



eGYM: Engineers Gym

JOÃO PAULO COSTA FERNANDES

novembro de 2020

eGYM: Engineers Gym

Reengenharia para a configurabilidade do sistema

João Paulo Costa Fernandes

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

Orientador: Nuno Silva

Co-orientador: Jorge Mendonça

Supervisores: Carla Pinto e Lurdes Babo

Dedicatória

A toda a minha família, especialmente aos meus pais e irmã, pelo apoio incondicional durante todo este percurso académico.

Resumo

Os baixos índices de desempenho nas áreas da Engenharia e principalmente nas áreas da Matemática em países desenvolvidos, como Portugal, causam alguma apreensão nos alunos e professores.

Com o grande avanço da tecnologia e com o aumento da sua relação com o ensino, são cada vez mais procuradas abordagens tecnológicas que auxiliem o progresso escolar dos alunos. Uma das tecnologias mais utilizadas no ensino-aprendizagem são as plataformas *Learning Management Systems* (LMS) que concentram numa plataforma online um número ilimitado de conteúdos lúdicos.

No ano letivo de 2017/2018, um grupo de docentes do Departamento de Matemática do ISEP tomaram a iniciativa e concretizaram uma parceria com o Departamento de Engenharia Informática para a construção de uma plataforma LMS.

No entanto, apesar de já apresentar duas versões diferentes, representando os seus ciclos de desenvolvimento, a plataforma eGYM ainda não apresenta o nível de qualidade pretendido pelos docentes/clientes.

O presente documento demonstra o processo de reengenharia efetuado ao sistema, para que seja possível integrar novos conceitos e funcionalidades que conseguissem eliminar limitações de sistemas anteriores e ir de encontro aos novos objetivos propostos.

Para que melhor se compreendam estes objetivos, é realizada uma análise extensiva do sistema já implementado, através do seu negócio e do seu software já existente.

No final de todo o processo é realizada uma avaliação do sistema, que permita determinar se a reengenharia introduzida no sistema foi bem-sucedida e acrescentou valor ao produto.

Palavras-chave: Ensino-Aprendizagem, *Learning Management System*, *Virtual Learning*, Desenvolvimento *Web*

Abstract

The low-performance rates in the areas of Engineering and mainly in the areas of Mathematics in developed countries, such as Portugal, cause some apprehension in students and teachers.

With the great advance of technology and the increase in its relationship with teaching, technological approaches that help students' school progress are increasingly sought. One of the most used technologies in teaching and learning is the LMS platforms, which concentrate an unlimited number of playful contents on an online platform.

In 2017/2018, a group of teachers from the Mathematics Department of ISEP took the initiative and established a partnership with the Informatics Department to build an LMS platform.

However, despite already presenting two different versions, representing its development cycles, the eGYM platform does not yet present the level of quality desired by the teachers.

This document presents all the reengineering processes carried out in the system so that it would be possible to integrate new concepts and functionalities that could eliminate limitations from previous systems and meet the new objectives proposed.

In order to better understand these objectives, an extensive analysis of the already implemented system is carried out, through its business and its existing software.

At the end of the entire process, an evaluation of the system is done, which allows to determine whether the reengineering introduced in the system was successful and added value to the product.

Keywords: Teaching-Learning, Learning Management System, Virtual Learning, Web Development

Agradecimentos

Agradecer primeiramente ao meu orientador, o Professor Doutor Nuno Silva, pelo apoio, disponibilidade e orientação desde o primeiro dia na concretização deste projeto.

Agradecer também ao professor Jorge Mendonça, a professora Carla Pinto e a professora Lurdes Barbo, supervisores e coorientadores do projeto, pela forma como me acolheram e pela sua total disponibilidade para dialogar e ajudar em qualquer problema enfrentado.

Agradecer à minha família e amigos próximos pelo apoio ao longo destes anos.

E por fim um agradecimento muito especial aos meus pais e irmã pelo seu apoio incondicional nesta longa caminhada, para que no final tudo tenha valido a pena.

Índice

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Contexto | 1 |
| 1.2 | Problema..... | 4 |
| 1.3 | Objetivos..... | 6 |
| 1.4 | Abordagem preconizada..... | 7 |
| 1.5 | Estrutura do documento | 8 |
| 2 | Estado de Arte | 9 |
| 2.1 | Enquadramento teórico | 9 |
| 2.1.1 | <i>Virtual Learning Environment</i> | 10 |
| 2.1.2 | <i>Content Management System</i> | 10 |
| 2.1.3 | <i>Learning Management System</i> | 11 |
| 2.1.4 | <i>Learning Content Management System</i> | 11 |
| 2.1.5 | Plugins..... | 12 |
| 2.2 | Soluções existentes | 12 |
| 2.2.1 | Moodle..... | 13 |
| 2.2.2 | Blackboard Learn..... | 14 |
| 2.2.3 | TalentLMS..... | 15 |
| 2.2.4 | Análise comparativa | 15 |
| 3 | Análise..... | 19 |
| 3.1 | Análise de Negócio | 19 |
| 3.1.1 | 1ª iteração de desenvolvimento 2017/2018 | 19 |
| 3.1.2 | 2ª iteração de desenvolvimento 2018/2019 | 21 |
| 3.1.3 | 3ª iteração de desenvolvimento 2019/2020 (atual) | 22 |
| 3.2 | Análise do software precedente | 24 |
| 3.2.1 | <i>Front Office</i> | 24 |
| 3.2.2 | <i>Back Office</i> | 25 |
| 3.3 | Síntese de Limitações | 25 |
| 3.3.1 | Configurabilidade e extensibilidade de atividades e conteúdos | 26 |
| 3.3.2 | Configurabilidade de autenticação e autorização..... | 26 |
| 3.3.3 | Jogos multijogador (Online)..... | 27 |
| 3.3.4 | Competições | 27 |
| 3.3.5 | Estatística | 28 |
| 3.3.6 | Adoção de componentes/módulos externos..... | 29 |
| 4 | Engenharia de Requisitos | 31 |
| 4.1 | Requisitos funcionais | 31 |
| 4.1.1 | Supervisor | 31 |
| 4.1.2 | Utilizador do <i>Back Office</i> | 33 |
| 4.1.3 | Utilizador do <i>Front Office</i> | 34 |

| | | |
|----------|---|------------|
| 4.2 | Requisitos não funcionais | 35 |
| 4.3 | Resumo | 36 |
| 5 | Desenho | 37 |
| 5.1 | Nível 1 (de contexto) | 38 |
| 5.1.1 | Vista Lógica..... | 38 |
| 5.1.2 | Vista de Processo | 38 |
| 5.2 | Nível 2 (de contentores) | 39 |
| 5.2.1 | Vista Lógica..... | 39 |
| 5.2.2 | Vista de Processo | 40 |
| 5.2.3 | Vista de Implementação | 41 |
| 5.2.4 | Vista de Implantação | 42 |
| 5.3 | Nível 3 (de componentes)..... | 42 |
| 5.3.1 | <i>Front End</i> | 42 |
| 5.3.2 | <i>Back End</i> | 44 |
| 5.3.3 | <i>Competition API</i> | 45 |
| 5.4 | Nível 4 (de classes/código) | 48 |
| 6 | Implementação..... | 49 |
| 6.1 | Descrição Tecnológica | 49 |
| 6.1.1 | Node.js | 49 |
| 6.1.2 | Angular | 52 |
| 6.1.3 | WebSocket & Socket.IO | 56 |
| 6.1.4 | <i>HttpInterceptor (Caching)</i> | 60 |
| 6.1.5 | Localização & Internacionalização (i18n) | 63 |
| 6.2 | Testes..... | 66 |
| 6.2.1 | Metodologia | 66 |
| 6.2.2 | <i>Continuous Delivery / Continuous Integration</i> | 68 |
| 7 | Resultados Obtidos | 71 |
| 7.1 | Funcionalidades Principais..... | 71 |
| 7.1.1 | Configurabilidade e extensibilidade de atividades e conteúdos | 71 |
| 7.1.2 | Configurabilidade de autenticação e autorização | 92 |
| 7.1.3 | Inexistência de jogos multijogador | 97 |
| 7.1.4 | Introdução de competições..... | 104 |
| 7.1.5 | Aprofundamento da Estatística existente..... | 113 |
| 7.2 | Funcionalidades Relevantes | 118 |
| 7.2.1 | Validação dos Formulários | 118 |
| 7.2.2 | Localização | 120 |
| 8 | Experiências e Avaliação..... | 123 |
| 8.1 | Enquadramento teórico | 123 |
| 8.1.1 | Grandezas..... | 123 |
| 8.1.2 | Hipóteses | 124 |
| 8.1.3 | Metodologia de avaliação | 124 |

| | | |
|-----------|---|------------|
| 8.2 | Resultados | 126 |
| 8.2.1 | Utilidade | 127 |
| 8.2.2 | Usabilidade | 130 |
| 8.2.3 | Desempenho..... | 133 |
| 8.3 | Análise..... | 134 |
| 8.3.1 | Utilidade | 135 |
| 8.3.2 | Usabilidade | 135 |
| 8.3.3 | Desempenho..... | 135 |
| 8.3.4 | Conclusão..... | 136 |
| 9 | Sumário e trabalho futuro | 137 |
| 9.1 | Objetivos alcançados | 137 |
| 9.2 | Objetivos não alcançados..... | 138 |
| 9.3 | Limitações e trabalho futuro..... | 138 |
| 9.4 | Apreciação final e pessoal | 139 |
| 10 | Anexos..... | 143 |
| 10.1 | Análise de Valor | 143 |
| 10.1.1 | Introdução | 143 |
| 10.1.2 | Modelo NCD..... | 144 |
| 10.1.3 | Valor, valor para o cliente e valor percebido | 149 |
| 10.1.4 | Proposta de valor..... | 150 |
| 10.1.5 | Modelo Canvas | 151 |
| 10.1.6 | Método AHP | 152 |

Lista de Figuras

| | |
|--|----|
| Figura 1 – Índices de performance da Matemática na União Europeia em 2018 (theOECD, 2020) | 2 |
| Figura 2 – Quantidade de utilizadores nas maiores plataformas LMS em 2018 (Capterra, 2019). | 3 |
| Figura 3 – Número de <i>smartphones</i> existentes e esperados no mundo desde 2016 (Statista, 2020) | 3 |
| Figura 4 – Modelo de domínio da 1ª iteração | 20 |
| Figura 5 – Modelo de domínio da 2ª iteração | 21 |
| Figura 6 - Modelo de domínio para a presente dissertação Versão 1 | 22 |
| Figura 7 - Modelo de domínio para a presente dissertação Versão 2 (final) | 23 |
| Figura 8 – Diagrama de casos de uso do supervisor | 32 |
| Figura 9 – Diagrama de casos de uso do gestor de conteúdos | 34 |
| Figura 10 – Diagrama de casos de uso do Estudante | 35 |
| Figura 11 – Vista Lógica do contexto do sistema | 38 |
| Figura 12 – Vista de Processos do contexto do sistema | 38 |
| Figura 13 – Vista Lógica inicial do sistema | 39 |
| Figura 14 – Vista Lógica final do sistema | 40 |
| Figura 15 - Vista de Processos para operações de leitura | 41 |
| Figura 16 - Vista de Processos para operações de escrita | 41 |
| Figura 17 - Vista de implementação do sistema | 42 |
| Figura 18 – Diagrama de implantação do sistema | 42 |
| Figura 19 – Vista Lógica do <i>Front End</i> do <i>Front Office</i> e do <i>Back Office</i> | 43 |
| Figura 20 – Vista de Processos do <i>Front End</i> do <i>Front Office</i> e do <i>Back Office</i> | 43 |
| Figura 21 – Vista Lógica do <i>Back End</i> do <i>Front Office</i> e do <i>Back Office</i> | 44 |
| Figura 22 – Vista de Processos do <i>Back End</i> do <i>Front Office</i> e do <i>Back Office</i> | 45 |
| Figura 23 - Vista Lógica da <i>Competition API</i> | 46 |
| Figura 24 – Vista de Processos da <i>Competition API</i> | 47 |
| Figura 25 – Estrutura da base de dados não relacional | 48 |
| Figura 26 - Redirecionamento de um pedido para uma <i>route</i> , na <i>Competition API</i> | 50 |
| Figura 27 – Rotas disponíveis para a entidade competição, na <i>Competition API</i> | 50 |
| Figura 28 - Controlador e respetiva função para o retorno das competições existentes, na <i>Competition API</i> | 51 |
| Figura 29 – Serviço e respetiva função para a procura de competições existentes, na <i>Competition API</i> | 51 |
| Figura 30 - <i>Schema</i> de uma competição, na <i>Competition API</i> | 52 |
| Figura 31 – Excerto de código onde se verifica a definição dos <i>components</i> para cada URL, no <i>Front Office</i> | 53 |
| Figura 32 - Excerto de código com a definição das variáveis de um <i>component</i> , no <i>Front Office</i> | 53 |

| | |
|--|----|
| Figura 33 - Excerto de código do <i>template</i> para a página de uma dada competição, no <i>Front Office</i> | 54 |
| Figura 34 - Excerto de código de um <i>component</i> onde se valida a navegação para uma respetiva questão de uma competição, no <i>Front Office</i> | 54 |
| Figura 35 – Excerto de código de um <i>service</i> dedicado as competições, no <i>Front Office</i> | 55 |
| Figura 36 – Excerto de código de um <i>component</i> onde se verifica a injeção de um <i>service</i> assim como a sua utilização, no <i>Front Office</i> | 56 |
| Figura 37 – Serviço Socket.IO, no <i>Front Office</i> | 57 |
| Figura 38 - Excerto de código para uma pesquisa de sala de jogo, no <i>Front Office</i> | 58 |
| Figura 39 – Algoritmo de emparelhamento de jogadores, na <i>Game API</i> | 58 |
| Figura 40 - Excerto de código para a captura de pedidos de emparelhamento, na <i>Game API</i> .. | 59 |
| Figura 41 – Excerto de código para a captura do evento da entrada de um novo utilizador na sala de jogo, no <i>Front Office</i> | 59 |
| Figura 42 - Excerto de código para o lançamento do evento aquando do abandono de uma sala de jogo, no <i>Front Office</i> | 59 |
| Figura 43 - Excerto de código para a captura do evento de abandono de uma sala de jogo, na <i>Game API</i> | 59 |
| Figura 44 – Excerto de código para a atualização dos utilizadores numa sala de jogo depois de algum abandonar, no <i>Front Office</i> | 60 |
| Figura 45 – Excerto de código que representa o início de um jogo multijogador, na <i>Game API</i> | 60 |
| Figura 46 – Excerto de código que demonstra a captura do evento que marca o início de um jogo multijogador, no <i>Front Office</i> | 60 |
| Figura 47 - Excerto de código de um <i>HttpInterceptor</i> , no <i>Front Office</i> | 61 |
| Figura 48 - Excerto de código de uma verificação na <i>cache</i> pelo pedido efetuado num <i>HttpInterceptor</i> , no <i>Front Office</i> | 62 |
| Figura 49 - Excerto de código com o <i>reset</i> da <i>cache</i> ao fim de dez segundos, no <i>Front Office</i> . | 62 |
| Figura 50 - Excerto de código com o pedido de eliminação de <i>cache</i> para um dado pedido, no <i>Front Office</i> | 63 |
| Figura 51 - Marcação de texto através do uso do atributo <i>i18n</i> , no <i>Front Office</i> | 64 |
| Figura 52 – Marcação do texto com um identificador específico através do uso do atributo <i>i18n</i> , no <i>Front Office</i> | 64 |
| Figura 53 - Excerto de código com a marcação do texto " <i>Password</i> ", no <i>Front Office</i> | 64 |
| Figura 54 - Tradução de um excerto de texto da linguagem original para Português, no <i>Front Office</i> | 64 |
| Figura 55 - Definição dos locais suportados pela aplicação no ficheiro, no <i>Front Office</i> | 65 |
| Figura 56 – Definição do uso de localização numa aplicação no seu ficheiro de configuração, no <i>Front Office</i> | 65 |
| Figura 57 – Determinação do local a utilizar para o utilizador, no <i>Front Office</i> | 65 |
| Figura 58 – Excerto de código para os diferentes testes para uma área, no <i>Back End</i> do <i>Back Office</i> | 67 |
| Figura 59 – Configuração das ligações às bases de dados, no <i>Back End</i> do <i>Back Office</i> | 68 |
| Figura 60 – Ficheiro de configuração da <i>pipeline</i> | 68 |

