle_architecture) [Online; Acedido: Mar 3, 2020]
- [22] Papéis do Moodle  
<https://docs.moodle.org/dev/Roles> [Online; Acedido: Nov 19, 2019]
- [23] Contextos do Moodle  
<https://docs.moodle.org/39/en/Context> [Online; Acedido: Mar 06, 2020]
- [24] API de acesso do Moodle (*Capabilities*)  
[https://docs.moodle.org/dev/Access\\_API#require\\_capability.28.29](https://docs.moodle.org/dev/Access_API#require_capability.28.29) [Online; Acedido: Mar 07, 2020]
- [25] Papéis e *Capabilities* do Moodle  
[https://docs.moodle.org/19/en/Roles\\_and\\_capabilities](https://docs.moodle.org/19/en/Roles_and_capabilities) [Online; Acedido: Mar 07, 2020]
- [26] Lista de *Capabilities* do Moodle  
<https://docs.moodle.org/39/en/Category:Capabilities> [Online; Acedido: Mar 07, 2020]
- [27] API de *output* do Moodle  
[https://docs.moodle.org/dev/Output\\_API](https://docs.moodle.org/dev/Output_API) [Online; Acedido: Abr 22, 2020]
- [28] API de gráficos do Moodle  
[https://docs.moodle.org/dev/Charts\\_API](https://docs.moodle.org/dev/Charts_API) [Online; Acedido: Abr 19, 2020]
- [29] API de manipulação de dados do Moodle  
[https://docs.moodle.org/dev/Data\\_manipulation\\_API](https://docs.moodle.org/dev/Data_manipulation_API) [Online; Acedido: Mar 21, 2020]
- [30] API de eventos do Moodle  
[https://docs.moodle.org/dev/Events\\_API](https://docs.moodle.org/dev/Events_API) [Online; Acedido: Mai 13, 2020]
- [31] *Logging* do Moodle  
<https://docs.moodle.org/39/en/Logging> [Online; Acedido: Mai 13, 2020]
- [32] Lista de eventos do Moodle  
[https://docs.moodle.org/39/en/Events\\_list\\_report](https://docs.moodle.org/39/en/Events_list_report) [Online; Acedido: Mai 13, 2020]
- [33] *Frankenstyle* - Convenção da Nomenclatura de *plugins* do Moodle  
<https://docs.moodle.org/dev/Frankenstyle> [Online; Acedido: Mar 3, 2020]
- [34] Tipos de *plugins* do Moodle  
[https://docs.moodle.org/dev/Plugin\\_types](https://docs.moodle.org/dev/Plugin_types) [Online; Acedido: Mar 9, 2020]
- [35] *Plugins* do tipo “Módulo de Atividades e Recursos” do Moodle  
[https://docs.moodle.org/dev/Activity\\_modules](https://docs.moodle.org/dev/Activity_modules) [Online; Acedido: Mar 10, 2020]
- [36] *Plugins* do tipo “Bloco” do Moodle  
<https://docs.moodle.org/dev/Blocks> [Online; Acedido: Mar 10, 2020]
- [37] *Plugins* do tipo “Tema” do Moodle  
<https://docs.moodle.org/dev/Themes> [Online; Acedido: Mar 10, 2020]
- [38] *Plugins* do tipo “Formato de Curso” do Moodle  
[https://docs.moodle.org/dev/Course\\_formats](https://docs.moodle.org/dev/Course_formats) [Online; Acedido: Mar 10, 2020]



- [39] Plugins do tipo “Inscrição” do Moodle  
[https://docs.moodle.org/dev/Enrolment\\_plugins](https://docs.moodle.org/dev/Enrolment_plugins) [Online; Acedido: Mar 10, 2020]
- [40] Plugins do tipo “Autenticação” do Moodle  
[https://docs.moodle.org/dev/Authentication\\_plugins](https://docs.moodle.org/dev/Authentication_plugins) [Online; Acedido: Mar 10, 2020]
- [41] Plugins do tipo “Repositório” do Moodle  
[https://docs.moodle.org/dev/Repository\\_plugins](https://docs.moodle.org/dev/Repository_plugins) [Online; Acedido: Mar 10, 2020]
- [42] Plugins do tipo “Filtro” do Moodle  
<https://docs.moodle.org/dev/Filters> [Online; Acedido: Mar 10, 2020]
- [43] Classe “block\_base” para a criação de um *plugin* do tipo “Bloco” do Moodle  
[https://wmski.org/api/3.8/d0/df2/classblock\\_\\_base.html](https://wmski.org/api/3.8/d0/df2/classblock__base.html) [Online; Acedido: Mar 16, 2020]
- [44] Descrição da classe “block\_base” para a criação de um *plugin* do tipo “Bloco” do Moodle  
[https://wmski.org/api/3.8/d0/df2/classblock\\_\\_base.html#details](https://wmski.org/api/3.8/d0/df2/classblock__base.html#details) [Online; Acedido: Mar 16, 2020]
- [45] Função “get\_content()” do Moodle  
[https://docs.moodle.org/dev/Blocks/Appendix\\_A#get\\_content.28.29](https://docs.moodle.org/dev/Blocks/Appendix_A#get_content.28.29) [Online; Acedido: Mar 16, 2020]
- [46] Variável “\$this->content” do Moodle  
[https://docs.moodle.org/dev/Blocks/Appendix\\_A#.24this-.3Econtent](https://docs.moodle.org/dev/Blocks/Appendix_A#.24this-.3Econtent) [Online; Acedido: Mar 16, 2020]
- [47] Funcionalidades da aplicação *mobile* do Moodle  
[https://docs.moodle.org/39/en/Moodle\\_app\\_features](https://docs.moodle.org/39/en/Moodle_app_features) [Online; Acedido: Jul 12, 2020]
- [48] Suporte para *plugins* da versão *mobile* do Moodle  
[https://docs.moodle.org/dev/Mobile\\_support\\_for\\_plugins](https://docs.moodle.org/dev/Mobile_support_for_plugins) [Online; Acedido: Jul 11, 2020]
- [49] Componentes *Ionic*  
<https://ionicframework.com/docs/components> [Online; Acedido: Jul 12, 2020]
- [50] Configurações de *quizzes* do Moodle  
[https://docs.moodle.org/39/en/Quiz\\_settings](https://docs.moodle.org/39/en/Quiz_settings) [Online; Acedido: Mai 16, 2020]
- [51] Ferramenta de acompanhamento de conclusão de uma atividade do Moodle  
[https://docs.moodle.org/39/en/Activity\\_completion](https://docs.moodle.org/39/en/Activity_completion) [Online; Acedido: Mai 22, 2020]
- [52] Como utilizar a ferramenta de acompanhamento de conclusão de uma atividade do Moodle  
[https://docs.moodle.org/39/en/Using\\_Activity\\_completion](https://docs.moodle.org/39/en/Using_Activity_completion) [Online; Acedido: Mai 22, 2020]
- [53] API da ferramenta de acompanhamento de conclusão de uma atividade do Moodle  
[https://docs.moodle.org/dev/Activity\\_completion\\_API](https://docs.moodle.org/dev/Activity_completion_API) [Online; Acedido: Mai 22, 2020]
- [54] Como utilizar “*Moodle Badges*”  
[https://docs.moodle.org/38/en/Using\\_badges](https://docs.moodle.org/38/en/Using_badges) [Online; Acedido: Out 2, 2019]
- [55] Como gerir “*Moodle Badges*”  
[https://docs.moodle.org/38/en/Managing\\_badges](https://docs.moodle.org/38/en/Managing_badges) [Online; Acedido: Out 3, 2019]
- [56] *Container* Bitnami do Moodle  
<https://bitnami.com/stack/moodle/containers> [Online; Acedido: Fev 28, 2020]