| | |
|--|----|
| Figura 61 – <i>Git branching</i> simplificado (Driessen, 2010) | 69 |
| Figura 62 - <i>Sidebar</i> de navegação do <i>Back Office</i> | 72 |
| Figura 63 - Criação de uma área em sistemas anteriores (João Fernandes and José Barros, 2018) | 73 |
| Figura 64 - Exemplo de criação de uma Área, no <i>Back Office</i> | 74 |
| Figura 65 - Definição dos módulos de uma Área, no <i>Back Office</i> | 74 |
| Figura 66 - Definição dos jogos multijogador de uma Área, no <i>Back Office</i> | 75 |
| Figura 67 – Listagem das áreas existentes por área de estudo, no <i>Front Office</i> | 75 |
| Figura 68 - Apresentação de uma Área, no <i>Front Office</i> | 76 |
| Figura 69 – Alerta do número insuficiente de pontos para desbloquear um conteúdo, no <i>Front Office</i> | 76 |
| Figura 70 – Processo de edição de uma Área, no <i>Back Office</i> | 77 |
| Figura 71 – Definição das áreas para as quais esta aula (módulo) se encontra presente, no <i>Back Office</i> | 78 |
| Figura 72 – Definição dos jogos didáticos de uma aula (módulo), no <i>Back Office</i> | 78 |
| Figura 73 - Processo de criação de uma vídeo-aula fornecendo o nome e o ficheiro, no <i>Back Office</i> | 79 |
| Figura 74 – Seleção de uma vídeo-aula num módulo depois de esta ser criada no sistema, no <i>Back Office</i> | 80 |
| Figura 75 – Processo de criação de um material de suporte fornecendo o nome e o ficheiro, no <i>Back Office</i> | 80 |
| Figura 76 – Seleção de um material de suporte num módulo depois de este ser criado no sistema, no <i>Back Office</i> | 81 |
| Figura 77 - Representação dos módulos e os seus respetivos conteúdos, no <i>Front Office</i> | 81 |
| Figura 78 – Reprodução de uma vídeo-aula, no <i>Front Office</i> | 82 |
| Figura 79 – Visualização de um material de suporte, no <i>Front Office</i> | 82 |
| Figura 80 – Informação visual do consumo de um conteúdo de um módulo, no <i>Front Office</i> .. | 83 |
| Figura 81 - Processo de edição de um módulo, no <i>Back Office</i> | 83 |
| Figura 82 - Criação de um Jogo em sistemas anteriores (Luís Peixoto, 2019) | 84 |
| Figura 83 – Exemplo do processo de criação de um jogo didático | 85 |
| Figura 84 - Processo de criação de uma questão no sistema anterior (Luís Peixoto, 2019) | 85 |
| Figura 85 – Seleção e criação de questões, através de uma janela de diálogo, no <i>Back Office</i> .. | 86 |
| Figura 86 – Exemplo da criação de uma questão, no <i>Back Office</i> | 86 |
| Figura 87 – Seleção de questões previamente criadas para um dado Jogo, no <i>Back Office</i> | 87 |
| Figura 88 – Gestão das questões existentes num dado Jogo, no <i>Back Office</i> | 87 |
| Figura 89 – Definição dos módulos aos quais um dado Jogo pertence, no <i>Back Office</i> | 88 |
| Figura 90 – Listagem dos jogos didáticos de um módulo, sobre “Training Games”, no <i>Front Office</i> | 88 |
| Figura 91 – Janela informativa de um dado jogo didático, no <i>Front Office</i> | 89 |
| Figura 92 – Interface de um jogo didático, no <i>Front Office</i> | 89 |
| Figura 93 - Mensagem de aviso sobre questão não respondidas, no <i>Front Office</i> | 90 |
| Figura 94 – Mensagem de fim de jogo didático, no <i>Front Office</i> | 90 |

| | |
|--|-----|
| Figura 95 - Mensagem de fim de Jogo quando a pontuação obtida não é a máxima, no <i>Front Office</i> | 91 |
| Figura 96 - Revisão de um jogo didático, no <i>Front Office</i> | 91 |
| Figura 97 - Indicação visual da melhor pontuação de um utilizador num dado Jogo, no <i>Front Office</i> | 92 |
| Figura 98 - Edição do Jogo “ <i>Testing Game With MathJax & Images</i> ”, no <i>Back Office</i> | 92 |
| Figura 99 - Convite de um utilizador para aceder ao <i>Back office</i> , no <i>Back Office</i> | 93 |
| Figura 100 – Email recebido por utilizadores convidados..... | 93 |
| Figura 101 – Navegação para a criação de novos <i>roles</i> , no <i>Back Office</i> | 94 |
| Figura 102 – Criação de um novo nível de acesso, no <i>Back Office</i> | 94 |
| Figura 103 – Atribuição de um <i>role</i> a um utilizador do <i>Back Office</i> , no <i>Back Office</i> | 95 |
| Figura 104 - Navegação lateral restrita, no <i>Back Office</i> | 95 |
| Figura 105 - Grupos de acesso, no <i>Back Office</i> | 96 |
| Figura 106 - Criação de um Grupo de Acesso, no <i>Back Office</i> | 96 |
| Figura 107 - Definição dos grupos de acesso para um dado módulo dentro de uma Área, no <i>Back Office</i> | 97 |
| Figura 108 – Exemplo do processo de criação de um Jogo multijogador, no <i>Back Office</i> | 98 |
| Figura 109 – Gestão das questões existentes num Jogo multijogador, no <i>Back Office</i> | 98 |
| Figura 110 – Definição das áreas para os quais um Jogo multijogador se encontra inserido, no <i>Back Office</i> | 99 |
| Figura 111 - Apresentação dos jogos multijogador existentes, no <i>Front Office</i> | 99 |
| Figura 112 – Informação relativa de um Jogo multijogador, no <i>Front Office</i> | 100 |
| Figura 113 – Emparelhamento entre dois utilizadores para um jogo multijogador, no <i>Front Office</i> | 100 |
| Figura 114 - Criação de uma sala de jogo antecipada por indicação dos seus utilizadores, no <i>Front Office</i> | 101 |
| Figura 115 - Contagem decrescente para o início de um jogo multijogador, no <i>Front Office</i> | 101 |
| Figura 116 - Interface de um jogo multijogador, no <i>Front Office</i> | 102 |
| Figura 117 – Mensagem apresentada aos utilizadores que se encontrem a espera da determinação do vencedor, no <i>Front Office</i> | 102 |
| Figura 118 - Mensagem apresentada ao vencedor de um jogo multijogador, no <i>Front Office</i> | 103 |
| Figura 119 – Revisão de um jogo multijogador, no <i>Front Office</i> | 103 |
| Figura 120 – Exemplo de uma criação de uma Competição, no <i>Back Office</i> | 104 |
| Figura 121 - Definição dos parâmetros data-hora para uma Competição, no caso específico da data de início para registos, no <i>Back Office</i> | 105 |
| Figura 122 – Definição de parâmetros adicionais no processo de criação de uma Competição, no <i>Back Office</i> | 106 |
| Figura 123 – Definição da calendarização de uma Competição, no <i>Back Office</i> | 106 |
| Figura 124 – Configuração de uma questão para uma Competição, no <i>Back Office</i> | 107 |
| Figura 125 - Adição de múltiplas datas agendadas e as suas respetivas questões para uma Competição, no <i>Back Office</i> | 108 |
| Figura 126 - Listagem de Competições agendadas, no <i>Front Office</i> | 109 |

| | |
|---|-----|
| Figura 127 - Listagem de Competição abertas para registo, no <i>Front Office</i> | 109 |
| Figura 128 - Listagem de Competições ativas de um utilizador, no <i>Front Office</i> | 110 |
| Figura 129 - Calendarização de uma Competição, no <i>Front Office</i> | 110 |
| Figura 130 - Competição em progresso, no <i>Front Office</i> | 111 |
| Figura 131 – Revisão de uma questão de uma Competição, no <i>Front Office</i> | 111 |
| Figura 132 - Classificação de uma dada Competição, no <i>Front Office</i> | 112 |
| Figura 133 - Exemplo de uma edição de uma Competição, no <i>Back Office</i> | 112 |
| Figura 134 – Estatística disponibilizada na versão anterior do sistema (Luís Peixoto, 2019) .. | 113 |
| Figura 135 – <i>Sidebar</i> de navegação com as opções de Estatística, no <i>Back Office</i> | 113 |
| Figura 136 – Estatística relativas a atividade existente no sistema, no <i>Back Office</i> | 114 |
| Figura 137 – Visualização estatística da distribuição de países dos utilizadores registados no sistema, no <i>Back Office</i> | 115 |
| Figura 138 – Pesquisa estatística da atividade existente no sistema entre 01/08/2020 e 14/10/2020, no <i>Back Office</i> | 116 |
| Figura 139 – Consulta estatística da popularidade dos conteúdos existentes nas áreas selecionadas, no <i>Back Office</i> | 117 |
| Figura 140 – Consulta estatística da popularidade dos jogos didáticos existentes nas áreas selecionadas, no <i>Back Office</i> | 117 |
| Figura 141 – Consulta estatística das percentagens de respostas corretas e erradas de um jogo, no <i>Back Office</i> | 118 |
| Figura 142 – Validações existentes em formulários, no <i>Back Office</i> | 119 |
| Figura 143 – Validação da introdução do tempo limite de um Jogo, no <i>Back Office</i> | 119 |
| Figura 144 - Validação de um email, no <i>Back Office</i> | 120 |
| Figura 145 - Validação de datas de uma Competição, no <i>Back Office</i> | 120 |
| Figura 146 – Informação do estado inválido de um formulário, no <i>Back Office</i> | 120 |
| Figura 147 – Visualização da opção para a seleção de uma língua, no <i>Front Office</i> | 121 |
| Figura 148 – Visualização das diferentes línguas oferecidas, no <i>Front Office</i> | 121 |
| Figura 149 – Visualização da atualização do texto existente na plataforma após a seleção da língua Portuguesa, no <i>Front Office</i> | 122 |
| Figura 150 - Barra de navegação traduzida para Português, no <i>Front Office</i> | 122 |
| Figura 151 – 1ª pergunta de utilidade do inquérito | 127 |
| Figura 152 - 2ª pergunta de utilidade do inquérito | 127 |
| Figura 153 – 3ª pergunta de utilidade do inquérito | 128 |
| Figura 154 - 4ª pergunta de utilidade do inquérito | 128 |
| Figura 155 – 5ª pergunta de utilidade do inquérito | 128 |
| Figura 156 - 6ª pergunta de utilidade do inquérito | 129 |
| Figura 157 – 7ª pergunta de utilidade do inquérito | 129 |
| Figura 158 – 8ª pergunta de utilidade do inquérito | 129 |
| Figura 159 - 1ª pergunta de usabilidade do inquérito | 130 |
| Figura 160 – 2ª pergunta de usabilidade do inquérito | 130 |
| Figura 161 – 3ª pergunta de usabilidade do inquérito | 130 |
| Figura 162 – 4ª pergunta de usabilidade do inquérito | 131 |
| Figura 163 – 5ª pergunta de usabilidade do inquérito | 131 |

| | |
|--|-----|
| Figura 164 – 6ª pergunta de usabilidade do inquérito | 131 |
| Figura 165 – 7ª pergunta de usabilidade do inquérito | 132 |
| Figura 166 – 8ª pergunta de usabilidade do inquérito | 132 |
| Figura 167 – 9ª pergunta de usabilidade do inquérito | 132 |
| Figura 168 - 10ª pergunta de usabilidade do inquérito | 133 |
| Figura 169 – 1ª pergunta de desempenho do inquérito..... | 133 |
| Figura 170 – 2ª pergunta de desempenho do inquérito..... | 133 |
| Figura 171 - 3ª pergunta de desempenho do inquérito | 134 |
| Figura 172 - 4ª pergunta de desempenho do inquérito | 134 |
| Figura 173 – O processo de inovação dividido em três partes (Koen, 2002) | 144 |
| Figura 174 – Modelo NCD | 145 |
| Figura 175 – Modelo Canvas | 151 |
| Figura 176 – Árvore hierárquica de decisão..... | 154 |

Lista de Tabelas

| | |
|---|-----|
| Tabela 1 – Análise comparativa de soluções existentes | 16 |
| Tabela 2 – Requisitos não funcionais segundo o modelo de classificação FURPS+..... | 36 |
| Tabela 3 – Sacrifícios/Benefícios | 150 |
| Tabela 4 – Escala Fundamental adaptada de Saaty (Saaty, 1990)..... | 154 |
| Tabela 5 – Matriz de comparação | 155 |
| Tabela 6 – Matriz de comparação dos critérios de segundo nível | 155 |
| Tabela 7 – Matriz normalizada dos critérios de segundo nível | 156 |
| Tabela 8 – Matriz normalizada dos critérios de segundo nível com prioridades relativas | 156 |
| Tabela 9 – Valores de IR para matrizes quadradas de ordem n | 157 |
| Tabela 10 – Matriz de comparação do critério Desempenho | 158 |
| Tabela 11 – Matriz de comparação do critério Desempenho de segundo nível | 158 |
| Tabela 12 – Matriz normalizada para o critério de Desempenho de segundo nível | 158 |
| Tabela 13 – Matriz normalizada para o critério de Desempenho de segundo nível com prioridades relativas | 159 |
| Tabela 14 – Matriz de comparação para o critério Custo | 159 |
| Tabela 15 – Matriz de comparação para o critério Custo de segundo nível | 159 |
| Tabela 16 Matriz normalizada para o critério Custo de segundo nível..... | 159 |
| Tabela 17 – Matriz normalizada para o critério Custo de segundo nível com prioridades relativas..... | 160 |
| Tabela 18 – Matriz de comparação para o critério Dificuldade de implementação | 160 |
| Tabela 19 – Matriz de comparação para o critério Dificuldade de implementação de segundo nível..... | 160 |
| Tabela 20 – Matriz normalizada para o critério Dificuldade de implementação de segundo nível..... | 160 |
| Tabela 21 – Matriz normalizada para o critério Dificuldade de implementação de segundo nível com prioridades relativas..... | 161 |

Acrónimos e Símbolos

| | |
|-------------|---|
| AHP | <i>Analytic Hierarchy Process</i> |
| API | <i>Application Programming Interface</i> |
| CDN | <i>Content Delivery Network</i> |
| CMS | <i>Course Management System</i> |
| CRUD | <i>Create, Read, Update and Delete</i> |
| DEI | Departamento de Engenharia Informática |
| DMA | Departamento de Matemática |
| DTO | <i>Data Transfer Object</i> |
| FFE | <i>Fuzzy Front End</i> |
| HTML | <i>HyperText Markup Language</i> |
| IC | Índice de Consistência |
| ISEP | Instituto Superior de Engenharia do Porto |
| IR | Índice Aleatório |
| JSON | <i>JavaScript Object Notation</i> |
| LCSM | <i>Learning Content Management System</i> |
| LMS | <i>Learning Management System</i> |
| NCD | <i>New Concept Development</i> |
| NPD | <i>New Product Development</i> |
| PISA | Programa Internacional de Avaliação de Alunos |
| RC | Razão de Consistência |
| REST | <i>Representational State Transfer</i> |
| SPA | <i>Single-Page Application</i> |
| SUS | <i>System Usability Scale</i> |
| TCP | <i>Transmission Control Protocol</i> |

| | |
|------------|-------------------------------------|
| UML | <i>Unified Modeling Language</i> |
| URL | <i>Uniform Resource Locator</i> |
| VSM | <i>Vector Space Model</i> |
| VLE | <i>Virtual Learning Environment</i> |

1 Introdução

Neste primeiro capítulo realiza-se uma introdução sobre a reengenharia do sistema através de cinco distintas secções.

Numa primeira secção realiza-se uma contextualização do projeto da presente dissertação, procurando assim enquadrar o leitor no respetivo projeto.

Na segunda secção procede-se à descrição do problema a resolver no contexto da dissertação, tendo em conta o sistema já existente.

A terceira secção é reservada para a enunciação dos objetivos atingir no final da dissertação.

Quanto à quarta secção esta descreve a abordagem preconizada para a presente dissertação.

Por fim numa quinta e última secção é descrita a presente estrutura do documento.

1.1 Contexto

Nos dias de hoje os resultados relativos à aprendizagem de conceitos relacionados com as Engenharias e em particular com a Matemática em países desenvolvidos apresentam ainda índices de desempenho baixos apesar se encontrarem numa tendência positiva (PISA, 2019).

Num estudo do Programa Internacional de Avaliação de Alunos (PISA) baseados em resultados provenientes do ano de 2018, Portugal classificou-se ligeiramente acima da média dos 35 países avaliados da OCDE, fixando-se na 23ª posição com uma pontuação de 494, quando a média da OCDE é de 489 pontos (theOECD, 2020) (Figura 1).

O sistema de pontuação utilizado pelo PISA, representa uma medição da facilidade de reconhecimento que os alunos têm do papel que a Matemática desempenha no mundo real, através de julgamentos e decisões bem fundamentadas sobre problemas apresentados.

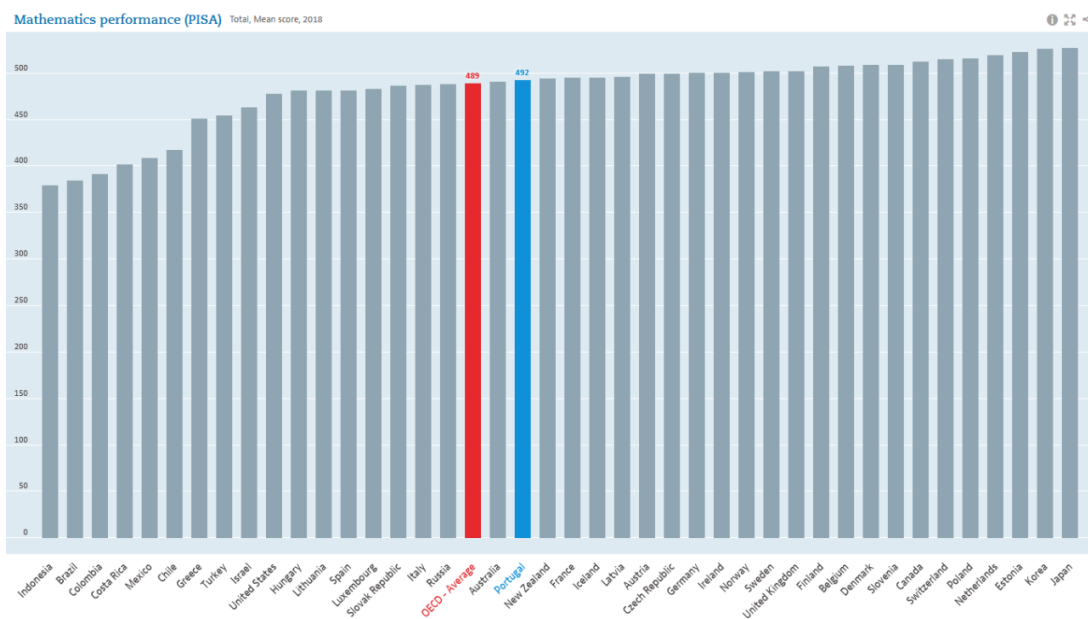


Figura 1 – Índices de performance da Matemática na União Europeia em 2018 (theOECD, 2020)

É habitual os estudantes definirem a Matemática como um dos grandes obstáculos no seu progresso escolar, criticando muitas vezes a complexidade da mesma. Mas o que pode tornar a Matemática tão difícil está sujeito a opiniões variadas. Alguns apontam para problemas psicológicos como ansiedade (Maloney and Beilock, 2012), enquanto outros baseiam a sua opinião no facto de a Matemática ser uma disciplina cumulativa, onde os alicerces de conceitos posteriores se baseiam em conceitos anteriores, tornando assim a Matemática numa grande “bola de neve” (Fleming, 2019).

No entanto, o problema nem sempre recai sobre os alunos. Um dos problemas causadores destas taxas de insucesso é a escolha de uma abordagem errada na transmissão dos conteúdos entre os professores e os seus alunos. Os problemas nem sempre são apresentados da melhor maneira ou as soluções não representam a melhor maneira de exprimir os resultados aos seus alunos.

Com os grandes avanços tecnológicos dos últimos tempos, uma das abordagens mais populares é a utilização de plataformas LMS (*Learning Management Systems*). Estes sistemas concentram um vasto e variado número de conteúdos numa única localização, fornecendo acesso ilimitado a material de aprendizagem (eLearning Industry, 2017).

Num estudo realizado em 2018 pela plataforma Capterra (Capterra, 2019), as 20 maiores plataformas LMS do mercado agregariam mais de 450 milhões de utilizadores, sendo que apenas as cinco maiores representavam mais de metade deste número, cerca de 250 milhões (Figura 2).

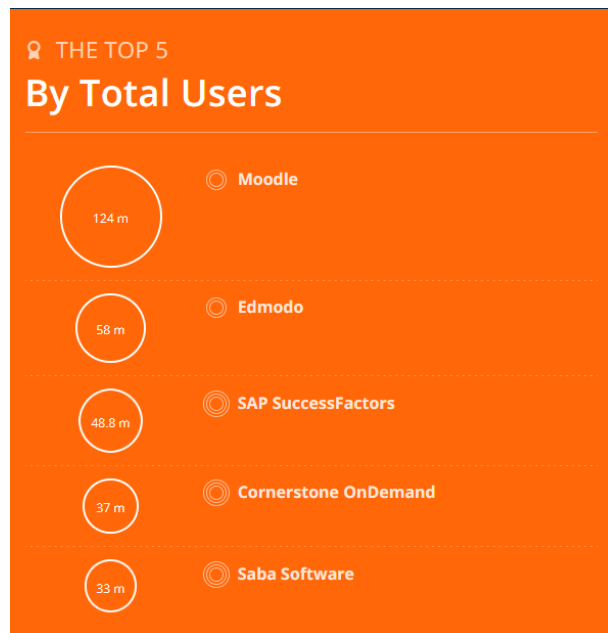


Figura 2 – Quantidade de utilizadores nas maiores plataformas LMS em 2018 (Capterra, 2019).

Uma das suas maiores atrações passa pela sua possível utilização através de muitos tipos de dispositivos, seja ele computador ou *smartphone*, acompanhando assim o impressionante processo de digitalização mundial que acontece nos dias de hoje (Figura 3).

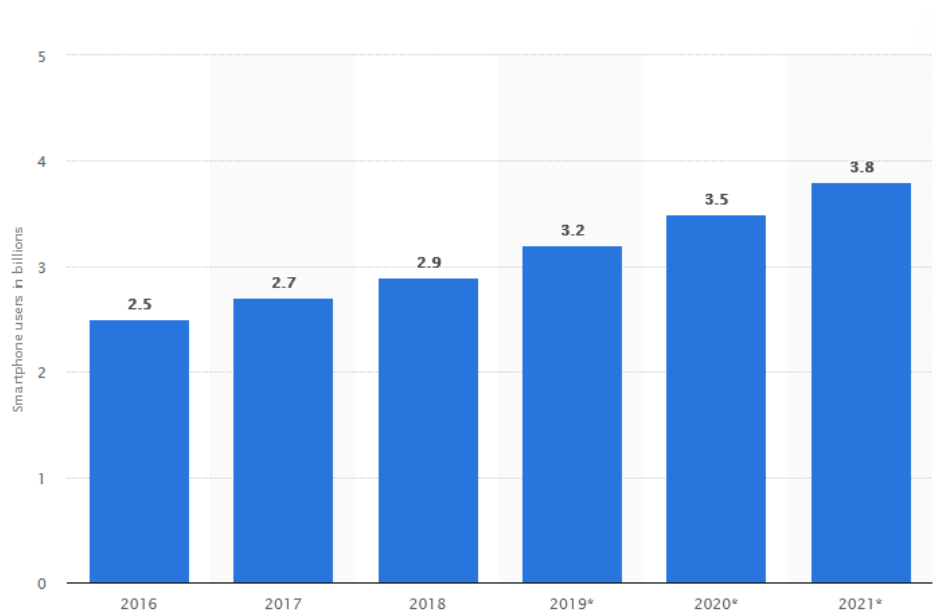


Figura 3 – Número de *smartphones* existentes e esperados no mundo desde 2016 (Statista, 2020)

Porque existe uma correlação permanente entre as Matemáticas e as Engenharias, faz com que o ISEP enfrente o mesmo problema no seu sistema de aprendizagem. Torna-se, assim, necessário procurar soluções que combatam e melhorem os sistemas de aprendizagem

clássicos, o denominado ensino presencial, utilizados no ISEP e em várias instituições de ensino.

Um grupo de professores do Departamento de Matemática (DMA) do ISEP, que procura ativamente por soluções que possam aumentar os índices de aproveitamento dos seus estudantes, decidiu idealizar a criação de uma plataforma de aprendizagem digital. Em parceria com o Departamento de Engenharia Informática do ISEP, os docentes deram início ao desenvolvimento de uma plataforma LMS, onde os alunos das diferentes áreas das Matemáticas dos cursos de Engenharia, pudessem estudar de uma forma digital e interativa.

Surge assim a plataforma eGYM. Este novo sistema pretende oferecer aos seus utilizadores diversos conteúdos e atividades semelhantes às existentes no ensino superior, com ênfase especial nas diferentes áreas das Matemáticas, num ambiente virtual e interativo promovendo novas formas de ensino-aprendizagem. O sistema pretende ainda funcionar como uma plataforma de apoio ao progresso escolar dos seus alunos, tendo em conta as diferentes dificuldades de conteúdos e atividades oferecidas, podendo ser utilizada como uma plataforma de consolidação de conceitos académicos ou até mesmo como plataforma de preparação para novos conceitos a ser lecionados.

Esta plataforma teve o seu início de desenvolvimento no letivo de 2017/2018 no âmbito da unidade curricular Projeto/Estágio (PESTI) da Licenciatura de Engenharia Informática pelos alunos João Fernandes (o autor desta presente dissertação) e José Barros. No ano letivo seguinte, 2018/2019, dois outros alunos, Luís Peixoto e Paulo Magalhães iniciaram um novo ciclo de desenvolvimento.

1.2 Problema

Em qualquer software desenvolvido é comum que o produto nunca chegue a ser completado, estando sempre aberto a novas alterações, sejam estas através correções, modificações em funcionalidades existentes ou através da introdução de novas funcionalidades.

A plataforma eGYM enquadra-se nesta situação; apesar de ter sido sujeita a várias alterações de melhoria nos seus dois ciclos de desenvolvimento, ainda se encontra longe do nível pretendido para uma plataforma LMS.

Na sua essência, uma plataforma LMS prima pela capacidade de gestão dos conteúdos existentes. Normalmente aliada ao conceito de gestão encontra-se o termo de configurabilidade. Por configurabilidade de conteúdos compreende-se a capacidade do sistema adotar e manipular tipos de conteúdos não especificados em tempo de desenvolvimento, mas especificados em tempo de configuração (depois de desenvolvimento mas anterior à execução), não estando assim condicionado a um número finito ou pré-definido de tipos de conteúdo. É também esperado que estes conteúdos possam ser extensíveis ao longo do sistema e do tempo, podendo ser reaproveitados em qualquer momento e em qualquer situação, conforme o desejado.

Para além disto a plataforma interage de uma maneira intensiva com os seus utilizadores. Podem assim existir na plataforma utilizadores com características e permissões de acesso diferentes, necessitando assim que o sistema se encontre disponível para oferecer uma configurabilidade de autenticação e de autorização para os mesmos. Esta configurabilidade pode permitir limitar as ações que os utilizadores possam executar no sistema, mas também limitar os conteúdos e atividades que estes possam consumir.

Um dos problemas das versões anteriores do eGYM passa pela inexistência de jogos multijogadores na plataforma. Por jogo multijogador entende-se um tipo de atividade, que pode ser realizada por vários utilizadores da plataforma em tempo real, sendo que estes participam e jogam uns contra os outros, havendo no final a coroação de um vencedor.

Uma das maiores atrações da plataforma eGYM passa pela existência de competições que permitem aos seus utilizadores competir em tempo real com outros utilizadores da plataforma. Este conceito surgiu no desenvolvimento da primeira iteração de desenvolvimento do eGYM, mas que, no entanto, não acabou por ser implementada. Por competições entendem-se locais reservados para diferentes utilizadores competirem entre si, introduzindo assim uma maior veia competitiva ao sistema, num período limitado e definido para o tal.

Considerando a finalidade do sistema a desenvolver e as suas características torna-se necessário que os seus supervisores e administradores tenham acesso a um conjunto de dados estatísticos que permitam observar o desempenho dos seus utilizadores, assim como o desempenho dos conteúdos e atividades desenvolvidas, permitindo assim descobrir se os conteúdos e atividades desenvolvidas servem o propósito pretendido. É assim esperado que o sistema ofereça uma *overview* estatística de todos os conteúdos e atividades existentes nas plataformas, permitindo para cada um destes, diferentes filtros de pesquisa.

Por último, mas não menos importante, sentiu-se uma necessidade de adotar a utilização de componentes externos, designados de plugins, para possíveis ocasiões onde a configurabilidade embutida no sistema não consiga resolver as necessidades dos utilizadores. Apesar de o novo sistema pretendido contar com um alto nível de configurabilidade, haverá sempre situações onde esta configurabilidade não se mostre suficiente e como tal a adoção de componentes externos será uma solução a ter em conta para tais situações.

Assim, para este novo ciclo de desenvolvimento, considera-se que o software apresenta os seguintes problemas:

- **Inexistência de configurabilidade de autenticação e autorização;**
- **Inexistência de configurabilidade e extensibilidade de atividades e conteúdos;**
- **Inexistência de jogos multijogador;**
- **Inexistência de competições;**
- **Funcionalidades do módulo de estatística muito limitadas;**
- **Adoção de componentes/módulos funcionais externos (plugins).**

1.3 Objetivos

O objetivo principal deste projeto dissertação concentra-se na reengenharia da plataforma LMS já desenvolvida, introduzindo características de configurabilidade e extensibilidade, tentando assim eliminar as limitações anteriormente identificadas.

O novo sistema deve, assim, aliar estas novas características de extensibilidade e configurabilidade aos conceitos de negócio identificados, integrando com todas as funcionalidades anteriormente desenvolvidas.

No âmbito desta dissertação, é possível delinear os seguintes objetivos:

1. Capacidade de gestão dinâmica de autorizações existentes na plataforma:
 - a. O sistema deve permitir a introdução de novos papéis de utilizadores (*roles*) com diferentes níveis de acesso;
 - b. O sistema deve permitir a configuração destes novos papéis de uma forma dinâmica e aprofundada permitindo assim um maior controlo dos acessos dos seus utilizadores;
2. Configuração de diferentes atividades e conteúdos:
 - a. Configurabilidade de conteúdos, permitindo adicionar diferentes tipos de conteúdos diferentes;
 - b. Introdução de conteúdos de diferentes localizações;
 - c. Capacidade de extensão de conteúdos, permitindo reutilizações dos mesmos.
 - d. Configurabilidade de atividades, aumentando o número e diversidade de atividades pedagógica possíveis;
3. Desenvolvimento de jogos multijogador:
 - a. Os utilizadores da plataforma podem participar em jogos multijogador;
 - b. Estas jogos são semelhantes aos jogos didáticos de um único jogador, existentes nas diferentes áreas da Engenharia, apresentando, no entanto, informação sobre o progresso de todos os utilizadores na sala de jogo;
 - c. Para o início de um jogo multijogador, devem existir pelo menos dois jogadores, e até ao máximo de quatro jogadores, prontos a participar num dado momento;
4. Introdução de competições:
 - a. Os utilizadores da plataforma podem competir num certo espaço temporal em competições criadas por supervisores;
 - b. Como em qualquer situação competitiva, é esperado a implementação de *rankings*;
5. Aprofundamento da estatística existente:
 - a. O sistema deve aprofundar o conceito estatístico já implementado;
 - b. O sistema deve oferecer dados estatísticos para cada tipo de conteúdo e atividade existente no sistema;
 - c. Cada pesquisa deve estar acompanhada de diferentes filtros de pesquisa;
6. Introdução de módulos funcionais externos:
 - a. Que permitam a adição de novas funcionalidades através de componentes externos;

- b. Que permitam a extensão de funcionalidades já existentes através da utilização de componentes externos.