- [57] Instalador da pilha Moodle (Bitnami)  
<https://bitnami.com/stack/moodle/installer> [Online; Acedido: Fev 29, 2020]
- [58] “Time API” do Moodle  
[https://docs.moodle.org/dev/Time\\_API](https://docs.moodle.org/dev/Time_API) [Online; Acedido: Mar 19, 2020]
- [59] API de formulários do Moodle  
[https://docs.moodle.org/dev/Form\\_API](https://docs.moodle.org/dev/Form_API) [Online; Acedido: Mar 17, 2020]
- [60] *Plugin* do tipo “Bloco” (“*Blocks: Analytics graphs*”)  
[https://moodle.org/plugins/block\\_analytics\\_graphs](https://moodle.org/plugins/block_analytics_graphs) [Online; Acedido: Fev 24, 2020]
- [61] *Plugin* do tipo “Grade Reports” (“*Grade Reports: Grade Distribution*”)  
[https://moodle.org/plugins/gradereport\\_gradedist](https://moodle.org/plugins/gradereport_gradedist) [Online; Acedido: Fev 24, 2020]
- [62] *Plugin* do tipo “Report” (“*Report: Overview Statistics*”)  
[https://moodle.org/plugins/report\\_overviewstats](https://moodle.org/plugins/report_overviewstats) [Online; Acedido: Fev 24, 2020]
- [63] *Plugin* do tipo “Bloco” (“*Blocks: Level up! - Gamification*”)  
[https://moodle.org/plugins/block\\_xp](https://moodle.org/plugins/block_xp) [Online; Acedido: Abr 29, 2020]
- [64] *Plugin* do tipo “Módulo de Atividades e Recursos” (“*Mod: Video Time*”)  
[https://moodle.org/plugins/mod\\_videotime](https://moodle.org/plugins/mod_videotime) [Online; Acedido: Mai 22, 2020]
- [65] *Plugin* do tipo “Módulo de Atividades e Recursos” (“*Mod: H5P*”)  
[https://moodle.org/plugins/mod\\_hvp](https://moodle.org/plugins/mod_hvp) [Online; Acedido: Mar 05, 2020]
- [66] *Plugin* do tipo “Módulo de Atividades e Recursos” (“*Mod: Attendance*”)  
[https://moodle.org/plugins/mod\\_attendance](https://moodle.org/plugins/mod_attendance) [Online; Acedido: Jun 18, 2020]
- [67] Desbloquear cursos com o “Level up! - Gamification” <https://levelup.plus/docs/article/unlock-course-at-level> [Online; Acedido: Abr 30, 2020]
- [68] *Plugin* do tipo “Filtro” (“*Filter: Shortcodes*”)  
[https://moodle.org/plugins/filter\\_shortcodes](https://moodle.org/plugins/filter_shortcodes) [Online; Acedido: Jul 3, 2020]
- [69] Política de privacidade do Moodle  
[https://docs.moodle.org/39/en/Privacy\\_laws\\_and\\_Moodle](https://docs.moodle.org/39/en/Privacy_laws_and_Moodle) [Online; Acedido: Mai 14, 2020]
- [70] Regulamento Geral sobre a Proteção de Dados do Moodle  
<https://docs.moodle.org/39/en/GDPR> [Online; Acedido: Mai 16, 2020]
- [71] Documentação para *developers* do Vimeo  
<https://developer.vimeo.com/player/sdk> [Online; Acedido: Mai 24, 2020]
- [72] Documentação para *developers* do Vimeo  
<https://github.com/vimeo/player-api/tree/archived> [Online; Acedido: Mai 24, 2020]
- [73] Utilizar *web services* para criar *tokens* - Moodle  
[https://docs.moodle.org/39/en/Using\\_web\\_services#Create\\_a\\_token](https://docs.moodle.org/39/en/Using_web_services#Create_a_token) [Online; Acedido: Jul 23, 2020]
- [74] *Plugin* do tipo “Bloco” (“*Blocks: SMS*”)  
[https://moodle.org/plugins/block\\_sms](https://moodle.org/plugins/block_sms) [Online; Acedido: Mai 8, 2020]
- [75] API para envio de SMSs - Nexmo  
<https://developer.nexmo.com/api/sms> [Online; Acedido: Jul 7, 2020]
- [76] API para envio de SMSs - Twilio  
<https://www.twilio.com/docs/sms/send-messages> [Online; Acedido: Jul 7, 2020]

- [77] API para envio de SMSs - Clickatell  
<https://www.clickatell.com/products/sms/> [Online; Acedido: Jul 7, 2020]
- [78] Funcionamento da API do Twilio  
<https://www.twilio.com/learn/twilio-101/what-is-twilio> [Online; Acedido: Jul 10, 2020]
- [79] Como se processa o envio de SMSs através do Twilio  
<https://www.twilio.com/docs/sms/send-messages> [Online; Acedido: Jul 10, 2020]
- [80] *Plugin “HTML” do tipo “Bloco”*  
[https://docs.moodle.org/39/en/HTML\\_block](https://docs.moodle.org/39/en/HTML_block) [Online; Acedido: Fev 26, 2020]
- [81] 1059-1993 - *IEEE Guide for Software Verification and Validation Plans*. IEEE
- [82] Cache do Moodle  
<https://docs.moodle.org/39/en/Caching> [Online; Acedido: Mar 18, 2020]

*Esta página foi intencionalmente deixada em branco.*

## A Módulos de Atividades e Recursos do Moodle

### • Módulos de Atividades

As Atividades do Moodle representam itens dinâmicos predefinidos, de vários tipos, com os quais os alunos interagem.

Para além das Atividades predefinidas, o administrador pode instalar novos tipos a ser adicionados nos tópicos centrais das disciplinas.

#### – Base de Dados

O Módulo de Atividade “Base de Dados” possibilita aos seus participantes criar, gerir e procurar informação num conjunto de entradas (registos). A estrutura das entradas é definida pelos próprios professores, que determinam o número de campos a preencher, sendo que os tipos de campos permitidos incluem: caixas de seleção, listas pendentes, botões de rádio, caixas de texto, URL, imagens e ficheiros.

A estrutura de visualização da informação quando se cria, edita ou lista os registos da “Base de Dados” pode ser definida através de modelos da própria “Base de Dados”.

As “Base de Dados” podem ser partilhadas entre disciplinas e o professor também pode importar e exportar os seus registos.

Se a opção de hiperligação dos registos estiver ativa, é possível criar hiperligações entre os termos inseridos e essas palavras, de forma automática, sempre que apareçam noutros locais da disciplina.

Um professor pode permitir comentários aos registos, podendo estes ser avaliados por professores ou alunos (avaliação por pares), originando assim uma nota final, registada na pauta.

O Módulo de Atividade “Base de Dados” é então usada para, por exemplo:

1. Criar um repositório colaborativo de hiperligações/livros/referências.
2. Exibir trabalhos realizados pelos alunos e disponibilizá-los (permitindo comentários).

#### – Chat

O “Chat” permite que os participantes tenham discussões síncronas e em tempo real, sob a forma de texto.

O Módulo de Atividade “Chat” pode ter lugar uma única vez, como pode ser repetida em qualquer horário pretendido. As sessões são guardadas, podendo, ou ser disponibilizadas a todos os utilizadores, ou restritas a um determinado número/tipo de utilizador.

Revelam-se úteis quando se pretende:

1. Realizar sessões de perguntas e respostas com um convidado.
2. Realizar sessões para auxiliar os alunos a se prepararem para testes.
3. Realizar reuniões regulares para partilhar experiências.

#### – Ferramenta LTI/ Ferramenta Externa

O Módulo de Atividade “Ferramenta *Learning Tools Interoperability* - Interoperabilidade de Ferramentas de Aprendizagem (LTI)” Permite aos alunos interagir com recursos de aprendizagem que se encontrem alojados noutros sites, proporcionando

acesso a um novo tipo de Atividades, ou materiais de aprendizagem.

Para criar uma Atividade deste tipo, é necessário um fornecedor (de ferramentas) que o suportem.

Uma “Ferramenta LTI” tanto pode ser criada pelo professor, como configurada pelo administrador da plataforma (à qual o professor também teria acesso).

As “Ferramentas LTI” revelam-se distintas dos recursos URL nos seguintes aspetos:

1. São sensíveis ao contexto, tendo acesso, portanto, a informações sobre o utilizador que a emprega (como o seu nome, da instituição a que pertence ou das disciplinas às quais está associado);
2. Permitem ler, atualizar e apagar notas associadas à Atividade;
3. As configurações das “Ferramenta LTI” criam ligações confiáveis entre o site e o fornecedor da ferramenta, permitindo uma comunicação segura entre eles.

São, portanto, úteis para:

1. Oferecer uma maneira de os professores se vincularem a Atividades de aprendizagem externas na sua página do curso no Moodle e, quando disponíveis, ter as notas sejam enviadas de volta ao Moodle.

#### – Fórum

O “Fórum” permitem que os participantes tenham discussões assíncronas por um período prolongado, podendo adicionar ficheiros e imagens às mesmas. Caso os participantes se encontrem inscritos no “Fórum”, receberão notificações de novas publicações no mesmo. Podem ser criados:

1. Fóruns padrão, onde uma pessoa pode iniciar uma nova discussão a qualquer momento;
2. Fóruns onde cada aluno publica uma própria discussão;
3. Fóruns de perguntas e respostas, onde os alunos, antes de ver a publicação de outros alunos, devem, eles próprios, fazer uma publicação.

Um professor pode definir o modo de inscrição como opcional, automático, ou impedir a inscrição completamente.

Se necessário, os alunos podem ser impedidos de publicar mais do que um determinado número de publicações por um dado período, impedindo que certos utilizadores dominem as discussões.

As publicações no “Fórum” podem ser classificadas por professores ou alunos (avaliação por pares).

As classificações podem ser agregadas para formar uma nota final, registada no livro de notas.

Assim, os fóruns têm muitas utilizações, como:

1. Servir de espaço social para os alunos se conhecerem;
2. Anunciar cursos;
3. Discutir o conteúdo de um curso ou de algum material;
4. Retomar temas levantados em sessões presenciais;

5. Permitir a realização de discussões apenas com professores (através de um “Fórum” oculto);
6. Servir de centro de ajuda onde tanto professores, como alunos podem dar conselhos;
7. Servir de área de suporte individual com comunicações particulares entre alunos e professores.

#### – Glossário

O Módulo de Atividade “Glossário” funciona de forma semelhante a um dicionário, permitindo que os participantes criem uma lista de termos e respectivos significados.

Um professor pode permitir que determinados arquivos sejam anexados aos termos e, passando esses anexos por imagens, serão exibidas junto à definição do termo.

Estes termos poderão ser visualizados por ordem alfabética, por data, categoria ou autor, podendo ficar automaticamente visíveis a todos após serem inseridos pelos alunos, ou após a aprovação do professor.

Se a opção de hiperligações for ativada no “Glossário”, sempre que os termos surgirem em palavras e/ou frases nos conteúdos da disciplina, serão automaticamente convertidos em hiperligações ao respetivo termo e descrição no “Glossário”.

O professor também pode permitir que sejam adicionados comentários aos termos e que estes sejam avaliados pelos professores ou pelos alunos.

Os “Glossários” têm muitos usos, como:

1. Servir de base colaborativa de palavras-chave;
2. Servir de espaço de identificação pessoal, onde os novos alunos podem adicionar o seu nome e dados pessoais;
3. Servir de recurso de consulta sobre um assunto da disciplina;
4. Servir de área de partilha de vídeos, imagens ou ficheiros de som;
5. Conjunto de conceitos a serem revistos e lembrados.

#### – Inquérito

Os Módulos de Atividades “Inquérito” permitem que um professor crie um “Inquérito” personalizado para obter *feedback* dos participantes, usando uma variedade de tipos de questões, incluindo escolha múltipla, respostas de “sim/não” ou de resposta curta

As respostas ao “Inquérito” podem ser anónimas, se desejado, e os resultados podem ser exibidos a todos os participantes, ou restritos apenas aos professores.

Quaisquer Atividades de “Inquérito” que fiquem à disposição na página principal do site (*landing page*), também podem ser realizadas e concluídas por utilizadores não autenticados.

Os Inquéritos podem então ser utilizados para:

1. Fazer avaliações de disciplinas, de modo a melhorar os conteúdos para futuros participantes;
2. Permitir que os participantes se inscrevam nos módulos da disciplina, por exemplo;

3. Produzir Inquéritos configuráveis, a fim de, por exemplo, recolher informação dos visitantes sobre escolhas de disciplinas ou políticas escolares, assim como Inquéritos *anti-bullying* nos quais os alunos podem relatar incidentes anonimamente, por exemplo.

– **Inquérito Predefinido**

A presente Atividade possibilita a inserção de diferentes tipos de “Inquéritos Predefinidos”, que sirvam para avaliar e estimular a aprendizagem em ambientes web.

Os professores podem utilizá-los para recolher dados que poderão ajudá-lo a conhecer a turma e a refletir sobre o seu próprio método de ensino.

– **Lição**

Uma “Lição” permite que o professor forneça conteúdo e/ou pratique Atividades de maneira interessante e flexível.

A presente Atividade é constituída por uma série de páginas com conteúdo, em que cada uma pode terminar com uma pergunta.

Os tipos de pergunta podem ser de escolha múltipla, correspondência ou resposta curta. Dependendo da escolha correta ou não do aluno, este avança para a página seguinte ou retrocede para a anterior.

Uma “Lição” pode ser avaliada, tendo, portanto, pontuação, ou ser apenas de treino, sem pontuação.

Deste modo, uma “Lição” pode ser usada para:

1. Possibilitar a aprendizagem de conteúdos específicos;
2. Criar simulações e exercícios de tomada de decisão;
3. Proporcionar conteúdos para diferentes estilos de aprendizagem (por exemplo, os alunos podem optar, de acordo com a sua preferência, por uma página de vídeo, por um *podcast* ou por texto);
4. Rever conhecimentos adaptados ao nível de conhecimento de cada aluno, com diferentes conjuntos de perguntas (perguntas estas que serão exibidas em função das respostas dadas às perguntas iniciais).

– **Pacote *Sharable Content Object Reference Model* - Modelo de Referência de Objeto de Conteúdo Compartilhável (SCORM)**

Um “Pacote SCORM” representa um conjunto de ficheiros agregados em ficheiros *zip* sob as normas padrão para objetos de aprendizagem.

Os Módulos de Atividades “Pacote SCORM” permite carregar esses mesmos pacotes (ou *Aviation Industry Computer-Based Training Committee* - Comité de Treino por Computador da Indústria da Aviação (AICC)) numa disciplina.

O conteúdo é normalmente distribuído por várias páginas, com navegação entre as mesmas, existindo várias opções de visualização como numa janela *pop-up* (com índice, botões de navegação, etc.).

Os “Pacotes SCORM” geralmente incluem perguntas, em que as notas são registadas no relatório de avaliação da disciplina.

Os Módulos de Atividades “Pacote SCORM” podem ser usados:



1. Para apresentar conteúdo e animações multimédia;
2. Como ferramenta de avaliação.

#### – **Sondagem**

Os Módulos de Atividades “Sondagem” permite ao professor colocar uma pergunta, especificando múltiplas opções de resposta.

Os resultados da “Sondagem” podem ser publicados logo após os alunos responderem, após uma data específica, ou ficarem apenas visíveis para o professor, sendo que os resultados podem ser publicados juntamente com os nomes dos alunos ou de forma anónima.

A “Sondagem” pode ser utilizada:

1. Como uma pergunta rápida para estimular o pensamento sobre um tópico, testando assim rapidamente a compreensão dos alunos;
2. Para facilitar a tomada de decisões, por exemplo, permitindo que os alunos votem em algo.

#### – **Teste/Quiz**

Os Módulos de Atividades “Teste” permite aos professores criar testes com diversos tipos de perguntas, tais como: perguntas de escolha múltipla, de verdadeiro/falso, de correspondência, de resposta curta e calculadas.

O professor pode configurar o “Teste” para múltiplas tentativas, de maneira a poder ser realizado várias vezes e com perguntas aleatórias, ou com limite de tempo para a sua realização (benéfico caso se tencionasse adicionar o recurso de gamificação “Desafio” - capítulo 2.6.1).

Cada tentativa é automaticamente avaliada e registada na pauta da disciplina. O professor pode escolher se pretende dar *feedback* e/ou mostrar as respostas corretas ao aluno durante a tentativa, apenas após terminar a tentativa ou após o “Teste” fechar.

O professor pode escolher entre uma variedade de comportamentos das perguntas, incluindo o modo adaptável (facultando ajudas antes do aluno responder de novo) e *feedback* imediato baseado no grau de certeza (onde o aluno indica qual o seu grau de certeza naquela resposta).

Os “Testes” podem ser utilizados:

1. Como mini-testes de compreensão de um tema;
2. Para realizar um treino para um exame;
3. Para dar *feedback* sobre o desempenho do aluno;
4. Para auto avaliação.

#### – **Trabalho**

No Módulo de Atividade “Trabalho”, os alunos podem submeter ficheiros, tais como documentos *word*, folhas de cálculo, imagens, sons e vídeos. Em alternativa, ou como complemento, o “Trabalho” pode incluir a submissão de um texto usando o próprio editor de texto da plataforma. Esta Atividade pode também servir para lembrar os alunos de Atividades presenciais, como exames ou trabalhos, não sendo neste caso necessário submeter nenhum ficheiro ou texto. Os alunos podem submeter

o “Trabalho” individualmente ou em grupo.

Ao avaliar os “Trabalhos”, os professores podem inserir comentários de *feedback* e submeter ficheiros, nomeadamente o ficheiro de trabalho do aluno com comentários ou um ficheiro de áudio com o *feedback*. Os “Trabalhos” podem ser avaliados utilizando uma escala numérica, uma escala personalizada ou um critério avançado de avaliação, como as grelhas qualitativas. As notas dos alunos são registadas na pauta da disciplina.

A Atividade “Trabalho” permite ao professor:

1. Definir tarefas.
2. Disponibilizar o enunciado de um trabalho.
3. Recolher submissões dos alunos e rever, avaliar e dar feedback.
4. Lembrar os alunos de Atividades presenciais, como exames ou trabalhos, não sendo neste caso necessário submeter nenhum ficheiro ou texto.

– **Wiki**

A Atividade “Wiki” permite que os utilizadores adicionem e editem uma série de páginas com conteúdos, podendo estas ser interligadas. Um “Wiki” pode ser colaborativo (onde todos podem editar), ou individual (onde cada utilizador tem o seu próprio “Wiki”, onde só ele o pode editar).

É guardado um histórico das diferentes versões de cada página, listando as alterações feitas por cada utilizador.

Os “Wikis” podem ser utilizados:

1. Para notas de aulas de grupo ou planos de estudo;
2. Para os professores planearem o sumário do trabalho ou de uma reunião a realizar;
3. Para os alunos criarem de forma colaborativa um livro *online*, criando conteúdos sobre um tema definido pelo professor;
4. Para criar histórias ou poemas, de forma colaborativa, onde cada aluno escreve uma linha ou um verso; Como um diário pessoal para inserir notas sobre a matéria (“Wiki” individual).

– **Workshop**

O Módulo de Atividade “Workshop” permite inserir, analisar e avaliar o trabalho dos alunos, avaliação esta que é feita pelos colegas. Os alunos podem submeter qualquer conteúdo, como documentos *excel* e *word* ou escrever o texto diretamente no editor do Moodle. Os trabalhos são avaliados através de uma grelha de avaliação com vários critérios definidos pelo professor. Este pode disponibilizar trabalhos exemplo e uma avaliação de referência para que os alunos possam praticar o processo de avaliação e compreender a grelha de avaliação. Os alunos têm a oportunidade de avaliar um ou mais trabalhos dos seus colegas. Os autores dos trabalhos atribuídos para avaliação e os respetivos avaliadores pode ser mantidos anónimos. Os alunos obtêm duas notas no “Workshop” - uma pelo seu trabalho e outra pela avaliação que fez ao colega. Ambas as notas são registadas no relatório de avaliação da disciplina.

- **Recursos:**

Conteúdo estático que é utilizado pelo professor para dar suporte aos alunos.

- **Ficheiro**

O presente Recurso permite que um professor insira um “Ficheiro” como um recurso de uma disciplina. Sempre que o navegador o permitir, o “Ficheiro” será exibido dentro da interface da disciplina, caso contrário os alunos poderão descarregar o “Ficheiro”.

Este Recurso pode incluir ficheiros de suporte, como, por exemplo, uma página HTML pode ter imagens incorporadas.

É necessário que os alunos tenham os devidos programas instalados nos seus computadores para poderem visualizar os ficheiros.

Um “Ficheiro” pode ser usado para:

1. Partilhar apresentações com a turma;
2. Incluir um mini site como um recurso da disciplina;
3. Disponibilizar ficheiros de rascunho de certos programas para que os alunos possam editar e submeter para avaliação.

- **Livro**

O Recurso “Livro” permite que o professor crie um conteúdo com várias páginas com formato idêntico a um livro, com capítulos e sub capítulos.

Assim, um “Livro” pode ser usado para:

1. Disponibilizar material para leitura nas diferentes unidades de aprendizagem;
2. Disponibilizar um manual com várias páginas;
3. Disponibilizar um portefólio com trabalhos do aluno.

- **Pacote *Instructional Management System* - Sistema de Gestão Instrucional (IMS)**

Um “Pacote IMS” é um conjunto de ficheiros que são compactados de acordo com uma norma estipulada, para que possam ser reutilizados em diferentes plataformas.