1.4 Abordagem preconizada

Tratando-se de uma reengenharia de um sistema desenvolvido, a primeira fase passa pelo levantamento das limitações existentes no sistema anterior, construindo assim o problema a resolver. Depois de definidas todas as limitações consideradas pelo cliente, procede-se a enumeração dos objetivos a alcançar para esta nova iteração de desenvolvimento e para a presente dissertação.

Com esta primeira fase concluída procede-se à análise do negócio existente, procurando analisar todos os conceitos existentes de modo a compreender as limitações encontradas e os objetivos a alcançar. Esta análise de negócio é na sua maioria realizada em conjunto com o cliente, permitindo assim, sempre que necessário, esclarecer questões que surjam em relação ao negócio. Esta é sempre uma fase fundamental na reengenharia de sistemas, pois sem uma boa compreensão do negócio, torna-se impossível atingir os objetivos pretendidos.

Depois de analisado o negócio anteriormente existente, procede-se à reestruturação do modelo de negócio existente através da introdução ou remoção de conceitos, tentando, assim, alcançar o modelo de negócio pretendido para esta reengenharia do sistema. Mais uma vez todo este processo é na sua maioria acompanhado pelo cliente, salientando o diálogo entre ambas as partes.

Com o modelo de negócio bem estruturado e estando ambas as partes de acordo com o mesmo, inicia-se o processo de levantamento de requisitos do novo sistema. É aqui que são enumerados os diferentes tipos de utilizadores existentes, assim como as suas funcionalidades individuais ou coletivas. É também nesta fase que são considerados todos os requisitos de sistema.

Com as fases anteriores concluídas, dá-se início ao processo de desenho do sistema em diferentes níveis de granularidade e de acordo com as tecnologias escolhidas, conseguindo-se assim obter uma estrutura esqueleto do sistema. É nesta altura que são estudadas e analisadas diferentes tecnologias de implementação, e decidir quais utilizar.

Com o desenho terminado, inicia-se finalmente a fase de construção do sistema, sendo que neste caso esta assenta numa reengenharia e não no desenvolvimento de um sistema totalmente novo. O processo de construção tem como base as decisões tomadas nas fases anteriores.

Por fim, mas não menos importante, procede-se à realização da avaliação da reengenharia efetuada no sistema. Apesar das suas grandezas e metodologias a utilizar serem definidas antes do início do processo de implementação, é apenas neste momento que se procede ao

levantamento das respostas dos inquiridos. Pode-se assim com uma análise dos resultados das respostas obtidas, avaliar a qualidade do sistema, tendo em conta as grandezas definidas.

1.5 Estrutura do documento

Para além deste, o documento está estruturado em mais outros 7 capítulos:

- **Contexto:** apresenta uma breve análise da versão anterior do sistema quanto as suas aplicações, detalhando as funcionalidades implementadas e as suas limitações;
- **Estado de Arte:** apresenta o levantamento e análise efetuados sobre o estado de arte, descrevendo conceitos teóricos e apresentando um conjunto de soluções existentes no mercado comparáveis com a preconizada;
- **Análise de Valor:** desenvolve uma análise de valor do sistema, através do modelo NCD e de conceitos de valor. Com a parte teórica desenvolvida, é enunciada a proposta de valor e o modelo de negócio do sistema, terminando com um estudo do método AHP para o presente projeto;
- **Análise:** apresenta um estudo do negócio do problema, através da apresentação de diferentes modelos de domínio que foram surgindo ao longo das versões do sistema. Para além do negócio é também analisado o software precedente, terminando com uma síntese das limitações encontradas;
- **Engenharia de Requisitos:** enuncia todos os requisitos funcionais para cada tipo de utilizador e os requisitos não funcionais presentes no sistema;
- **Desenho:** apresenta o desenho do sistema, através de uma representação e descrição arquitetural e detalhada;
- **Implementação:** descreve os processos tecnológicos utilizados durante a implementação, através das diferentes tecnologias utilizadas, fazendo ainda uma referência os testes realizados;
- **Resultados Obtidos:** demonstra os resultados obtidos através da apresentação das interfaces do sistema, funcionando como um pequeno manual de utilização;
- **Experiências e Avaliação:** apresenta a avaliação efetuada para o software desenvolvido, através da definição das grandezas e hipóteses a considerar, fazendo por fim uma análise das metodologias de avaliação a utilizador;
- **Sumário e trabalho futuro:** finaliza o documento com um sumário dos objetivos alcançados e não alcançados para a presente dissertação, abordando ainda as limitações encontradas e trabalho futuro, terminando com uma apreciação final.

2 Estado de Arte

O presente capítulo aborda duas secções que visam contextualizar o leitor do âmbito de desenvolvimento do projeto.

A primeira secção visa detalhar de uma forma teórica, conceitos subjacentes ao projeto desenvolvido. Serão apresentadas noções básicas de negócio de modo a facilitar a compreensão do sistema desenvolvido.

A segunda secção tem como objetivo desenvolver um estudo do estado de arte existente através de soluções ou abordagens que já existam no mercado, que tentem de uma maneira parcial ou global combater os problemas e realizar os objetivos deste projeto. Será ainda efetuada uma análise comparativa das soluções encontradas.

2.1 Enquadramento teórico

Ao longo do presente documento, é possível verificar inúmeras referências para o termo “LMS”. Este tipo de sistema, denominado de *Learning Management System*, nasceu com o desenvolvimento das técnicas de ensino eletrónico, conhecido como *e-Learning*. No entanto para além dos LMS, surgem também outros tipos de sistemas semelhantes, com acrónimos diferentes, que tornam todo o processo de definição deste tipo de plataformas bastante confuso.

Assim nesta secção são abordados e expandidos os diferentes tipos de sistemas existentes, no ensino virtual, que possam na hora de definição da plataforma eGYM como um LMS, causar alguma confusão e incerteza sobre a escolha efetuada. É ainda realizada uma pequena abordagem ao conceito de plugins.

2.1.1 Virtual Learning Environment

Um Ambiente Virtual de Aprendizagem, do inglês *Virtual Learning Environment*, representa um sistema que disponibiliza aos seus utilizadores materiais de aprendizagem digitais.

No entanto estes não tentam replicar uma sala de aula virtual, mas sim oferecer uma experiência de suporte extracurricular permitindo assim aos seus utilizadores adquirir conceitos fora de um ambiente de sala de aula, estando disponíveis em qualquer lugar e momento normalmente através de uma plataforma *web*.

Acarretam imensos benefícios para quem procura oferecer técnicas de aprendizagem digitais, possibilitando assim chegar aqueles que não podem praticar métodos de ensinos tradicionais devido a problemas geográficos ou problemas de falta de tempo.

Nos dias de hoje com o grande progresso tecnológico e com a sua associação ao Ensino, é comum que instituições de ensino possuam VLE que suportem o progresso académico dos seus alunos como é o caso do Blackboard ou do Moodle.

Estes tipos de ambientes são normalmente conhecidos como *Content Management System* ou *Learning Management System* ou até mesmo *Learning Content Management System*, entre outros.

2.1.2 Content Management System

Um Sistema de Gestão de Conteúdo, do inglês *Content Management System*, representa um sistema que oferece aos seus utilizadores a capacidade de criar, editar e apresentar conteúdo, sem que o utilizador necessite de conhecimento de linguagens HTML ou outras. Para Meerts (Meerts, 2013) um CMS:

"[...] fornece aos instrutores um conjunto de ferramentas e uma framework que permite a criação relativamente fácil do conteúdo do curso on-line e o subsequente ensino e gerenciamento desse curso, incluindo várias interações com os alunos que participam no curso¹."

Este sistema permite a utilização de conteúdos variados como textos, imagens, documentos, vídeos e outros formatos digitais, permitindo que os seus administradores tenham total autonomia no conteúdo apresentados para os restantes utilizadores.

¹ Tradução livre do autor. No original "... provides an instructor with a set of tools and a framework that allows the relatively easy creation of online course content and the subsequently teaching and management of that course including various interactions with students taking the course."

2.1.3 Learning Management System

Um Sistema de Gestão da Aprendizagem, do inglês *Learning Management System*, representa um sistema síncrono ou assíncrono que disponibiliza uma série de recursos, suportando o processo de aprendizagem dos estudantes. William R. Watson e Sunnie Lee Watson (Watson and Watson, 2007) descrevem estes sistemas como “[...] [sistemas] que oferecem funcionalidades para além do conteúdo instrucional, como gestão e rastreamento, instrução personalizada e integrações em todo o sistema [...]”².

Na sua essência um LMS procura oferecer uma plataforma digital que disponibilize conteúdos digitais aos seus utilizadores. Para além destes conteúdos digitais o sistema consegue ainda oferecer técnicas de ensino mais tradicionais, sendo assim um sistema eficaz tanto para estudantes que apenas consomem conteúdos digitais e para estudantes que integrem um tipo de ensino misto, denominado de *b-learning*³.

O sistema tem ainda a capacidade de personalizar os conteúdos oferecidos e a maneira como estes são executados. Apesar de a criação dos conteúdos não estar presente num sistema deste tipo, quando pré-carregados com conteúdos é possível personalizar quais os conteúdos a exibir aos seus utilizadores, permitindo assim algum nível de personalização e controlo dos conteúdos existentes na plataforma.

Para além da disponibilização e controlo de conteúdos, um LMS oferece ainda ferramentas que permitem as entidades correspondentes, observar relatórios e análises aprofundadas dos seus utilizadores como por exemplo: o número de utilizadores num dado módulo; as horas que um utilizador frequenta um dado módulo; taxas de sucesso ou insucesso de um dado módulo;

2.1.4 Learning Content Management System

Um Sistema de Gestão de Conteúdo de Aprendizagem, do inglês *Learning Content Management System*, é uma plataforma que permite criar, gerir, armazenar conteúdo digital. Por outras palavras o LMCS pode ser descrito como um sistema dedicado para a criação e reutilização de conteúdo digital. Watson e Watson (Watson and Watson, 2007) mencionam “O foco de um LMCS é o conteúdo, pois ‘enfrenta os desafios de criar, reutilizar, gerir e fornecer conteúdo’”⁴

² Tradução livre do autor. No original “... which offers functionality beyond instructional content such as management and tracking, personalized instruction and integration across the system ...”

³ *Blended learning* ou *b-learning* refere-se a sistemas de formação digitais onde maior parte dos conteúdos são oferecidos pela internet, no entanto o sistema inclui ainda técnicas mais tradicionais como agendamentos presenciais.

⁴ Tradução livre do autor. No original “The focus with LCMS is content as “it tackles the challenges of creating, reusing, managing, and delivering content”

Quando se analisa um LCMS é importante perceber que cada conteúdo ou material didático no sistema é entendido como uma unidade individual, permitindo assim um elevado nível de conteúdo personalizado e a sua reutilização do mesmo.

O sistema oferece ferramentas que facilitam a personalização e o compartilhamento dos conteúdos criados por toda a organização. Estas ferramentas permitem assim que os seus utilizadores criem conteúdos que melhor atendam aos requisitos dos seus utilizadores. Assim que criados, estes conteúdos podem ser distribuídos por toda a organização através de recursos de compartilhamento.

Além da criação de conteúdo, este tipo de sistema tem um ênfase na comunicação entre colaboradores através de sistemas de mensagens internas ou outras técnicas de colaborações. Criação de conteúdos em diferentes idiomas torna-se assim possível através da comunicação entre colaboradores elevando assim o sistema para uma possível dimensão internacional de cooperação.

Normalmente este tipo de sistemas são utilizados com a finalidade de produzir unicamente conteúdo extramente personalizado, podendo depois este ser vendido a diferentes organizações que procurem integrar módulos nos seus LMS.

2.1.5 Plugins

Plugins ou módulos de extensão são software complementares que permitem adicionar novas funcionalidades específicas a um dado sistema, sendo estes geralmente leves e apenas carregados quando necessários.

São normalmente criados com o intuito de permitir que *developers* terceiros consigam desenvolver novos sistemas que estendam de um outro sistema, para facilmente criar funcionalidades ou para reduzir o tamanho de uma aplicação.

Plugins de editores de texto ou leitores de vídeos são alguns exemplos de plugins bastante comuns.

2.2 Soluções existentes

Antes de efetuar uma análise de soluções existentes é necessário primeiramente efetuar uma análise do mercado, para determinar soluções que se considerem úteis ou relevantes para a presente dissertação. Como tal e utilizando diversos estudos já efetuados sobre o mercado por diversas entidades (Capterra 2019; Browne, 2019; Kotama and Saputra 2020), foram selecionados três distintos sistemas LMS que mais se assemelham ao sistema pretendido para a presente dissertação e que procuram resolver de uma maneira geral ou específica o problema descrito na secção 1.2:

- **Moodle:** para além de se considerar como o sistema mais popular globalmente, o Moodle é normalmente o sistema utilizados nas instituições de ensino superior, como é o caso do ISEP.
- **Blackboard Learn:** apontado como um dos maiores sistemas globais, este é um sistema que tem vindo a crescer consideravelmente nos últimos tempos, sendo considerada uma alternativa ao Moodle para instituições de ensino superior, como é o caso da Universidade do Minho.
- **TalentLMS:** presente nos estudos referenciados anteriormente como um dos melhores sistemas LMS existentes no mercado atual, o TalentLMS descreve-se como o sistema LMS mais intuitivo e acessível do mercado, oferecendo mesmo uma versão gratuita do produto aos seus utilizadores, sem custos adicionais.

Assim para a presente secção serão efetuadas análises em detalhe para cada uma das plataformas enunciadas anteriormente, construir uma pequena descrição assim como uma listagem das suas características chaves.

2.2.1 Moodle

O Moodle é um LMS gratuito que permite aos educadores de ensino criar a sua própria plataforma privada com diversos cursos dinâmicos que suportam a aprendizagem, a qualquer hora e em qualquer lugar (Moodle, 2020).

Primado pela sua alta customização, o Moodle apresenta-se como uma solução viável para qualquer tipo de utilizador seja este um professor, aluno ou administrador (Moodle, 2020).

Relativamente ao Moodle consideram-se as seguintes funcionalidades (Moodle, 2020):

- Gestor conveniente de ficheiros aceitando um vasto número de diferentes tipos de ficheiros, sendo possível a qualquer altura adicionar um novo tipo de ficheiro;
- Controlo de conteúdo existente possibilitando a criação de novas disciplinas e módulos dentro das mesmas, podendo estas ser inteiramente digitais ou com um tipo de ensino híbrido;
- Oferece materiais estáticos desde simples vídeos, páginas de texto, páginas *web* até materiais dinâmicos como questionários, tarefas, aulas personalizadas;
- Central de notificações permitindo notificar os seus utilizadores de novas atribuições ou de *deadlines* próximas;
- Acompanhamento de progressos e conclusões de módulos existentes, tanto para professores como para alunos, permitindo acompanhar o progresso educativo;
- Gestão de permissões e funções de utilizadores da plataforma, permitindo assim configurar diferentes tipos de acesso para diferentes utilizadores;

- Gestão de plugins, permitindo instalar e desativar consoante as necessidades;
- Gratuito sendo um sistema *open-source* sobre uma licença GPL. No entanto o sistema em *open-source* apresenta algumas limitações, podendo se tornar num *freemium* consoante as necessidades da instituição.

2.2.2 Blackboard Learn

O Blackboard Learn é um VLE e um LMS desenvolvido pela Blackboard Inc. em 2017. Tal como outros LMS, o Blackboard Learn é um sistema acessível através de uma plataforma *web* que oferece gestão de cursos e módulos, integração com sistemas de informação e autenticação (Blackboard, 2020a).

Seguindo a essência de um LMS, o Blackboard Learn foi desenvolvido com o intuito de oferecer conteúdos online ao invés das tradicionais técnicas de ensino pessoal, permitindo o ensinamento de diferentes módulos com poucas ou até nenhum contacto pessoal (Blackboard, 2020a).

“With a modern intuitive, fully responsive interface, Blackboard Learn™ delivers an unmatched learning experience. Bring learning to life with an LMS that is simple and easy to use, yet powerful, that will enable teaching and learning to happen anywhere at any time” (Blackboard, 2020a).

Podem assim ser identificadas as seguintes características (Blackboard, 2020b):

- Central de notificações, permitindo que professores notifiquem alunos a qualquer momento;
- Sistema de chat permitindo que utilizadores online em tempo real comuniquem com outros utilizadores presentes no mesmo modulo;
- Sistema de email permitindo que alunos enviem emails a professores. Permite também que professores enviem emails em massa para um grupo específico de estudantes;
- Facilidade na criação de novos módulos com diferentes conteúdos para o ensino digital;
- Personalização do conteúdo oferecido aos utilizadores através de vídeos, documentos, artigos e tarefas;
- Avaliações através de exames ou questionários;
- Ferramentas analíticas facultando assim às instituições evidências do que funciona, permitindo assim melhorar as técnicas implementadas para aumentar as taxas de sucesso dos seus utilizadores;

- Controlo de utilizadores registados, permitido configurar diferentes níveis de acesso para diferentes utilizadores.

2.2.3 TalentLMS

O TalentLMS é uma plataforma de aprendizagem virtual também conhecido por um software *cloud LMS*.

A plataforma tem como objetivo principal oferecer ferramentas de aprendizagem para os seus utilizadores, mas também oferece ferramentas de gestão como: a criação de testes, de relatórios; sistema de notificações e fóruns.

Quanto ao TalentLMS são observadas as seguintes características (TalentLMS, 2020):

- Criação de módulos através da reutilização de apresentações ou vídeos que já existam ou até mesmo através do consumo de conteúdos de páginas *web*;
- Diferentes tipos de mecanismos de avaliações, suportando múltiplos tipos de questões e respostas;
- Customização de acesso a módulos e conteúdos;
- Sistema de recompensas, onde os utilizadores podem amearhar pontos, níveis e recompensas através da realização de diferentes atividades, sendo estes pontos visíveis para outros através de *leaderboards*⁵;
- Ferramentas de comunicação através de mensagens pessoais, calendários e fóruns;
- Relatórios e análises analíticas extensivas de todo o ambiente;
- Limite de acessos e funções de utilizadores através de tipos de utilizadores personalizados;
- Integração com serviços externos desde serviços de pagamento como Paypal e Stripe a componentes externos como WordPress.

2.2.4 Análise comparativa

Como em qualquer desenvolvimento de um novo produto é sempre necessário analisar o mercado e descobrir se existe alguma solução que permita resolver todos os problemas a enfrentar no novo produto.

⁵*Leaderboards* são classificações gerais, onde os utilizadores podem ver quais são os melhores utilizadores tendo em conta uma escala específica. Para o TalentLMS a escala utilizada corresponde ao número de cursos que os utilizadores completam, apresentando assim uma classificação dos utilizadores com mais cursos completos.

Tendo em conta que a solução a desenvolver resulta de uma reengenharia de uma solução já existente, torna-se necessário que seja tida em consideração em simultâneo as características do sistema antigo e do novo sistema.

Como tal toda a comparação realizada sobre as soluções existentes baseia-se em características que procurem resolver os problemas existentes nesta nova reengenharia (cf. secção 1.2).

Representa-se assim na Tabela 1 a análise comparativa efetuada:

Tabela 1 – Análise comparativa de soluções existentes

| Características | Moodle | Blackboard Learn | TalentLMS |
|--|----------------|-------------------------|------------------|
| Configurabilidade de autorização dos seus utilizadores | ✓ | ✓ | ✓ |
| Configurabilidade de conteúdos e atividades | ✓ | ✓ | ✓ |
| Extensibilidade de conteúdos e atividades | ✓ | ✓ | ✓ |
| Sistema de progressão de conteúdos e atividades | ✓ ⁶ | × | ✓ |
| Atividades multijogador em tempo real (online) | X | X | X |

⁶ Disponível apenas com a utilização de plugins, não se encontrando disponível no sistema base.

| | | | |
|--|----------------|---|---|
| Estatística extensiva e personalizada | ✓ | ✓ | ✓ |
| Competições | ✓ ⁶ | × | × |
| Adoção de componentes externos (plugins) | ✓ | ✓ | ✓ |

Através de uma análise da Tabela 1, entende-se facilmente o facto de o Moodle ser umas das plataformas LMS mais populares do mundo e o facto de ser amplamente utilizada como a plataforma de eleição do ensino superior, mostrando-se como o sistema mais robusto tendo em conta o número de características oferecidas.

Na sua generalidade todas as soluções analisadas encontram-se preparadas para oferecer os conceitos de configurabilidade e extensibilidade de conteúdos e atividades, sendo, no entanto, isto esperado para os sistemas de ensino virtual como o LMS. Quanto ao sistema de progressão pretendido, que ofereça uma curva de aprendizagem suave, nos conteúdos oferecidos é possível observar que das três soluções estudadas apenas o Blackboard Learn não se encontra preparado para tal. É importante realçar que para o Moodle, esta característica é apenas alcançada com a utilização de plugins, sendo que esta mesma não se apresenta disponível no sistema base.

É também possível observar que mais uma vez todas as soluções se encontram preparadas para oferecer os conceitos de configurabilidade das permissões e autorizações dos seus utilizadores. Mais uma vez, esta característica é essencial nos sistemas LMS sendo por isso esperado que a mesma seja oferecida, fazendo com que a sua ausência colocasse em causa a sua classificação como um sistema deste tipo.

Quanto as atividades multijogador em tempo real, como os jogos online, é possível observar que nenhuma das soluções se encontra preparada para tal. As plataformas encontram-se preparadas para oferecer atividades como jogos, podendo mesmo vários utilizadores participar nestes jogos, no entanto nenhum destes conta com a participação em tempo real, como uma sala de jogo. Para o caso específico do Moodle, já foram propostas várias propostas de implementações para este tipo de jogos multijogador como a de Kotama (Kotama and Saputra, 2020), no entanto, até a data atual o Moodle não adotou nenhuma destas implementações para os seus plugins oficiais.

Em relação a estatística oferecida mais uma vez verifica-se que todas as soluções se encontram preparadas para oferecer aos seus utilizadores um conjunto de dados estatísticos extensos e personalizados. Esta característica é um ponto forte deste tipo de sistemas LMS, tendo em conta que são essenciais para determinar se os conteúdos e atividades criados apresentam taxas de sucesso altas e vão de encontro ao seu objetivo aquando da sua criação.

Relativamente as competências é possível mais uma vez observar que nem todas as soluções analisadas conseguem oferecer aos seus utilizadores esta característica, que se representa como uma característica de atração para o novo sistema pretendido. O Moodle apresenta-se aqui nesta situação, mais uma vez, como a única plataforma capaz de oferecer esta característica, no entanto tal como para o sistema de progressão, a característica surge mais uma vez da adoção de componentes externos oficiais.

Quanto aos componentes externos, os plugins, todas as soluções analisadas estão preparadas para adotar os mesmos. A utilização destes componentes externos é cada vez mais importante neste tipo de plataformas, permitindo cada vez mais utilizar funcionalidades não desenvolvidas de uma maneira oficial, potenciando assim o valor das mesmas.

Em suma e tendo em conta a análise efetuada a Tabela 1, é possível concluir que nenhuma das três soluções se encontra preparada para oferecer um sistema capaz de resolver todas as limitações e objetivos definidos para o sistema da presente dissertação, sendo por isso procedido à reengenharia do mesmo.

3 Análise

O presente capítulo descreve o contexto de negócio e tecnologia do projeto através da análise do respetivo negócio e do software precedente.

Numa primeira secção descreve-se o negócio através de uma perspetiva de engenharia de software, que permitirá a melhor compreensão do projeto e dos conceitos envolvidos.

Numa segunda secção procede-se à análise do software precedente, procurando analisar com mais detalhada cada uma das aplicações principais já existentes.

Por fim numa terceira e uma secção enunciam-se e detalham-se todas as limitações encontradas no sistema, tendo em conta a análise efetuada.

3.1 Análise de Negócio

Esta secção descreve e analisa o domínio do problema (i.e., negócio), capturada através de diagramas UML, e dessa forma tentando diminuir o fosso semântico entre os peritos do domínio e a equipa de desenvolvimento.

Ao longo da secção são detalhados e analisados diferentes modelos do domínio, mostrando as alterações que o negócio sofreu ao longo do desenvolvimento do sistema.

3.1.1 1ª iteração de desenvolvimento 2017/2018

Na primeira iteração ocorreu a construção do sistema de raiz, introduzindo a maioria dos conceitos de negócios existentes ao longo das seguintes iterações (Figura 4).

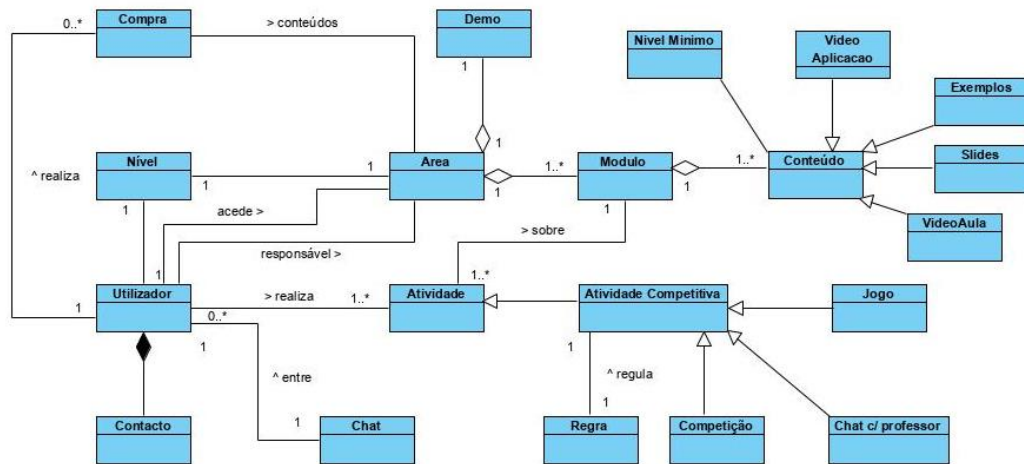


Figura 4 – Modelo de dominio da 1ª iteração

Analisando a Figura 4 é possível observar os seguintes conceitos:

- **Utilizador:** representa os utilizadores da plataforma que estão registados no sistema;
- **Contacto:** lista de utilizadores que sejam contactos de um utilizador para propósitos de comunicação;
- **Chat:** atividade de troca de mensagens entre dois utilizadores;
- **Compra:** este conceito representa a compra de conteúdos *premium*.
- **Nível:** nível de um dado utilizador, numa dada área (ex: *beginner*, *intermediate*, *expert*) representando o nível de conteúdos que o utilizador conseguia desbloquear.
- **Área:** representa uma disciplina, sendo composta por um conjunto de atividades e conteúdos;
- **Demo:** representa normalmente um vídeo que faz uma pequena demonstração das atividades e conteúdos da área;
- **Atividade:** representa um conjunto de conteúdos e atividades competitivas;
- **Atividade competitiva:** atividade que atribui pontos ao utilizador;
- **Módulo:** conceito que agrupo um conjunto de conteúdos como um capítulo de um livro;
- **Conteúdo:** representa os ficheiros e vídeos que são partilhados com os alunos para um dado módulo (vídeo-aplicação, exemplos, slides, vídeo-aulas).

3.1.3 3ª iteração de desenvolvimento 2019/2020 (atual)

Para a presente dissertação, a inclusão de novas funcionalidades e dos novos conceitos de configurabilidade e extensibilidade, obrigou a novas alterações no negócio, de modo a que o negócio consiga suportar os objetivos delineados (Figura 6).

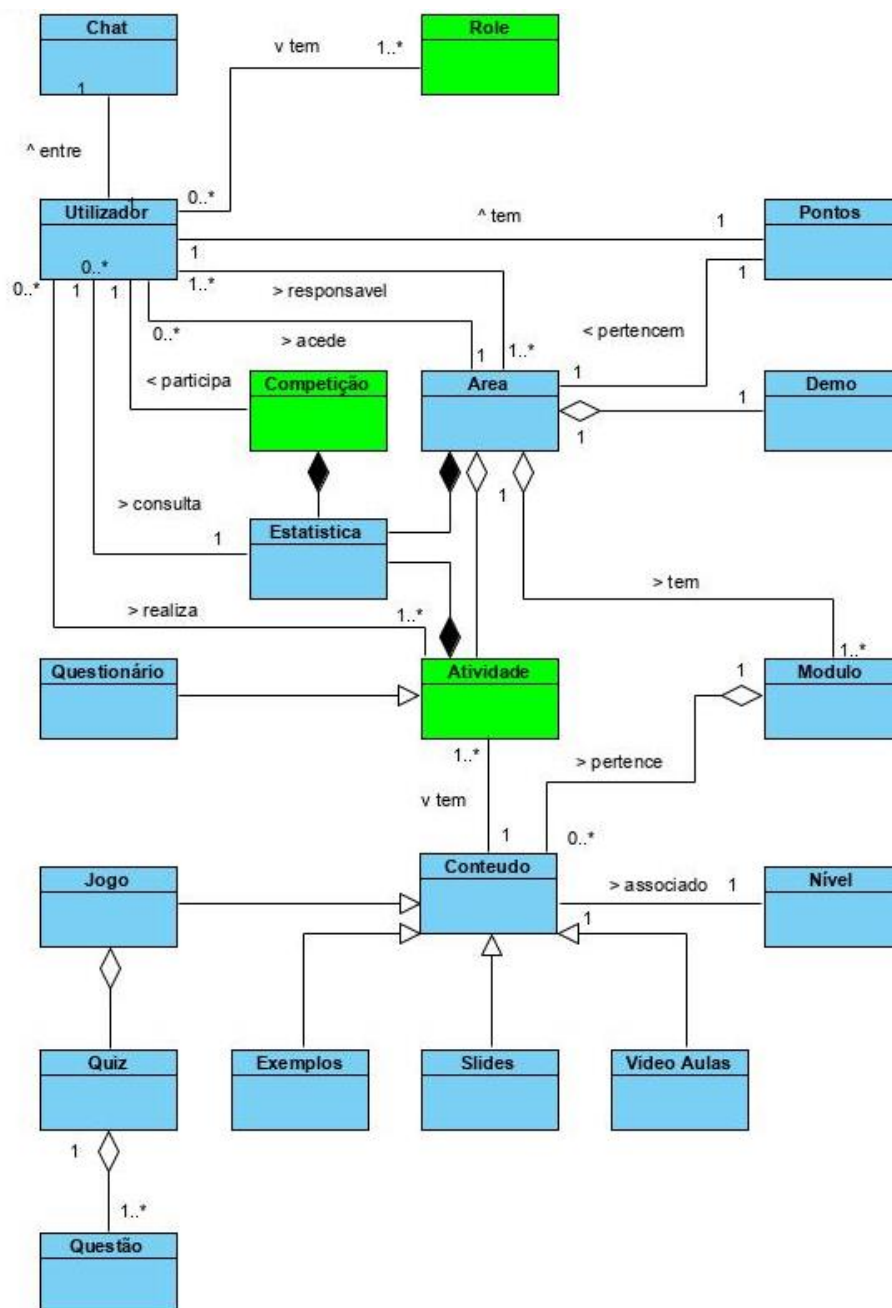


Figura 6 - Modelo de domínio para a presente dissertação Versão 1

É assim introduzido no negócio o conceito de competições. Para além das competições, todos os conteúdos existentes na plataforma relacionam-se diretamente com uma atividade permitindo assim a sua extensibilidade ao longo do sistema. Através da Figura 6 é ainda

- **Demo:** entidade removida do negócio. Considerou-se que não seria mais necessário o uso de demonstrações numa área;
- **Grupos de Acesso:** este novo conceito no negócio agrupa diversos utilizadores, sendo depois estes aplicados a uma ou mais áreas. Com esta entidade pretende-se definir um conjunto de permissões de acesso a um grupo de utilizadores;
- **Jogos multijogador:** novo conceito que representa um tipo de jogos realizados em tempo real por um grupo de utilizadores;
- **Documentos de Suporte:** representa a reunião dos conceitos de “Exemplos” e “Slides” descritos nos modelos de domínio anterior. Para este novo modelo considerou-se que a sua separação não representava de maneira correta o negócio;
- **Nível:** o nível encontra-se também agora relacionado com os jogos multijogador.