O Recurso “Pacote IMS” permite que pacotes criados externamente ao Moodle (formato *zip*) possam ser adicionados a uma disciplina.

O conteúdo é geralmente exibido em várias páginas com navegação entre as mesmas.

Existem várias opções para a exibição do conteúdo, como numa janela *pop-up*, com menu de navegação, botões, etc.

Um Pacote IMS pode ser usado para apresentar conteúdos multimédia e animações.

- **Página** O Recurso “Página” permite criar uma página web através do editor *html*, podendo a respetiva página exibir texto, imagens, som, vídeo, hiperligações e código *html* incorporado (como o *Google Maps*, por exemplo).

As vantagens ao utilizar este Recurso ao invés do Recurso “Ficheiro” é que a “Página” pode ser mais fácil de aceder para, por exemplo, utilizadores em dispositivos móveis, sendo também mais fácil de atualizar.

Assim, uma página pode ser utilizada para:

1. Apresentar os termos e condições ou o programa da disciplina;
2. Incorporar vários vídeos ou ficheiros de som, juntamente com um texto explicativo.

– **Pasta**

O Recurso “Pasta” permite que o professor disponibilize vários ficheiros apenas com uma hiperligação, evitando assim o excesso de ficheiros na página principal da disciplina.

Uma pasta pode ser criada, carregando-se ficheiro a ficheiro, ou carregando um único *zip* com vários ficheiros e extrair os mesmos dentro da pasta.

A “Pasta” pode ser utilizada para:

1. Disponibilizar uma série de ficheiros num tópico (a bibliografia necessária para um exame em formato *pdf*, por exemplo).
2. Partilhar ficheiros entre os professores da disciplina (numa pasta oculta à qual os alunos não têm acesso).

– **Separador**

Este módulo possibilita que, tanto texto, como conteúdo multimédia, sejam inseridos na página do curso entre os links para outros Recursos e Atividades. São Recursos extremamente versáteis que, quando usadas premeditadamente, podem ajudar a melhorar a própria aparência de um curso.

Assim, a título de exemplo, podem ser utilizadas para:

1. Dividir uma longa lista de Atividades com um subtítulo ou uma imagem;
2. Exibir um ficheiro de som ou vídeo incorporados diretamente na página principal da disciplina;
3. Adicionar uma breve descrição a um tópico.

– **URL**

Permite ao professor disponibilizar uma hiperligação para um conteúdo externo na página principal da disciplina. Existe uma série de opções de visualização do URL: pode ser incorporado; surgir numa nova janela; mediante opções avançadas, pode passar informações como o nome do aluno, se necessário.

## B Ficheiros Predefinidos Para a Criação de um *Plugin* de Tipo Bloco

### Ficheiro `blocks/nomedoplugin/db/install.xml`

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XMLDB PATH="blocks/nomedoplugin/db" VERSION="20200319" COMMENT="XMLDB file for
  Moodle blocks/nomedoplugin">
3 <TABLES>
4 <TABLE NAME="nomedatabela" COMMENT="Default comment for block_nomedoplugin,
  please edit me">
5 <FIELDS>
6 <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"
  />
7 <FIELD NAME="user_id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="
  false" />
8 ...
9 </FIELDS>
10 <KEYS>
11 <KEY NAME="primary" TYPE="primary" FIELDS="id" />
12 </KEYS>
13 </TABLE>
14 </TABLES>
15 </XMLDB>

```

### Ficheiro `blocks/nomedoplugin/db/access.php`

```

1 <?php
2 defined('MOODLE_INTERNAL') || die();
3
4 $capabilities = array(
5
6     'block/nomedoplugin:myaddinstance' => array(
7         'captype' => 'write',
8         'contextlevel' => CONTEXT_SYSTEM,
9         'archetypes' => array(
10            'user' => CAP_ALLOW
11        ),
12
13        'clonepermissionsfrom' => 'moodle/my:manageblocks'
14    ),
15
16    'block/nomedoplugin:addinstance' => array(
17        'riskbitmask' => RISK_SPAM | RISK_XSS,
18
19        'captype' => 'write',
20        'contextlevel' => CONTEXT_BLOCK,
21        'archetypes' => array(
22            'editingteacher' => CAP_ALLOW,
23            'manager' => CAP_ALLOW
24        ),
25
26        'clonepermissionsfrom' => 'moodle/site:manageblocks'
27    ),
28 );

```

**Ficheiro blocks/nomedoplugin/lang/pt/block\_nomedoplugin.php**

```
1 <?php
2 $string['pluginname'] = 'Nome do plugin';
3 $string['title'] = 'Titulo do plugin';
4 ...
```

**Ficheiro blocks/nomedoplugin/version.php**

```
1 <?php
2 defined('MOODLEINTERNAL') || die();
3
4 $plugin->version = 2020070301; // The current plugin version (Date:
   YYYYMMDDXX)
5 $plugin->requires = 2019111200; // Requires this Moodle version
6 $plugin->component = 'block_nomedoplugin'; // Full name of the plugin (used for
   diagnostics)
7 $plugin->maturity = MATURITY_STABLE; //
```

**Ficheiro blocks/nomedoplugin/block\_nomedoplugin.php**

```
1 <?php
2 defined('MOODLEINTERNAL') || die();
3
4
5 class block_nomedoplugin extends block_base {
6
7     // Initialize the plugin
8     public function init() {
9         $this->title = get_string('pluginname', 'block_nomedoplugin');
10    }
11
12    // Block's contents
13    public function get_content() {
14
15        if ($this->content !== NULL) {
16            return $this->content;
17        }
18
19        $this->content = new stdClass;
20        ...
21    }
22 }
```

## C Estrutura e Apresentação dos *Plugins* Criados e Adicionados

- *Plugin “Daily Mood”* (block\_daily\_mood)

```
blocks
├── daily_mood
│   ├── db
│   │   ├── access.php
│   │   ├── install.xml
│   │   ├── mobile.php
│   │   └── services.php
│   ├── pix
│   │   └── termometer.png
│   ├── classes
│   │   ├── event
│   │   │   └── humor_set.php
│   │   ├── output
│   │   │   └── mobile.php
│   ├── templates
│   │   ├── daily_mood_set.mustache
│   │   └── mobile_daily_mood.mustache
│   ├── lang
│   │   ├── en
│   │   │   └── block_daily_mood.php
│   │   └── pt
│   │       └── block_daily_mood.php
│   ├── block_daily_mood.php
│   ├── common.php
│   ├── externallib.php
│   ├── version.php
│   ├── simplehtml_form.php
│   ├── slider.php
│   └── styles.css
```

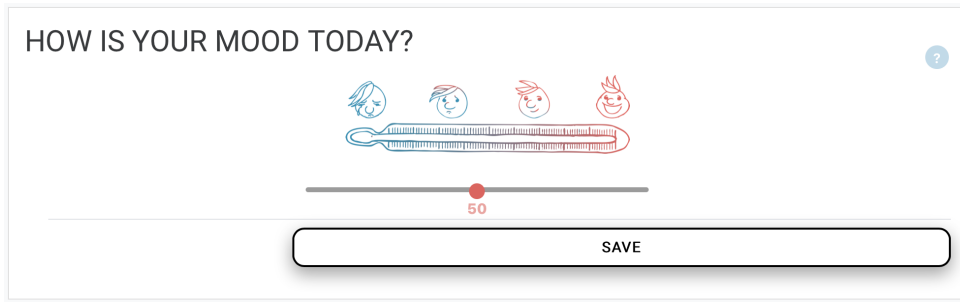


Figura C1: *Plugin* “Diário do Humor” (block\_daily\_activities) no perfil do aluno - versão web

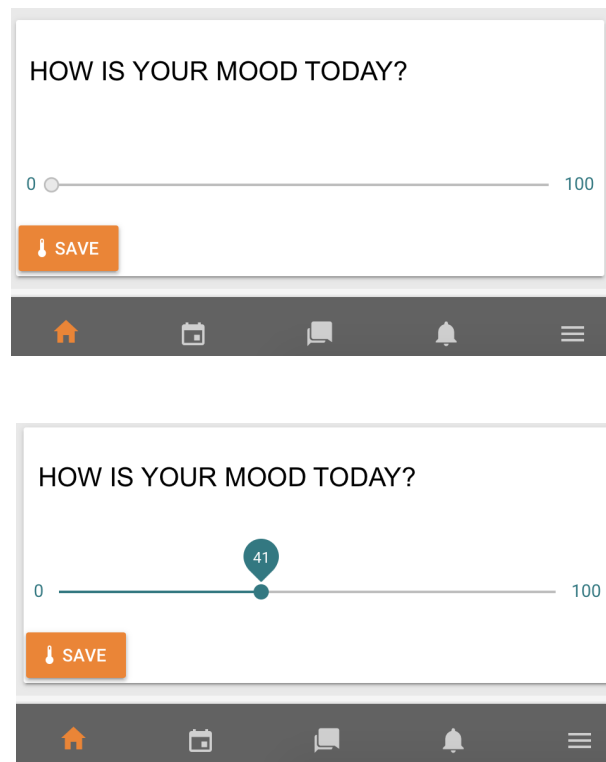


Figura C2: *Plugin* “Diário do Humor” (block\_daily\_activities) no perfil do aluno - versão *mobile*



• **Plugin “Daily Activities” (block\_daily\_activities) - primeira versão**