3.2 Análise do software precedente

A solução precedente encontrava-se dividida em duas plataformas: o *Back Office* e o *Front Office*.

De seguida serão analisadas ambas as plataformas em subsecções dedicadas.

3.2.1 *Front Office*

O *Front Office* representava a plataforma encarregue de disponibilizar os conteúdos didáticos para os seus utilizadores.

A plataforma estaria assim repleta de diferentes áreas, representando estas disciplinas, com um vasto número de conteúdos de ensino como módulos e jogos. No entanto estas áreas seriam apenas acessíveis aos seus utilizadores, assim que estes efetuassem um pedido de acesso que teria que posteriormente ser aceite pelos supervisores.

De modo a oferecer a melhor experiência possível aos seus utilizadores, os conteúdos de cada área estariam associados a diferentes níveis de dificuldades, oferecendo assim uma curva de progressão para que os alunos adquirissem novos conceitos de uma forma progressiva e iterativa. Para desbloquear um nível seguinte, um utilizador teria de consumir um número de conteúdos até que o seu número de pontos⁷ fosse suficiente para desbloquear conteúdos de dificuldades superiores.

⁷ Com cada atividade realizada como uma aula, a visualização de um documento ou a concretização de jogos, o utilizador recebia um número de pontos associados a uma área específica.

Um dos pontos fortes da plataforma concentrava-se na existência de um sistema de comunicação embutido na plataforma para que utilizadores conseguissem comunicar entre si de forma direta e em tempo real. O sistema de chat oferecia assim um meio de visualizar os utilizadores online num dado momento, assim como esperado de um sistema deste género guardar um histórico das mensagens partilhadas entre dois utilizadores. Estava ainda nos planos o sistema funcionar como uma ponte na comunicação entre professores e alunos para situações específicas.

3.2.2 Back Office

O *Back Office* representava a plataforma dedicada de gestão para os supervisores do sistema, sendo assim uma plataforma de acesso restrito para apenas utilizadores qualificados para tal.

É nesta plataforma que eram efetuados todos os níveis gestão relativos a conteúdos, módulos e áreas oferecendo operações básicas de *CRUD*⁸. Os supervisores poderiam assim nesta plataforma inserir novos documentos, vídeos e imagens aos seus módulos ou até mesmo novos jogos às áreas existentes para que estes sejam posteriormente consumidos pelos alunos.

Para melhorar os conteúdos oferecidos aos utilizadores da plataforma *Front Office* eram oferecidas também ferramentas de análise analítica para que os supervisores conseguissem observar possíveis tendências negativas nos métodos de estudo dos alunos. Estavam assim presentes gráficos detalhados onde poderiam ser observados, por exemplo o número de utilizadores por área ou as percentagens de sucesso em conteúdos de avaliação, entre outros.

Acrescentando as funcionalidades anteriormente descritas, a plataforma estava também reservada para o controlo de utilizadores da plataforma *Front Office*. Eram oferecidas ferramentas para controlar as áreas e módulos que um dado utilizador consegue aceder, ativar ou desativar contas de utilizadores na sua integridade, assim como a possibilidade de convidar novos utilizador para o *Back Office*.

3.3 Síntese de Limitações

Apesar de representar um sistema robusto e de oferecer um número considerável de funcionalidades aos utilizadores, o estado em que este se encontra ainda não corresponde ao idealizado na altura da sua criação. De seguida são analisadas em particular cada uma das limitações apresentadas na secção 1.2.

⁸ Acrónimo para as operações de *Create*, *Read*, *Update* e *Delete*, que traduzem respetivamente para Criar, Ler, Atualizar e Apagar.

3.3.1 Configurabilidade e extensibilidade de atividades e conteúdos

Quando se cria um sistema desta dimensão é esperado que o mesmo consiga, até um certo ponto, adaptar a novas necessidades que não tenham sido pensadas na altura do desenvolvimento do mesmo. Uma das maiores limitações encontradas no sistema prévio é a inexistência ou reduzida configurabilidade do mesmo.

Relativamente aos conteúdos, o sistema suporta a criação dos mesmo dentro de um enquadramento esperado. Por outras palavras, toda a criação de conteúdo estava definida para suportar um conjunto de parâmetros estáticos, restringindo assim de certa forma os tipos de conteúdos que a plataforma poderia oferecer aos seus utilizadores. A plataforma consegue oferecer atividades como módulos ou jogos, mas se fosse pretendido criar um questionário, o sistema já não conseguia dar suporte a tal necessidade.

Quanto a extensibilidade a maneira como o sistema anterior foi pensado e construído não permitia que conteúdos ou atividades anteriormente criadas para uma dada área fossem extensivas para outras. Seja um ensino tradicional ou digital, quando se produz um novo conteúdo é esperado que este possa ser reaproveitado em diferentes ambientes. Quando se criava por exemplo um novo *quiz* no sistema com um número de questões/respostas, este *quiz* estaria sempre associado a uma área, por exemplo Álgebra Linear, sendo impossível utilizar o mesmo questionário numa outra área que partilhe conceitos, evitando assim trabalho adicional aos supervisores.

Outra limitação na extensibilidade de conteúdos concentra-se na não reutilização de ficheiros enviados para o servidor. Nos sistemas anteriores, aquando da criação ou edição de novos conteúdos e atividades que incluíssem ficheiros de vídeo ou de PDF, estes eram novamente enviados e guardados na *cloud*⁹ mesmo que já existissem em outros conteúdos. Por outras palavras, sempre que um conteúdo fosse criado, como um módulo, os ficheiros selecionados seriam novamente guardados na *CDN*, gerando assim novos links de acesso para a sua utilização.

3.3.2 Configurabilidade de autenticação e autorização

Outra limitação do sistema anteriormente desenvolvido passa pela incapacidade de configurar os tipos de utilizadores existentes no sistema. Quando se trabalha com utilizadores é necessário poder restringir as suas ações de uma forma personalizada, através da criação de papéis que podem ser específicos ou de grupo. O sistema oferecia controlo até um certo nível, através da ativação ou desativação de contas, no entanto a sua personalização ficava aquém do esperado.

⁹ Entidade não física que representa uma rede de servidores, que potenciam o armazenamento e a gestão de dados, sem necessitar de uma única física, podendo assim ser acedido por qualquer dispositivo.

No sistema anterior os papéis de utilizadores estavam restringidos para: (i) Professor; (ii) Estudante; (iii) Administrador. Juntando ao pequeno número de papéis existentes, as suas permissões eram estáticas e muito limitada. No novo sistema é pretendido que os administradores/supervisores possam criar papéis livremente e para cada papel delinear as ações que podem efetuar no sistema.

Para além destes papéis direcionados para o *Back Office*, foi notada a inexistência de grupos de controlo para os conteúdos disponibilizados no *Front Office*. É assim pretendido neste novo sistema que os supervisores e administradores do sistema possam controlar uma maneira diferente os conteúdos e atividades disponíveis para os seus utilizadores, sem necessitar de revogar o seu acesso total à plataforma.

3.3.3 Jogos multijogador (Online)

Nos tempos iniciais da estruturação do eGYM, remontando a 2018/2019, a existência de jogos multijogador foi sempre considerada. No entanto devido a sua maior complexidade de implementação e a grande dimensão do projeto, este requerimento acabou por nunca se materializar, não passando apenas de uma possível ideia.

No ano letivo anterior, durante a segunda iteração de desenvolvimento, os alunos encarregues do seu desenvolvimento, decidiram fazer uma primeira abordagem a sua implementação. No entanto esta acabou por apresentar-se bastante limitada, muito pelo facto de não ser implementada qualquer distinção entre este tipo de jogos e os jogos didáticos. Esta inexistência da distinção acabou por negativamente influenciar toda a sua implementação, obrigando o processo a seguir certos parâmetros e conceitos, limitando de grande forma a modelação desta nova entidade.

Para além disso, o processo de emparelhamento dos utilizadores apresentava-se bastante primitivo, permitindo apenas a realização de um jogo entre dois utilizadores distintos. Pretendia-se desde a sua idealização, que este tipo de jogos pudesse ser realizado em tempo real por vários utilizadores e não apenas dois.

Quanto ao jogo em si, a inexistência de qualquer identificador de progressão dos utilizadores em tempo real, representava uma grande limitação a eliminar. É esperado que num dado jogo que decorra em tempo real, exista sempre alguma espécie de informação ou aviso do progresso dos adversários de um dado utilizador. No entanto no sistema anteriormente desenvolvido, tal não era possível verificar.

3.3.4 Competições

Uma das maiores limitações do sistema anterior passa pela inexistência de competições. Esta funcionalidade já tinha sido pensada para o primeiro ciclo de desenvolvimento do sistema, no entanto nunca acabou por ser implementada.

Por competições define-se a possibilidade de os seus utilizadores realizarem um conjunto de atividades, como jogos, de uma forma competitiva contra outros utilizadores da plataforma, num dado espaço temporal definido e em tempo real, trazendo assim alguma veia competitiva à plataforma. Espera-se ainda a existência de uma lista classificatória dos utilizadores participantes, os denominados *rankings*, podendo assim em tempo real determinar os melhores utilizadores de uma competição, tendo em consideração diversas condições como o número de pontos obtidos, o tempo utilizado, o número de tentativas, entre outras.

No contexto do sistema uma competição deve representar um tipo de atividade único, onde se pretende durante um espaço de tempo determinar um único vencedor. Considere-se que os administradores do sistema pretendem durante o período de uma semana, determinar os melhores utilizadores face a um número de questões. Para tal uma nova competição é criada, onde estes utilizadores se podem registar e assim que esta começar os utilizadores registados podem começar a responder as questões existentes. Assim que terminado o período definido, a competição termina sendo averiguado o vencedor da mesma através dos *rankings* enunciados anteriormente.

3.3.5 Estatística

Novamente a implementação de conceitos bastante primitivos nas iterações do sistema anterior, origina o levantamento de novas limitações. No caso específico, aborda-se o processo estatístico implementado.

Mais uma vez, este conceito foi idealizado e ponderado desde o nascimento da plataforma, no entanto só se procedeu a sua primeira abordagem na segunda iteração do sistema, desenvolvida em 2019/2020.

Assim o sistema anterior procurou oferecer aos supervisores e administradores do sistema alguns dados estatísticos relacionados com a plataforma do *Front Office*, entre os quais:

- Número global de utilizadores;
- Número de Áreas;
- Número de Jogos;
- Número de Módulos;
- Número de Questões.

Para um sistema tão vasto e complexo como o eGYM estes dados estatísticos oferecidos, apesar de úteis, não apresentam o nível de detalhe pretendido. Espera-se que num sistema desta dimensão para além do número das diferentes entidades existentes, seja também possível introduzir outros filtros.

Como tal para a nova iteração desenvolvida é pretendido que o *Back Office* expanda o processo de estatística existente, oferecendo dados estatísticos para todas os conteúdos e atividades existentes no sistema, assim como dados estatísticos relacionados com os seus

utilizadores registados. Para além disto é ainda esperado que sobre estes dados possam ser utilizados diversos filtros, como por exemplo a pesquisa de um dado estatístico entre duas datas seleccionadas.

3.3.6 Adoção de componentes/módulos externos

Quando se cria um sistema que oferece grande configurabilidade, existe sempre situações que por mais que a configurabilidade seja grande, esta não consiga cobrir todas as necessidades que possam vir a surgir e que não tenham sido idealizadas.

Como tal existe uma necessidade de neste novo sistema adotar a utilização de componentes externos (plugins) que permitam assim no futuro desenvolver novas funcionalidades para o sistema de uma maneira mais simples e modular.

4

4 Engenharia de Requisitos

Este capítulo tem como objetivos elicitar, sistematizar e representar os requisitos funcionais e não funcionais do projeto, processo denominado de Engenharia de Requisitos.

A primeira secção aborda os requisitos funcionais através de diagramas de casos de uso, descrevendo as funcionalidades inerentes de cada ator/papel.

A segunda secção aborda os requisitos não funcionais definidos para o desenvolvimento do projeto.

4.1 Requisitos funcionais

Requisitos funcionais representam *use cases* que descrevem, normalmente através de diagramas UML e/ou texto, a interação entre o utilizador e o sistema para uma dada tarefa.

Para o sistema desenvolvido são consideradas interações com utilizadores de diferentes níveis de permissões, fazendo com que os requisitos levantados possam apenas corresponder a um tipo de utilizador.

Como tal para melhor compreensão dos mesmos, as próximas subsecções serão reservadas para descrever os requisitos levantados tendo em conta cada tipo de utilizador existente no sistema.

Apesar de o novo sistema assentar num sistema anteriormente desenvolvido, apenas serão detalhados casos de uso que sofreram alterações na sua estrutura de modo a suportar um maior nível de configuração ou casos de uso introduzidos neste ciclo de desenvolvimento.

4.1.1 Supervisor

Um supervisor representa uma entidade administrativa do sistema com permissões de nível elevado, sendo o seu nível restrito para apenas um número finito de utilizadores. Numa fase inicial são considerados supervisores do sistema, os clientes do mesmo.

No sistema este tipo de utilizadores é responsável por toda a sua configuração. Esta configuração vai desde a gestão dos conteúdos da plataforma como dos próprios utilizadores. Para além destes, este tipo de utilizador tem também associadas funções que apenas este pode ser responsável como atribuição de novos *roles* a utilizador do *Back Office*.

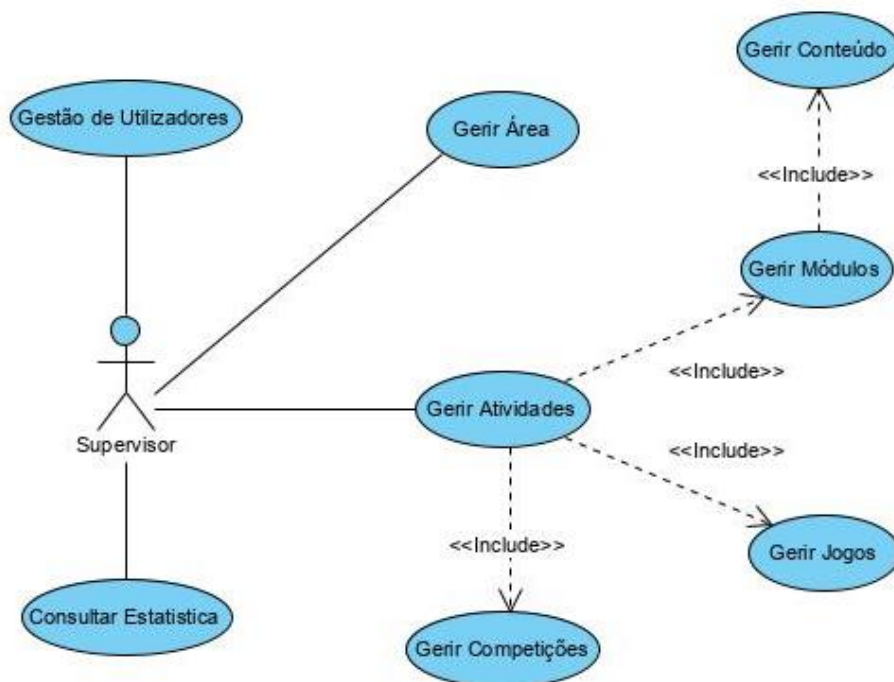


Figura 8 – Diagrama de casos de uso do supervisor

Como é possível observar na Figura 8, são representados de uma forma geral os casos de uso que envolvam algum tipo de configurabilidade como são os casos das atividades, áreas e gestão de utilizadores. Subentende-se que para todos os casos de uso, à exceção da gestão de utilizadores e consulta de estatística, a existência de operações *CRUD*.