```
blocks
├── daily_activities
│   ├── db
│   │   ├── access.php
│   │   └── install.xml
│   ├── pix
│   │   └── Daily_Activities_Image.png
│   ├── lang
│   │   ├── en
│   │   │   └── block_daily_activities.php
│   │   └── pt
│   │       └── block_daily_activities.php
│   ├── block_daily_activities.php
│   ├── activities.js
│   ├── version.php
│   ├── simple_html_form.php
│   └── styles.css
```

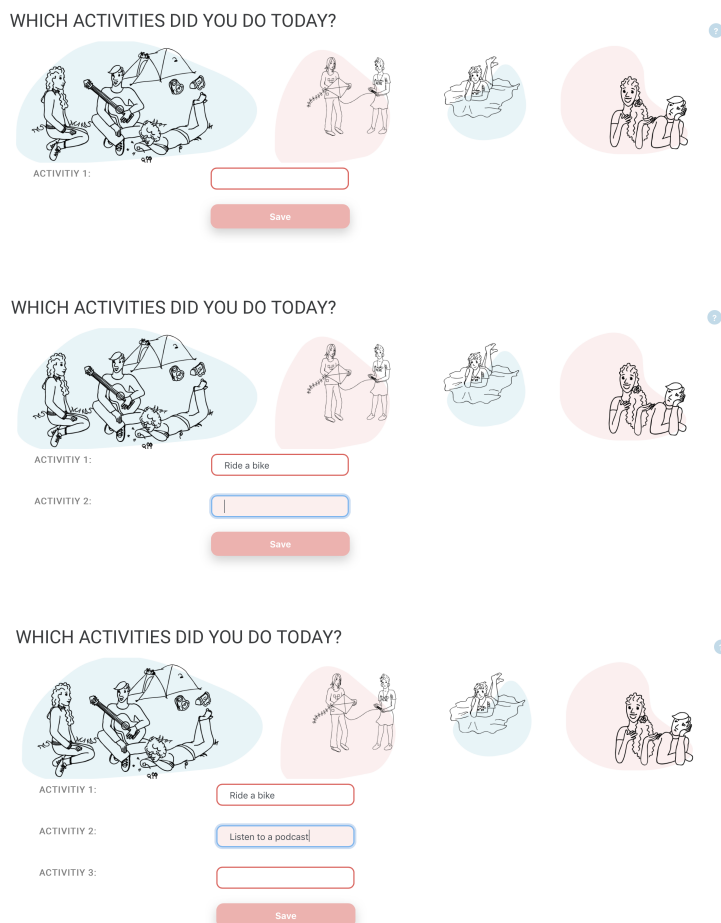


Figura C3: Apresentação do Bloco “Diário de Atividades” - Primeira Versão

• *Plugin “Daily Activities” (block\_daily\_activities) - última versão*