De uma maneira mais detalhada o supervisor pode:

- **Gestão Utilizadores:** criação e atribuição de novos *roles* que condicionem o acesso dos utilizadores do *Back Office*; convidar novos utilizadores para o *Back Office*; criação e atribuição de novos grupos de acesso que condicionem o acesso dos utilizadores no *Front Office*;
- **Consultar Estatística:** análise de dados estatísticos sobre ambas as plataformas relacionadas com os utilizadores; as diferentes áreas; os diferentes conteúdos e atividades; e as competições;
- **Gerir Área:** criação de áreas; edição de áreas; eliminação de áreas; gestão da visibilidade das mesmas para os utilizadores do *Front Office*;

- **Gerir Competições:** criação de competições; edição de competições; eliminação de competições; gestão da sua visibilidade para os utilizadores do *Front Office*.
- **Gerir Jogos:** criação de jogos; edição de jogos; eliminação de jogos;
- **Gerir Módulos:** criação de módulos; edição de módulos; eliminação de módulos (aulas);
- **Gerir Conteúdo:** criação de conteúdo; edição de conteúdo; eliminação de conteúdo (vídeo-aulas, documentos de apoio, slides);

4.1.2 Utilizador do *Back Office*

Um utilizador do *Back Office* representa um tipo de utilizador com acesso a mesma, sendo este acesso apenas possível através de um convite por parte dos utilizadores supervisores do mesmo.

Este tipo de utilizador pode realizar um conjunto variável de funções, podendo apenas funcionar como um gestor de conteúdos ou um gestor de dados estatísticos ou até mesmo uma reunião destes dois últimos.

Como tal todos os seus *use cases* devem considerar a restrição anteriormente referida. É assim considerado que este utilizador pode realizar todas as funções existentes no *Back Office*, exceto as reservadas para os supervisores do mesmo. Este utilizador pode, no entanto, em qualquer altura ascender a um supervisor do mesmo, quando atribuído pelas mesmas entidades.

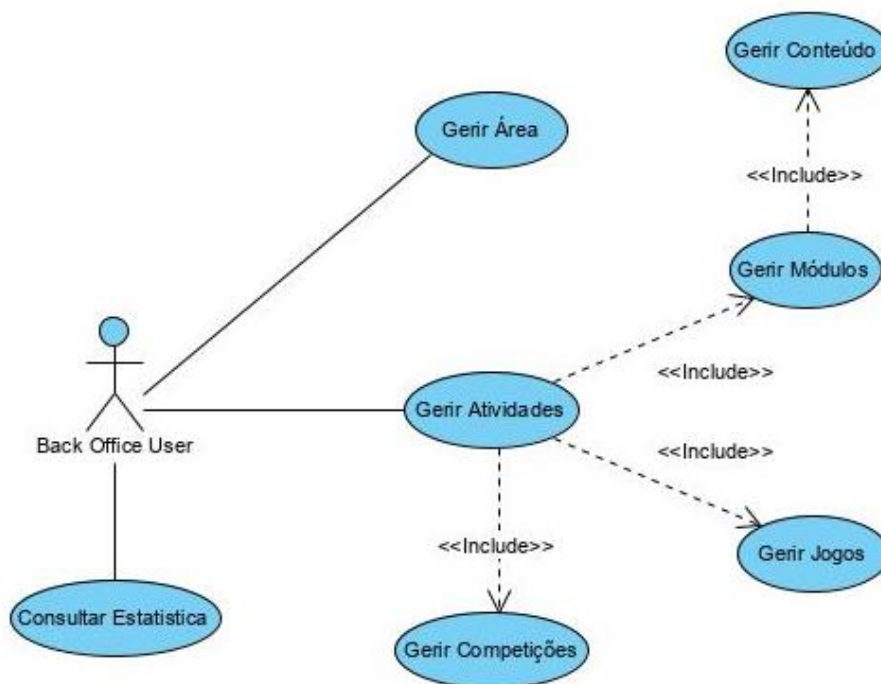


Figura 9 – Diagrama de casos de uso do gestor de conteúdos

Tendo como base a Figura 9 o utilizador do *Front Office* pode:

- **Consultar Estatística:** análise de dados estatísticos sobre ambas as plataformas relacionadas com os utilizadores; as diferentes áreas; os diferentes conteúdos e atividades; e as competições.
- **Gerir Área:** criação de áreas; edição de áreas; eliminação de áreas; gestão da visibilidade das mesmas para os utilizadores do *Front Office*.
- **Gerir Competições:** criação de competições; edição de competições; eliminação de competições; gestão da sua visibilidade para os utilizadores do *Front Office*.
- **Gerir Jogos:** criação de jogos; edição de jogos; eliminação de jogos;
- **Gerir Módulos:** criação de módulos; edição de módulos; eliminação de módulos (aulas);
- **Gerir Conteúdo:** criação de conteúdo; edição de conteúdo; eliminação de conteúdo (vídeo-aulas, documentos de apoio, slides);

4.1.3 Utilizador do *Front Office*

Um utilizador do *Front Office* representa um tipo de utilizador que tem acesso a mesma plataforma. É normalmente reservada para os utilizadores que se registem na mesma, mas os

seus *use cases* são também válidos para os utilizadores do *Back Office*, sendo que estes últimos têm acesso a ambas as plataformas.

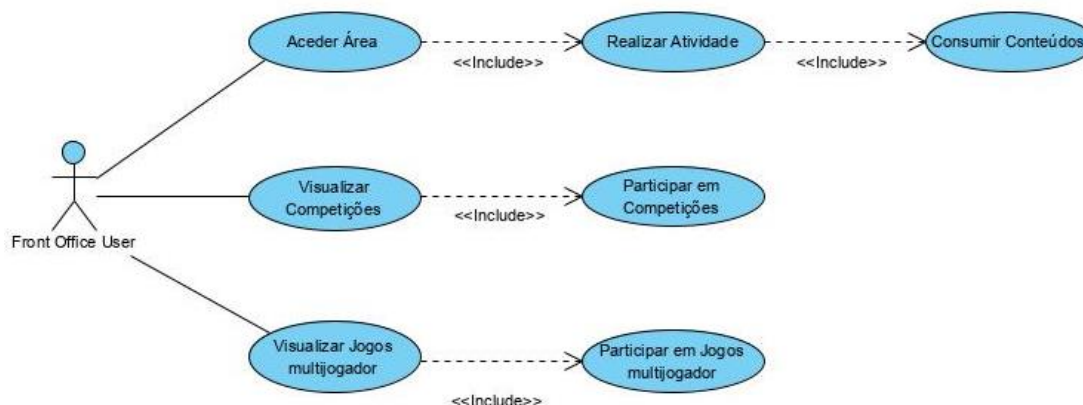


Figura 10 – Diagrama de casos de uso do Estudante

Com base na Figura 10 o utilizador do *Front Office* pode:

- **Aceder Área:** aceder a uma determinada área (disciplina);
- **Realizar Atividade:** dentro da respetiva área, o utilizador pode realizar atividades como os módulos;
- **Consumir Conteúdo:** para cada atividade o utilizador pode consumir diversos conteúdos como vídeo-aulas, documentos de suporte e jogos didáticos;
- **Visualizar Competições:** o utilizador do *Front Office* pode visualizar as competições existentes no sistema;
- **Participar Competições:** o utilizador pode participar em competições que são disponibilizadas pelos supervisores num espaço temporal limitado e definido;
- **Visualizar Jogos multijogador:** o utilizador pode visualizar os jogos multijogador disponíveis na plataforma do *Front Office*;
- **Participar em Jogos multijogador:** o utilizador pode participar em jogos multijogador, competindo contra outros utilizadores da plataforma em tempo real;

4.2 Requisitos não funcionais

Requisitos não funcionais demonstram a qualidade do sistema, apesar de não se encontrarem diretamente ligados as funcionalidades do sistema, definindo assim as propriedades e/ou restrições do mesmo.

Para a sua classificação é utilizado o modelo de classificação FURPS+. O acrónimo FURPS+ representa um conjunto de categorias de classificação sendo estas: funcionalidade, usabilidade, confiabilidade, desempenho e de suporte. O “+” foi introduzido numa evolução do modelo FURPS, introduzindo novos atributos de classificação como as restrições de design, os requisitos de implementação, interface e físicos.

Para o projeto atual, foram identificados os seguintes requisitos que se encontram representado na Tabela 21:

Tabela 2 – Requisitos não funcionais segundo o modelo de classificação FURPS+

| Classificação FURPS+ | Requisito não funcional |
|-------------------------|--|
| Usabilidade | O sistema deve suportar vários dialetos/línguas. |
| Usabilidade | A interface deve ser simples e intuitiva, de modo a que os utilizadores tenham uma melhor experiência. |
| Confiabilidade | Em caso de falha, o sistema deve recuperar o estado anterior à falha. |
| Desempenho | O sistema deve manter um tempo de resposta aceitável. |
| Suporte | O sistema deve suportar múltiplos <i>browsers</i> . |
| Suporte | O sistema deve ser modular de modo a facilitar a sua manutenção. |
| “+” Design | O sistema deve utilizar protocolos SSL. |
| “+” Implementação | O sistema deve ser implementado utilizando tecnologias gratuitas. |
| “+” Interface | O sistema deve utilizar uma <i>CDN</i> para o armazenamento dos conteúdos. |
| “+” Implantação/Físicos | O sistema deve ser implantado num servidor do DEI, ou outro sem custos financeiros. |

4.3 Resumo

Estando sistematizados e representados os requisitos funcionais e não funcionais do sistema, estes irão contribuir para direcionar o design do sistema descrito no próximo capítulo, bem como servir de validação/avaliação da solução construída.

5 Desenho

Neste capítulo apresenta-se o desenho do sistema em quatro níveis de granularidade diferentes, de acordo com a abordagem C4 (Brown, 2018):

- **Nível 1 (de contexto):** captura o sistema como um todo e se representam as interfaces disponibilizadas (seja a utilizadores ou a outros sistemas), e consumidas de outros sistemas;
- **Nível 2 (de contentores):** captura as partes do sistema que são executados independentemente uns dos outros e que se interligam através de interfaces entre eles;
- **Nível 3 (de componentes):** captura as partes (componentes) de cada um dos contentores anteriores;
- **Nível 4 (de classes/código):** captura as partes (e.g. componentes mais pequenos ou classes) que constituem os componentes identificados no nível 3;

Em cada nível, recorre-se a diversas vistas segundo o modelo 4+1 (Kruchten, 1995):

- **Vista Lógica:** representam-se as diferentes interações entre os utilizadores as aplicações que compõe o sistema;
- **Vista de Processo:** representa-se um fluxo simplificado da comunicação entre os processos que compõem o sistema;
- **Vista de Implementação:** representam-se os vários módulos que compõe o sistema;
- **Vista de Implantação:** representam-se componentes físicas que compõe o sistema;
- **Vista de componentes:** representam-se os diferentes componentes de uma aplicação;

Nem todas as vistas são adotadas e apresentadas em todos os níveis, pois nem sempre existem decisões tomadas que suportem a sua existência

As várias vistas serão representadas em diagramas UML, procurando para cada vista apresentar possíveis alternativas.

5.1 Nível 1 (de contexto)

Este primeiro nível apresenta uma granularidade mais grossa do sistema, capturando o sistema como um todo.

5.1.1 Vista Lógica

Nesta vista lógica, é representado o sistema como um único componente, representando as interfaces disponibilizadas pelo mesmo. Considera-se que todo o sistema se pode representar através de um único componente denominado eGYM.

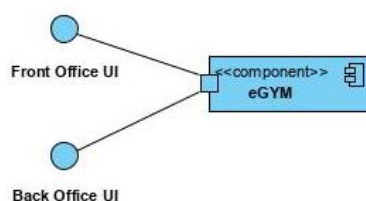


Figura 11 – Vista Lógica do contexto do sistema

5.1.2 Vista de Processo

Para a vista de processo são retratadas todas as interações do sistema como um todo, entre os seus componentes.

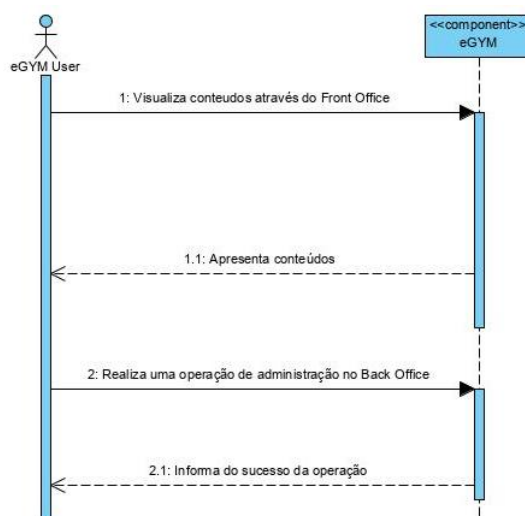


Figura 12 – Vista de Processos do contexto do sistema

5.2 Nível 2 (de contentores)

Esta secção descreve a granularidade de sistema arquitetural procurando através de diferentes vistas fazer um desenho das do sistema.

5.2.1 Vista Lógica

Na vista lógica do sistema encontram-se presentes todas as API existentes, sendo que se representam também as interfaces oferecidas pelos *Front End* de ambas as aplicações do *Front Office* e do *Back Office* para consumo dos seus utilizadores. Todas as API comunicam com um cluster MongoDB através do *Mongoose*, sendo que todas as comunicações entre API são efetuadas através de pedidos REST e por WebSocket.

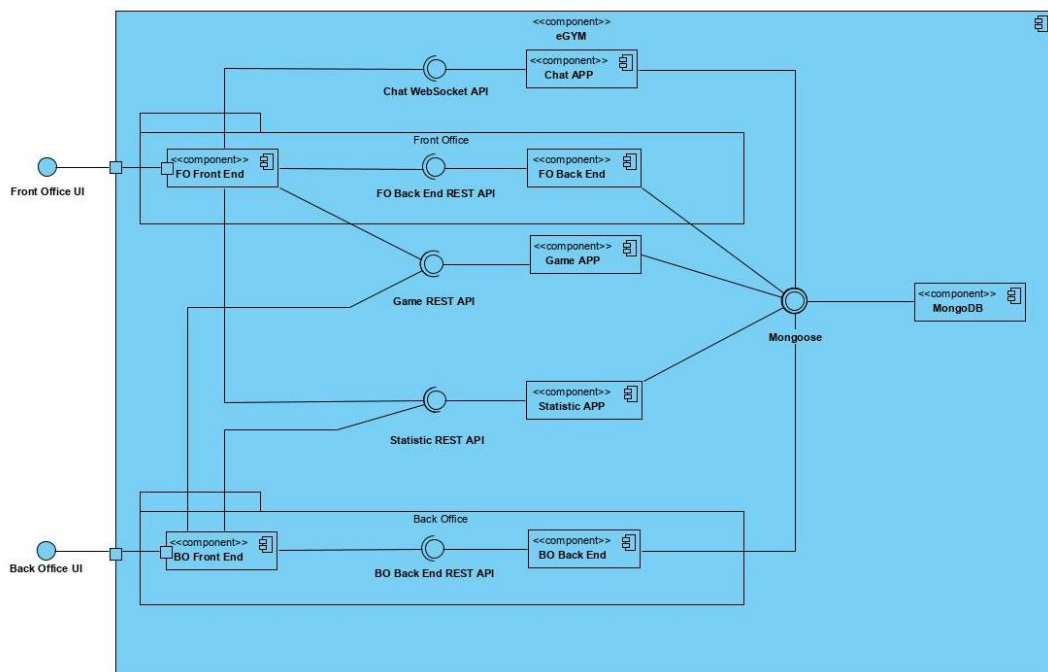


Figura 13 – Vista Lógica inicial do sistema

Numa primeira abordagem sobre a vista lógica do sistema, a vista construída está refletida na Figura 13, fazendo com que as competições ficassem implementada sobre o *Back End* do *Back Office*. No entanto, depois de realizada a análise de valor, foi possível comprovar que a introdução de uma nova API representava uma alternativa mais viável tendo em conta os critérios utilizados (cf. secção 10.1.6.7). Como tal foi desenhada a seguinte vista:

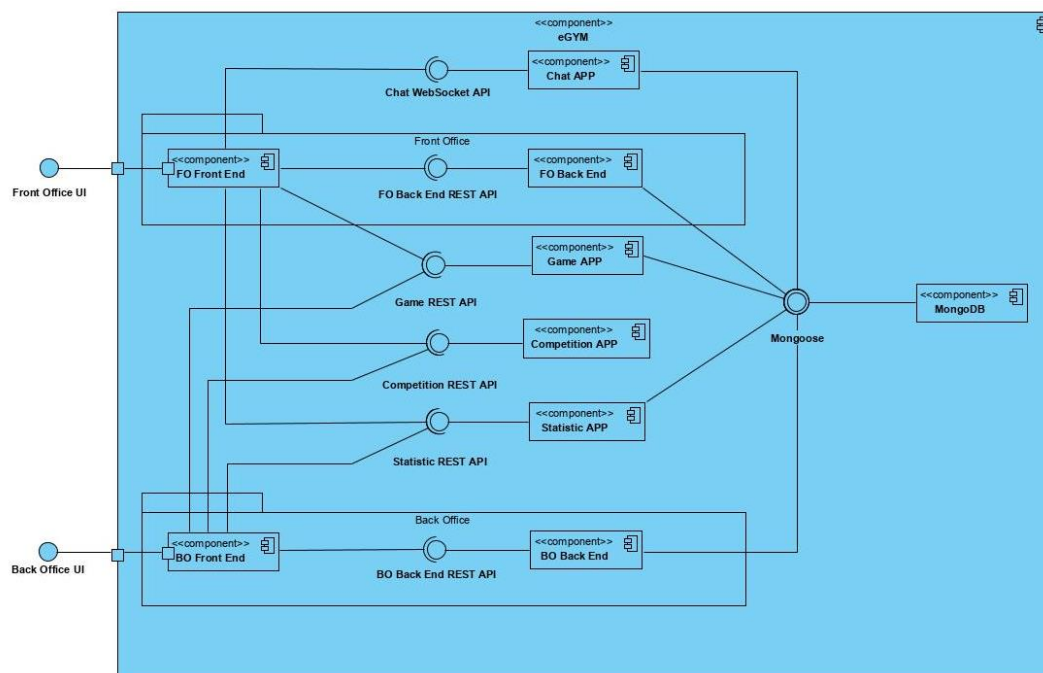


Figura 14 – Vista Lógica final do sistema

Para além da introdução da nova “*Competition API*”, é possível observar na Figura 14 a alteração nos tipos de comunicação na “*Competition API*” e na “*Game API*” com os *Front End* de ambas as plataformas. Para além de comunicação REST, existe agora comunicação através de WebSocket tendo em conta as finalidades de ambas as API.