```
blocks
├── daily_activities
│   ├── db
│   │   ├── access.php
│   │   ├── install.xml
│   │   ├── mobile.php
│   │   └── services.php
│   ├── pix
│   │   └── Daily_Activities_Image.png
│   ├── classes
│   │   ├── event
│   │   │   └── activity_set.php
│   │   ├── output
│   │   │   └── mobile.php
│   ├── templates
│   │   └── daily_activities.mustache
│   ├── js
│   │   └── daily_activities.js
│   ├── lang
│   │   ├── en
│   │   │   └── block_daily_activities.php
│   │   └── pt
│   │       └── block_daily_activities.php
│   ├── block_daily_activities.php
│   ├── common.php
│   ├── externallib.php
│   ├── version.php
│   ├── table_structure.php
│   ├── slider.php
│   └── styles.css
```

WHICH ACTIVITIES DID YOU DO TODAY?

List of Activities	Seg, 24 Aug	Ter, 25 Aug	Qua, 26 Aug	Qui, 27 Aug	Sex, 28 Aug	Sáb, 29 Aug	Dom, 30 Aug
<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Number of Activities	1	0	0	0	0	0	0

Figura C4: Versão Web - Apresentação do Bloco “Diário de Atividades” no Perfil do Aluno

New Site

WHICH ACTIVITIES DID YOU DO TODAY?

PREVIOUS NEXT

2 Sep

Write new activity

Write new activity

Write new activity

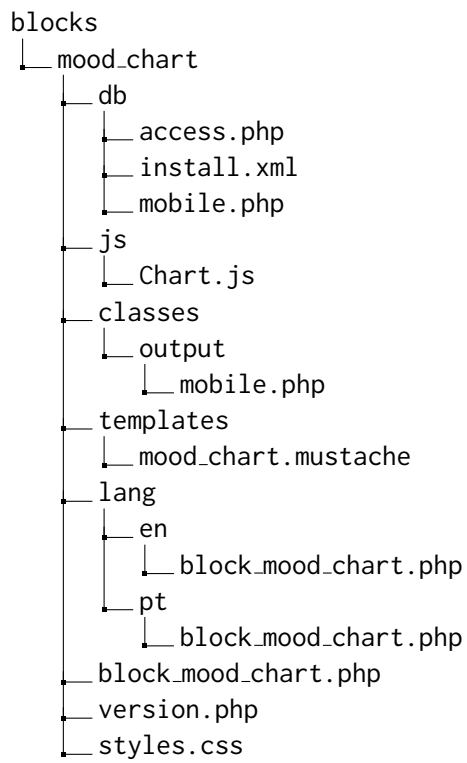
Write new activity

Write new activity

SAVE

Figura C5: Versão Mobile - Apresentação do Bloco “Diário de Atividades” no Perfil do Aluno

• *Plugin “Mood Chart” (block\_mood\_chart)*



MOOD AND ACTIVITIES CHART

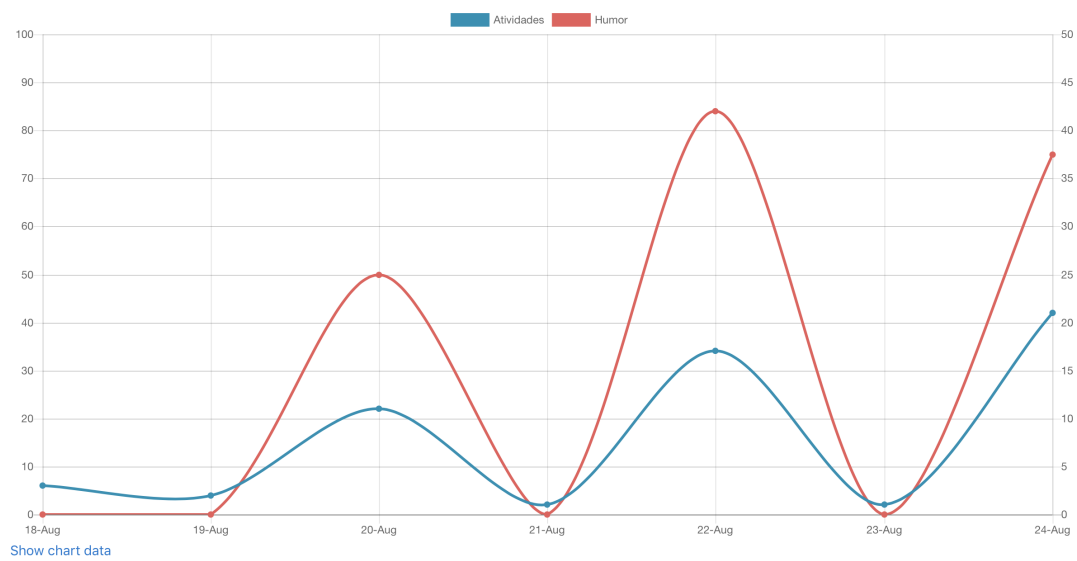


Figura C6: Apresentação do Bloco “Gráfico de Humor e Atividades”

- **Plugin “Level up! - Gamification” (block\_xp)**

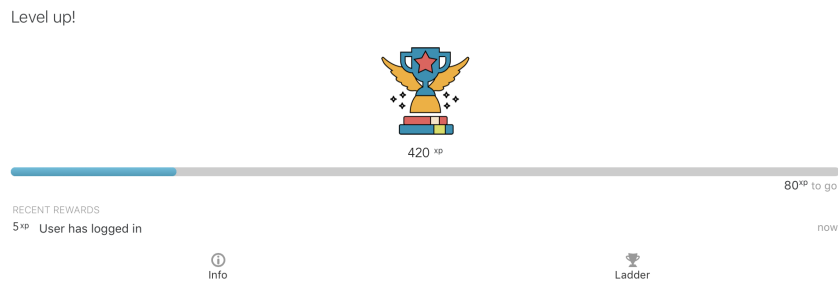


Figura C7: *Plugin “Level up! - Gamification” (block\_xp)* - apresentação do *plugin* na página inicial do aluno após a customização efetuada (versão web)



Figura C8: *Plugin “Level up! - Gamification” (block\_xp)* - apresentação da tabela de crachás do *plugin* na página inicial do aluno após a customização efetuada (versão web)

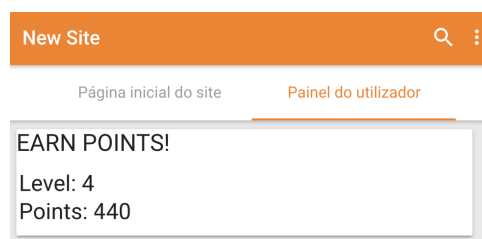


Figura C9: *Plugin “Level up! - Gamification” (block\_xp)* - apresentação na página inicial do aluno (versão *mobile*)

*Esta página foi intencionalmente deixada em branco.*

## D *Plugin “Level up! - Gamification” (block/xp) - Lista de Eventos*

Tabela D1: Eventos do *plugin “Level up! - Gamification” (block/xp) - contexto de “Sistema”*

Contexto	Descrição do Evento	Nome do Evento
Sistema	1. Disciplina visualizada.	\core\event\course_viewed
	<b>2. Login</b>	\core\event\user_loggedin
	<b>3. Humor definido</b>	\block_daily_mood\event\humor_set
	<b>4. Atividades definidas</b>	\block_daily_activities\event\activity_set

Tabela D2: Eventos do *plugin “Level up! - Gamification” (block/xp) - contexto de contexto de “Módulo de Atividades e Recursos” (Atividades)*

Contexto	Descrição do Evento	Nome do Evento
<i>Assignment</i> / Trabalho	<b>1. Enviada submissão de um trabalho</b>	\mod_assign\event\assessable_submitted
	2. Módulo de disciplina visualizado	\mod_assign\event\course_module_viewed
	3. Trabalho visualizado	\mod_assign\event\feedback_viewed
	4. Submissão visualizada	\mod_assign\event\submission_viewed
	5. O utilizador duplicou a sua submissão	\mod_assign\event\submission_duplicated
Livro	1. Capítulo visualizado	\mod_book\event\chapter_viewed
	2. Módulo de disciplina visualizado	\mod_book\event\course_module_viewed
<i>Chat</i>	1. Mensagem enviada	\mod_chat\event\message_sent
	2. Módulo de disciplina visualizado	\mod_chat\event\course_module_viewed
<i>Choice/ Sondagem</i>	1. Módulo de disciplina visualizado	\mod_choice\event\course_module_viewed
	2. Resposta à sondagem eliminada	\mod_choice\event\answer_deleted
	3. Resposta à sondagem gravada	\mod_choice\event\answer_created
	4. Sondagem feita (em desuso)	\mod_choice\event\answer_submitted
	5. Sondagem atualizada (em desuso)	\mod_choice\event\answer_updated

Contexto	Descrição do Evento	Nome do Evento
Base de Dados	1. Comentário apagado	\mod_data\event\comment_deleted
	2. Comentário criado	\mod_data\event\comment_created
	3. Módulo de disciplina visualizado	\mod_data\event\course_module_viewed
	4. Registo atualizado	\mod_data\event\record_updated
	5. Registo criado	\mod_data\event\record_created
	6. Registo eliminado	\mod_data\event\record_deleted
<i>Feedback</i> / Inquérito	1. Módulo de disciplina visualizado	\mod_feedback\event\course_module_viewed
	2. Resposta apagada	\mod_feedback\event\response_deleted
	<b>3. Resposta submetida</b>	<b>\mod_feedback\event\response_submitted</b>
Fórum	1. Módulo de disciplina visualizado	\mod_forum\event\course_module_viewed
	2. Disciplina procurada	\mod_forum\event\course_searched
	3. Discussão criada	\mod_forum\event\discussion_created
	4. Subscrição de discussão criada	\mod_forum\event\discussion_subscription_created
	5. Subscrição de discussão apagada	\mod_forum\event\discussion_subscription_deleted
	6. Discussão visualizada	\mod_forum\event\discussion_viewed
	7. <i>Post</i> criado	\mod_forum\event\post_created
	8. Foi publicado algum conteúdo	\mod_forum\event\assessable_uploaded
	9. Subscrição criada	\mod_forum\event\subscription_created
	10. Subscrição apagada	\mod_forum\event\subscription_deleted
	11. Relatório do utilizador visualizado	\mod_forum\event\user_report_viewed
Glossário	1. A categoria foi criada	\mod_glossary\event\category_created
	2. A categoria foi apagada	\mod_glossary\event\category_deleted
	3. A categoria foi atualizada	\mod_glossary\event\category_updated
	4. Comentário criado	\mod_glossary\event\comment_created
	5. Comentário apagado	\mod_glossary\event\comment_deleted
	6. Módulo de disciplina visualizado	\mod_glossary\event\course_module_viewed
	7. A entrada foi aprovada	\mod_glossary\event\entry_approved



Contexto	Descrição do Evento	Nome do Evento
Glossário	8. A entrada foi aprovada	\mod_glossary\event\entry_approved
	9. A entrada foi criada	\mod_glossary\event\entry_created
	10. A entrada foi apagada	\mod_glossary\event\entry_deleted
	11. A entrada não foi apagada	\mod_glossary\event\entry_disapproved
	12. A entrada foi atualizada	\mod_glossary\event\entry_updated
	13. A entrada foi visualizada	\mod_glossary\event\entry_viewed
<i>Lição</i>	1. Notas adicionadas	\mod_lesson\event\highscore_added
	2. Notas visualizadas	\mod_lesson\event\highscore_viewed
	3. Conteúdo da página visualizado	\mod_lesson\event\content_page_viewed
	4. Módulo de disciplina visualizado	\mod_lesson\event\course_module_viewed
	5. Lição terminou	\mod_lesson\event\lesson_ended
	6. Lição recomeçou	\mod_lesson\event\lesson_restarted
	7. Lição retomada	\mod_lesson\event\lesson_resumed
	8. Lição iniciada	\mod_lesson\event\lesson_started
	9. Questão respondida	\mod_lesson\event\question_answered
	10. Questão visualizada	\mod_lesson\event\question_viewed
Ferramenta Externa	1. Módulo de disciplina visualizado	\mod_lti\event\course_module_viewed
<i>Quiz/Teste</i>	1. Módulo de disciplina visualizado	\mod_quiz\event\course_module_viewed
	2. Tempo limite da tentativa de teste excedido.	\mod_quiz\event\attempt_becameoverdue
	3. Tentativa de teste abandonada.	\mod_quiz\event\attempt_abandoned
	4. Tentativa de teste iniciada	\mod_quiz\event\attempt_started
	<b>5. Tentativa de teste submetida</b>	<b>\mod_quiz\event\attempt_submitted</b>
	6. Tentativa de teste visualizada	\mod_quiz\event\attempt_viewed
Pacote SCORM	1. Estado do SCORM submetido	\mod_scorm\event\status_submitted
	2. Módulo de disciplina visualizado	\mod_scorm\event\course_module_viewed
	3. Nota bruta do SCORM submetida	\mod_scorm\event\scoreraw_submitted
	4. Sco executado	\mod_scorm\event\sco_launched

Contexto	Descrição do Evento	Nome do Evento
<i>Survey</i> / Inquérito Predefinido	1. Módulo de disciplina visualizado	\mod_survey\event\course_module_viewed
	<b>2. Resposta ao inquérito submetida</b>	<b>\mod_survey\event\response_submitted</b>
<i>Wiki</i>	1. Comentário apagado	\mod_wiki\event\comment_deleted
	2. Comentário criado	\mod_wiki\event\comment_created
	3. Comentários visualizados	\mod_wiki\event\comment_viewed
	4. Diferenças do <i>wiki</i> visualizadas	\mod_wiki\event\page_diff_viewed
	5. Histórico do <i>wiki</i> visualizado	\mod_wiki\event\page_history_viewed
	6. Mapa de páginas do <i>wiki</i> visualizado	\mod_wiki\event\page_map_viewed
	7. Módulo de disciplina visualizado	\mod_wiki\event\course_module_viewed
	8. Página do <i>wiki</i> atualizada	\mod_wiki\event\page_updated
	9. Página do <i>wiki</i> criada	\mod_wiki\event\page_created
	10. Página do <i>wiki</i> eliminada	\mod_wiki\event\page_deleted
	11. Página do <i>wiki</i> restaurada	\mod_wiki\event\page_version_restored
	12. Página do <i>wiki</i> visualizada	\mod_wiki\event\page_viewed
<i>Workshop</i>	1. Módulo de disciplina visualizado	\mod_workshop\event\course_module_viewed
	2. Submissão apagada	\mod_workshop\event\submission_deleted
	3. Submissão atualizada	\mod_workshop\event\submission_updated
	4. Submissão avaliada	\mod_workshop\event\submission_assessed
	5. Submissão criada	\mod_workshop\event\submission_created
	6. Submissão enviada	\mod_workshop\event\assessable_uploaded
	7. Submissão reavaliada	\mod_workshop\event\submission_reassessed
	8. Submissão visualizada	\mod_workshop\event\submission_viewed

Contexto	Descrição do Evento	Nome do Evento
<b>H5P</b>	<b>1. Módulo de disciplina visualizado</b>	<code>\mod_hvp\event\course_module_viewed</code>
	<b>2. Tentativa submetida</b>	<code>\mod_hvp\event\attempt_submitted</code>
	<b>3. Completude atualizada</b>	<code>\mod_hvp\event\course_completion_updated</code>
<b>VideoTime</b>	<b>1. Módulo de disciplina visualizado</b>	<code>\mod_videotime\event\course_module_viewed</code>
	<b>2. Completude atualizada</b>	<code>\mod_videotime\event\course_completion_updated</code>

Tabela D3: Eventos do *plugin* “*Level up! - Gamification*” (block/xp) - contexto de contexto de “Módulo de Atividades e Recursos” (Recursos)

Contexto	Descrição do Evento	Nome do Evento
Pasta	1. Foi descarregado um ficheiro ZIP da pasta	<code>\mod_folder\event\all_files_downloaded</code>
	2. Módulo de disciplina visualizado	<code>\mod_folder\event\course_module_viewed</code>
Pacote IMS	1. Módulo de disciplina visualizado	<code>\mod_ims\event\course_module_viewed</code>
Página	1. Módulo de disciplina visualizado	<code>\mod_page\event\course_module_viewed</code>
Ficheiro	1. Módulo de disciplina visualizado	<code>\mod_file\event\course_module_viewed</code>
	<b>2. Completude atualizada</b>	<code>\mod_file\event\course_completion_updated</code>
URL	1. Módulo de disciplina visualizado	<code>\mod_url\event\course_module_viewed</code>

Na listagem efetuada, todos os eventos a negrito representam os eventos utilizados. A preto estão os eventos já identificados pelo *plugin* block\_xp. A verde estão os novos eventos adicionados por *plugins* já existentes, adicionados no projeto. A azul estão os novos eventos criados e adicionados a *plugins* já existentes ou a novos *plugins* criados.

*Esta página foi intencionalmente deixada em branco.*

## E Tabelas de Testes Manuais Efetuados

### Testes Web

Tabela E1: Teste T\_01

<b>ID do Teste</b>	T_01
<b>Requisito Referente</b>	REQ_F_10.1
<b>Cenário do Teste</b>	Submeter Diário do Humor
<b>Passos do Teste</b>	<ol style="list-style-type: none"><li>1. O aluno encontra-se no seu perfil.</li><li>2. O aluno acede ao diário onde preenche diariamente o valor do seu humor sob a forma de um range slider.</li><li>3. O valor do humor varia entre zero e cem.</li><li>4. O aluno seleciona um valor.</li><li>5. O aluno submete o valor que selecionou.</li></ol>
<b>Dados do Teste</b>	humor = 30
<b>Resultados Esperados</b>	<ol style="list-style-type: none"><li>1. O aluno consegue submeter o valor selecionado.</li><li>2. O aluno deve ficar com o registo do valor que submeteu.</li><li>3. O aluno só tem possibilidade de realizar uma submissão (por dia).</li></ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E2: Teste T\_02

<b>ID do Teste</b>	T_02
<b>Requisito Referente</b>	REQ_F_10.2
<b>Cenário do Teste</b>	Submeter Diário de Atividades
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao diário onde preenche diariamente as atividades que realizou durante o dia.</li> <li>3. O aluno escreve na coluna das atividades, as que realizou durante o dia.</li> <li>4. O aluno seleciona com um <i>check</i>, na linha da atividade correspondente, as colunas dos dias que deseja (indicando que realizou a atividade nesse dia).</li> <li>5. O aluno submete o que preencheu.</li> </ol>
<b>Dados do Teste</b>	atividade1 = 'Listen to a podcast' atividade2 = 'Spend time on social media' atividade3 = '' atividade4 = 'Watch an episode of my favorite serie' atividade5 = 'Workout' id_day_of_week_0_0 <i>checked</i> id_day_of_week_0_2 <i>checked</i> id_day_of_week_1_0 <i>checked</i> id_day_of_week_1_1 <i>checked</i> id_day_of_week_1_2 <i>checked</i> id_day_of_week_1_3 <i>checked</i> id_day_of_week_1_4 <i>checked</i> id_day_of_week_1_5 <i>checked</i> id_day_of_week_1_6 <i>checked</i> id_day_of_week_1_7 <i>checked</i> id_day_of_week_2_0 <i>checked</i> id_day_of_week_3_6 <i>checked</i> id_day_of_week_4_1 <i>checked</i> id_day_of_week_4_3 <i>checked</i> id_day_of_week_4_6 <i>checked</i>
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno consegue submeter o que preencheu na tabela.</li> <li>2. Ao fazer <i>refresh</i> ao <i>browser</i>, os dados inseridos mantêm-se na tabela.</li> <li>3. Não se mantêm <i>checkadas checkboxes</i> em linhas onde nenhuma atividade tenha sido inserida na coluna das atividades</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E3: Teste T\_03

ID do Teste	T_03
Requisito Referente	REQ_F_10.2
Cenário do Teste	Adicionar Linha ao Diário de Atividades
Passos do Teste	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao diário onde preenche diariamente as atividades que realizou durante o dia.</li> <li>3. O aluno clica no botão “+”.</li> </ol>
Dados do Teste	Click no botão “+” da tabela.
Resultados Esperados	<ol style="list-style-type: none"> <li>1. É adicionada uma linha à tabela.</li> <li>2. O aluno consegue submeter dados inseridos na nova linha.</li> </ol>
Resultado Atual	Como esperado
Passou/Falhou	Passou

Tabela E4: Teste T\_04

ID do Teste	T_04
Requisito Referente	REQ_F_10.2
Cenário do Teste	Apagar Atividade no Diário de Atividades
Passos do Teste	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao diário onde preenche diariamente as atividades que realizou durante o dia.</li> <li>3. O aluno apaga uma das atividades da coluna de atividades.</li> <li>4. O aluno submete.</li> </ol>
Dados do Teste	atividade1 = ‘
Resultados Esperados	1. A atividade apagada não aparece mais na tabela, assim como as suas <i>checkboxes</i> que tinham sido <i>checkadas</i> .
Resultado Atual	Como esperado
Passou/Falhou	Passou

Tabela E5: Teste T\_05

<b>ID do Teste</b>	T_05
<b>Requisito Referente</b>	REQ_F_10
<b>Cenário do Teste</b>	Visualização de um Gráfico de Humor e Atividades
<b>Passos do Teste</b>	Passos realizados nos testes referentes aos Requisitos REQ_F_10.1. e REQ_F_10.2.
<b>Dados do Teste</b>	Dados de Teste introduzidos nos Requisitos REQ_F_10.1. e REQ_F_10.2.
<b>Resultados Esperados</b>	1. Os valores introduzidos no "Diário do Humor" (valor do humor) e no "Diário de Atividades" (número de atividades) aparecem no gráfico de linhas nos dias correspondentes.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E6: Teste T\_06

<b>ID do Teste</b>	T_06
<b>Requisito Referente</b>	REQ_F_15
<b>Cenário do Teste</b>	Redes Sociais
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O professor/psicólogo encontra-se no seu perfil.</li> <li>2. O perfil do professor/psicólogo fornece dois botões: um de partilha no <i>Facebook</i> e outro de partilha no <i>Twitter</i>.</li> <li>3. O professor/psicólogo clica no botão de partilha para o <i>Facebook</i> ou para o <i>Twitter</i>.</li> </ol>
<b>Dados do Teste</b>	Click no botão "Partilhar"
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. No caso de o professor/psicólogo não ter efetuado o <i>login</i> na sua rede social, é aberta uma nova página no <i>browser</i> para efetuar o <i>login</i> na respetiva rede. No caso de o professor/psicólogo já ter efetuado o respetivo <i>login</i>, a nova janela já possibilita a partilha diretamente.</li> <li>2. O professor/psicólogo consegue efetuar uma partilha da plataforma na respetiva rede social (ao clicar no botão do <i>Facebook</i> faz uma partilha no seu perfil do <i>Facebook</i>; ao clicar no botão, efetua uma partilha no seu perfil do <i>Twitter</i>).</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou



Tabela E7: Teste T\_07

<b>ID do Teste</b>	T_07
<b>Requisito Referente</b>	REQ_F_18
<b>Cenário do Teste</b>	Pontuação
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno consegue ver no seu perfil o local onde se encontra a sua pontuação.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E8: Teste T\_08

<b>ID do Teste</b>	T_08
<b>Requisito Referente</b>	REQ_F_18.1
<b>Cenário do Teste</b>	Ganhar 2 pontos - Preencher “jogo”
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil. 2. O aluno acede aos conteúdos a ele lhe destinados. 3. O aluno seleciona um “jogo”. 4. O aluno responde às perguntas e submete o “jogo”. 5. O aluno acede novamente ao seu perfil. 6. O aluno acede ao local onde se encontra a sua pontuação.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 14 pontos (dois pontos por sete perguntas).
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E9: Teste T\_09

<b>ID do Teste</b>	T_09
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Efetuar <i>login</i>
<b>Passos do Teste</b>	1. O aluno efetua <i>login</i> na plataforma e é redirecionado para o seu próprio perfil. 2. O aluno acede ao local onde se encontra a sua pontuação.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 5 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E10: Teste T\_10

<b>ID do Teste</b>	T_10
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Ver vídeo sem pergunta
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil. 2. O aluno acede aos conteúdos a ele lhe destinados. 3. O aluno seleciona um vídeo sem pergunta. 4. O aluno marca a tarefa como concluída. 5. O aluno acede novamente ao seu perfil. 6. O aluno acede ao local onde se encontra a sua pontuação.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno acede ao seu perfil. 2. O aluno acede ao local onde se encontra a sua pontuação. 3. A pontuação do aluno aumenta 5 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E11: Teste T\_11

<b>ID do Teste</b>	T_11
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Tarefa Diária (Diário do Humor)
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao “Diário do Humor”.</li> <li>3. O aluno seleciona o valor do humor desejado no <i>range slider</i> e submete.</li> <li>4. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno acede ao local onde se encontra a sua pontuação.</li> <li>2. A pontuação do aluno aumenta 5 pontos.</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E12: Teste T\_12

<b>ID do Teste</b>	T_12
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Tarefa Diária (Diário de Atividades)
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao “Diário de Atividades”.</li> <li>3. O aluno efetua o registo e submete.</li> <li>4. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno acede ao local onde se encontra a sua pontuação.</li> <li>2. A pontuação do aluno aumenta 5 pontos.</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E13: Teste T\_13

<b>ID do Teste</b>	T_13
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Ver Vídeo com Pergunta
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>3. O aluno seleciona um vídeo com pergunta.</li> <li>4. O aluno marca a tarefa como concluída.</li> <li>4. O aluno acede novamente ao seu perfil.</li> <li>5. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E14: Teste T\_14

<b>ID do Teste</b>	T_14
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Realizar <i>Quiz</i>
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno encontra-se no seu perfil.</li> <li>3. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>4. O aluno seleciona um <i>quiz</i>.</li> <li>5. O aluno preenche e submete o <i>quiz</i>.</li> <li>6. O aluno acede novamente ao seu perfil.</li> <li>7. O aluno acede ao local da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E15: Teste T\_15

<b>ID do Teste</b>	T_15
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Preencher Registo Semanal
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao local dos registos semanais.</li> <li>3. O aluno seleciona o registo semanal que tenciona preencher.</li> <li>4. O aluno preenche e submete o registo semanal</li> <li>5. O aluno acede novamente ao seu perfil.</li> <li>6. O aluno acede ao local da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E16: Teste T\_16

<b>ID do Teste</b>	T_16
<b>Requisito Referente</b>	REQ_F_18.4
<b>Cenário do Teste</b>	Ganhar 20 Pontos - Ouvir áudio
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>3. O aluno seleciona um áudio de meditação.</li> <li>4. O aluno marca a tarefa como concluída.</li> <li>5. O aluno acede ao seu perfil novamente.</li> <li>6. O aluno acede à zona da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 20 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E17: Teste T\_17

<b>ID do Teste</b>	T_17
<b>Requisito Referente</b>	REQ_F_18.4
<b>Cenário do Teste</b>	Ganhar 20 Pontos - Participar na Sessão Presencial
<b>Passos do Teste</b>	1.O professor/psicólogo encontra-se no seu perfil. 2. O professor/psicólogo cria uma sessão. 3. O professor/psicólogo assinala as presenças de uma sessão.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. Os alunos que se encontraram presentes (P) ou atrasados (L) recebem 20 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E18: Teste T\_18

<b>ID do Teste</b>	T_18
<b>Requisito Referente</b>	REQ_F_19
<b>Cenário do Teste</b>	Recompensa
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno consegue ver a sua recompensa mais recente. 2. O aluno consegue aceder à tabela de recompensas.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E19: Teste T\_19

<b>ID do Teste</b>	T_19
<b>Requisito Referente</b>	REQ_F_19.1
<b>Cenário do Teste</b>	Receber “crachá pequeno”
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil. 1. A pontuação do aluno atinge mais 100 pontos.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno recebe um “crachá pequeno”. 2. O novo crachá fica visível no seu perfil.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E20: Teste T\_20

<b>ID do Teste</b>	T_20
<b>Requisito Referente</b>	REQ_F_19.2
<b>Cenário do Teste</b>	Receber “crachá grande”
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil. 1. A pontuação do aluno atinge mais 200 pontos.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno recebe um “crachá grande”. 2. O novo crachá fica visível no seu perfil.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

**Testes Mobile**

Tabela E21: Teste T\_21

<b>ID do Teste</b>	T_21
<b>Requisito Referente</b>	REQ_F_10.1
<b>Cenário do Teste</b>	Submeter Diário do Humor
<b>Passos do Teste</b>	<ol style="list-style-type: none"><li>1. O aluno encontra-se no seu perfil.</li><li>2. O aluno acede ao diário onde preenche diariamente o valor do seu humor sob a forma de um range slider.</li><li>3. O valor do humor varia entre zero e cem.</li><li>4. O aluno seleciona um valor.</li><li>5. O aluno submete o valor que selecionou.</li></ol>
<b>Dados do Teste</b>	humor = 74
<b>Resultados Esperados</b>	<ol style="list-style-type: none"><li>1. O aluno consegue submeter o valor selecionado.</li><li>2. O aluno deve ficar com o registo do valor que submeteu.</li><li>3. O aluno só tem possibilidade de realizar uma submissão.</li></ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou



Tabela E22: Teste T\_22

<b>ID do Teste</b>	T_22
<b>Requisito Referente</b>	REQ_F_10.2
<b>Cenário do Teste</b>	Submeter Diário de Atividades
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao diário onde preenche diariamente as atividades que realizou durante o dia.</li> <li>3. O aluno escreve na coluna das atividades, as que realizou durante o dia.</li> <li>4. O aluno seleciona com um <i>check</i>, na linha da atividade correspondente.</li> <li>5. O aluno submete o que preencheu.</li> </ol>
<b>Dados do Teste</b>	atividade1 = 'Listen to a podcast' atividade2 = 'Spend time on social media' atividade3 = '' atividade4 = 'Watch an episode of my favorite serie' atividade5 = 'Workout' id_day_of_week_0_0 <i>checked</i> id_day_of_week_0_2 <i>checked</i> id_day_of_week_1_0 <i>checked</i> id_day_of_week_1_1 <i>checked</i> id_day_of_week_1_2 <i>checked</i> id_day_of_week_1_3 <i>checked</i> id_day_of_week_1_4 <i>checked</i> id_day_of_week_1_5 <i>checked</i> id_day_of_week_1_6 <i>checked</i> id_day_of_week_1_7 <i>checked</i> id_day_of_week_2_0 <i>checked</i> id_day_of_week_3_6 <i>checked</i> id_day_of_week_4_1 <i>checked</i> id_day_of_week_4_3 <i>checked</i> id_day_of_week_4_6 <i>checked</i>
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno consegue submeter o que preencheu na tabela.</li> <li>2. Ao fazer <i>refresh</i> ao <i>browser</i>, os dados inseridos mantêm-se na tabela.</li> <li>3. Não se mantêm <i>checkadas checkboxes</i> em linhas onde nenhuma atividade tenha sido inserida na coluna das atividades</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E23: Teste T\_23

<b>ID do Teste</b>	T_23
<b>Requisito Referente</b>	REQ_F_18
<b>Cenário do Teste</b>	Pontuação
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno consegue ver no seu perfil o local onde se encontra a sua pontuação.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E24: Teste T\_24

<b>ID do Teste</b>	T_24
<b>Requisito Referente</b>	REQ_F_18.1
<b>Cenário do Teste</b>	Ganhar 2 pontos - Preencher “jogo”
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>3. O aluno seleciona um “jogo”.</li> <li>4. O aluno responde às perguntas e submete o “jogo”.</li> <li>5. O aluno acede novamente ao seu perfil.</li> <li>6. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 14 pontos (dois pontos por sete perguntas).
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E25: Teste T\_25

<b>ID do Teste</b>	T_25
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Efetuar <i>login</i>
<b>Passos do Teste</b>	1. O aluno efetua <i>login</i> na plataforma e é redirecionado para o seu próprio perfil. 2. O aluno acede ao local onde se encontra a sua pontuação.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 5 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E26: Teste T\_26

<b>ID do Teste</b>	T_26
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Ver vídeo sem pergunta
<b>Passos do Teste</b>	1. O aluno encontra-se no seu perfil. 2. O aluno acede aos conteúdos a ele lhe destinados. 3. O aluno seleciona um vídeo sem pergunta. 4. O aluno marca a tarefa como concluída. 5. O aluno acede novamente ao seu perfil. 6. O aluno acede ao local onde se encontra a sua pontuação.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. O aluno acede ao seu perfil. 2. O aluno acede ao local onde se encontra a sua pontuação. 3. A pontuação do aluno aumenta 5 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E27: Teste T\_27

<b>ID do Teste</b>	T_27
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Tarefa Diária (Diário do Humor)
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao “Diário do Humor”.</li> <li>3. O aluno seleciona o valor do humor desejado no <i>range slider</i> e submete.</li> <li>4. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno acede ao local onde se encontra a sua pontuação.</li> <li>2. A pontuação do aluno aumenta 5 pontos.</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E28: Teste T\_28

<b>ID do Teste</b>	T_28
<b>Requisito Referente</b>	REQ_F_18.2
<b>Cenário do Teste</b>	Ganhar 5 pontos - Tarefa Diária (Diário de Atividades)
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao “Diário de Atividades”.</li> <li>3. O aluno efetua o registo e submete.</li> <li>4. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	<ol style="list-style-type: none"> <li>1. O aluno acede ao local onde se encontra a sua pontuação.</li> <li>2. A pontuação do aluno aumenta 5 pontos.</li> </ol>
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E29: Teste T\_28

<b>ID do Teste</b>	T_29
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Ver Vídeo com Pergunta
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>3. O aluno seleciona um vídeo com pergunta.</li> <li>4. O aluno marca a tarefa como concluída.</li> <li>4. O aluno acede novamente ao seu perfil.</li> <li>5. O aluno acede ao local onde se encontra a sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E30: Teste T\_29

<b>ID do Teste</b>	T_29
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Realizar <i>Quiz</i>
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno encontra-se no seu perfil.</li> <li>3. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>4. O aluno seleciona um <i>quiz</i>.</li> <li>5. O aluno preenche e submete o <i>quiz</i>.</li> <li>6. O aluno acede novamente ao seu perfil.</li> <li>7. O aluno acede ao local da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E31: Teste T.30

<b>ID do Teste</b>	T_30
<b>Requisito Referente</b>	REQ_F_18.3
<b>Cenário do Teste</b>	Receber 10 Pontos - Preencher Registo Semanal
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede ao local dos registos semanais.</li> <li>3. O aluno seleciona o registo semanal que tenciona preencher.</li> <li>4. O aluno preenche e submete o registo semanal</li> <li>5. O aluno acede novamente ao seu perfil.</li> <li>6. O aluno acede ao local da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 10 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E32: Teste T.31

<b>ID do Teste</b>	T_31
<b>Requisito Referente</b>	REQ_F_18.4
<b>Cenário do Teste</b>	Ganhar 20 Pontos - Ouvir áudio
<b>Passos do Teste</b>	<ol style="list-style-type: none"> <li>1. O aluno encontra-se no seu perfil.</li> <li>2. O aluno acede aos conteúdos a ele lhe destinados.</li> <li>3. O aluno seleciona um áudio de meditação.</li> <li>4. O aluno marca a tarefa como concluída.</li> <li>5. O aluno acede ao seu perfil novamente.</li> <li>6. O aluno acede à zona da sua pontuação.</li> </ol>
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. A pontuação do aluno aumenta 20 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou

Tabela E33: Teste T\_32

<b>ID do Teste</b>	T_32
<b>Requisito Referente</b>	REQ_F_18.4
<b>Cenário do Teste</b>	Ganhar 20 Pontos - Participar na Sessão Presencial
<b>Passos do Teste</b>	1.O professor/psicólogo encontra-se no seu perfil. 2. O professor/psicólogo cria uma sessão. 3. O professor/psicólogo assinala as presenças de uma sessão.
<b>Dados do Teste</b>	N/A
<b>Resultados Esperados</b>	1. Os alunos que se encontraram presentes (P) ou atrasados (L) recebem 20 pontos.
<b>Resultado Atual</b>	Como esperado
<b>Passou/Falhou</b>	Passou