5.2.2 Vista de Processo

Para a vista de processos são apresentadas as interações entre os diversos componentes do sistema de uma forma simples. As vistas construídas exemplificam de uma maneira simples as operações de leitura e escrita, respetivamente, entre os diferentes componentes do sistema.

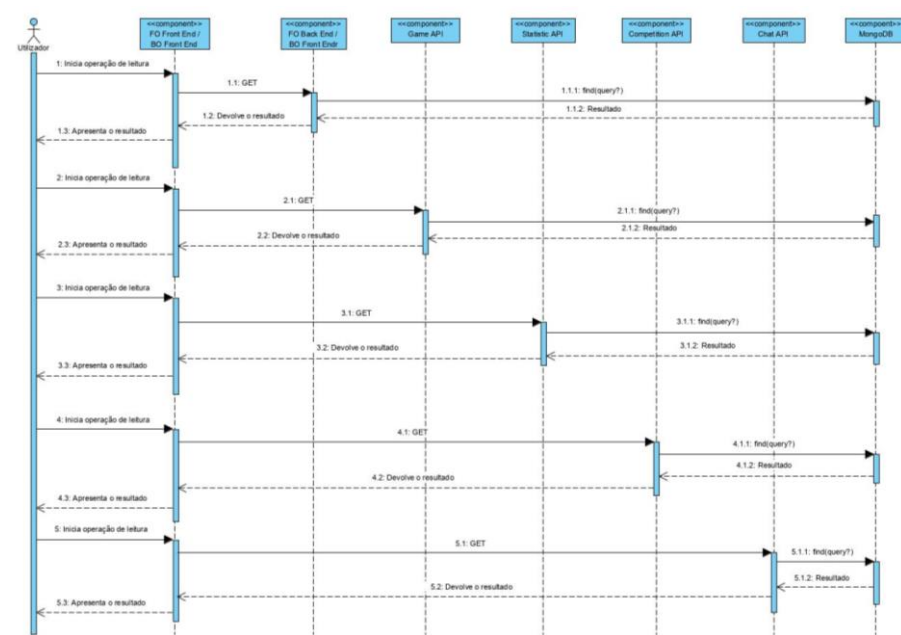


Figura 15 - Vista de Processos para operações de leitura

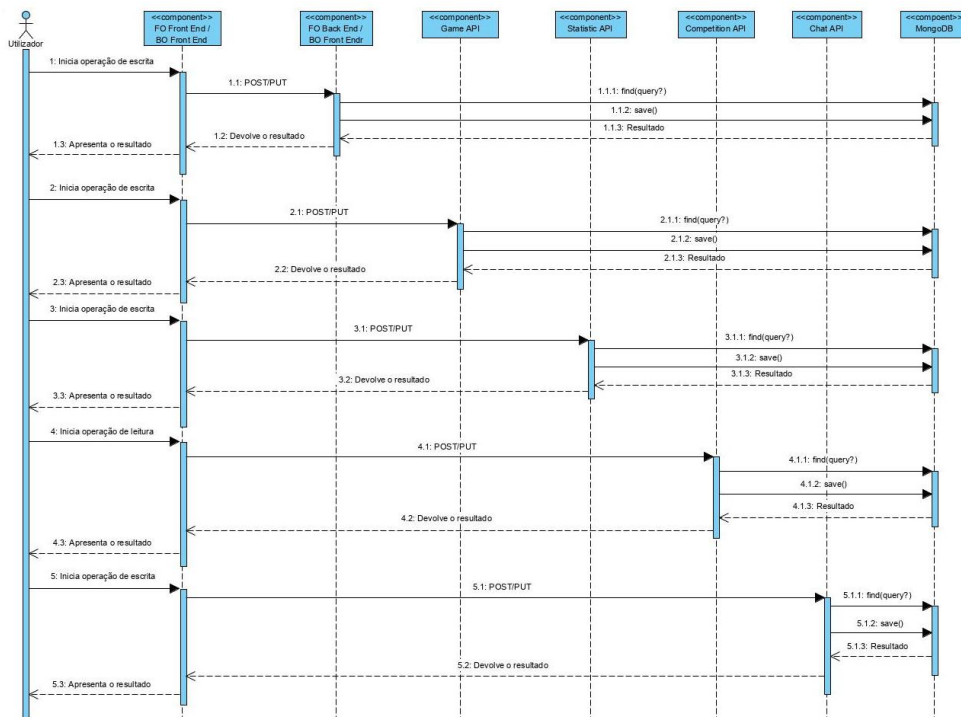


Figura 16 - Vista de Processos para operações de escrita

5.2.3 Vista de Implementação

Nesta vista são representados os módulos existentes no sistema, assim como as suas relações. É de notar que tanto o *Front Office* como o *Back Office* comunicam com as API intermedias, sendo estes identificados pelas siglas FO e BO respetivamente.

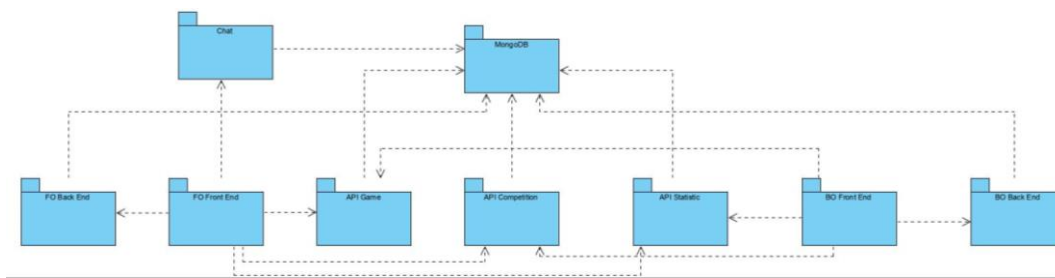


Figura 17 - Vista de implementação do sistema

5.2.4 Vista de Implantação

De seguida apresenta-se a vista de implantação do sistema que demonstra os componentes físicos do sistema:

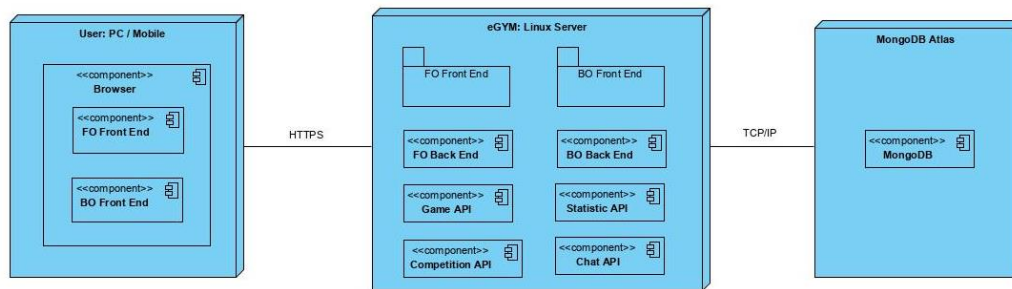


Figura 18 – Diagrama de implantação do sistema

Através da Figura 18 é possível observar que as interfaces das duas aplicações *Front Office* e *Back Office* estão disponíveis através de um *browser* sendo acedidas por um utilizador através de um computador ou dispositivo móvel. O componente MongoDB encontra-se hospedado no MongoDB Atlas, enquanto que os restantes componentes estão implantados num servidor Linux disponibilizado pelo DEI.

5.3 Nível 3 (de componentes)

Esta secção procura através de diferentes vistas detalhar o sistema como uma granularidade mais fina, fazendo assim um zoom nos componentes principais.

5.3.1 Front End

5.3.1.1 Vista Lógica

Relativamente ao *Front End* das duas aplicações do *Front Office* e *Back Office* são apresentadas estruturas semelhantes que contenham os componentes de uma aplicação Angular, através de uma vista de componentes:

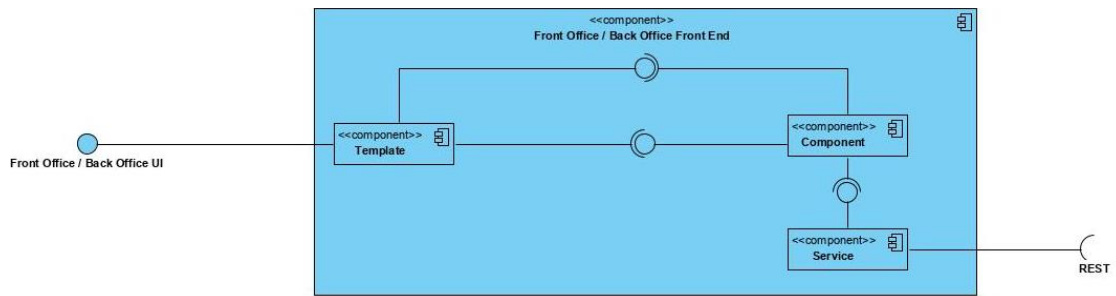


Figura 19 – Vista Lógica do *Front End* do *Front Office* e do *Back Office*

Através da análise da Figura 19 são observados os seguintes componentes:

- **Template:** corresponde ao ficheiro que constrói a interface que o utilizador consome, através de linguagens HTML e CSS;
- **Component:** representa a camada intermedia entre o *Template* e o *Service* com métodos e funções;
- **Service:** componente responsável pelas comunicações com o *Back End* através de pedidos REST através de pedidos HTTPS;

5.3.1.2 Vista de Processo

As interações entre os componentes referidos anteriormente são mais uma vez apresentadas sobre a forma de uma vista de processos. São analisadas as interações dos componentes para o acesso a uma interface e pela realização de uma operação, como a alteração de dados do perfil do utilizador.

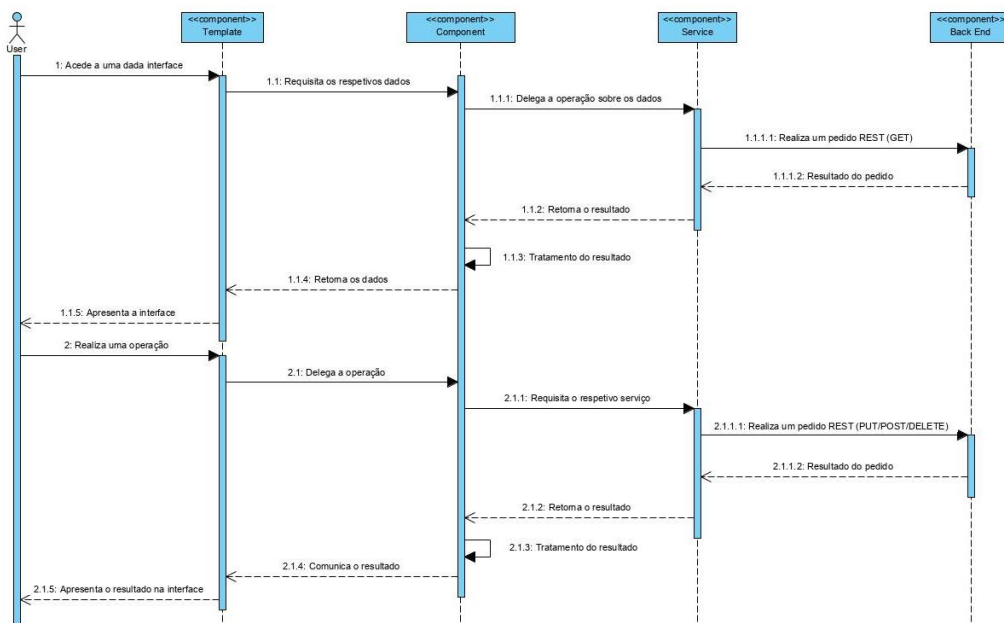


Figura 20 – Vista de Processos do *Front End* do *Front Office* e do *Back Office*

5.3.2 Back End

5.3.2.1 Vista Lógica

Quanto ao *Back End* das aplicações do *Front Office* e do *Back Office* apresentam também estruturas semelhantes, no entanto os seus componentes diferem dos apresentados na Figura 20 dado que a tecnologia utilizada para este modulo é o Node.js.

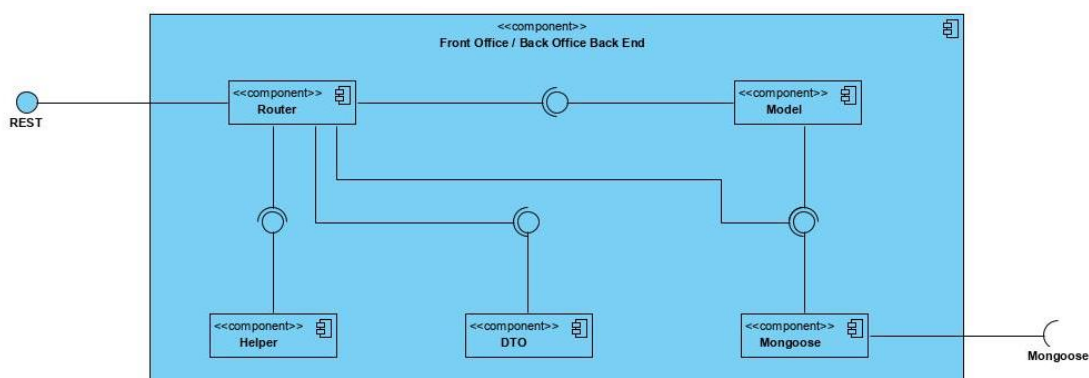


Figura 21 – Vista Lógica do Back End do Front Office e do Back Office

Através da análise da Figura 21 são observados os seguintes componentes:

- **Router:** um dos componentes mais importantes da aplicação, tendo em conta a sua função de redirecionar todos os pedidos do *Front End* para os ficheiros corretos;
- **Model:** tal como o *Front End* representa um objeto do sistema através de um ficheiro, que pode ser consumido por outros ficheiros;
- **DTO (Data Transfer Object):** componente onde se aplica uma transformação de uma entidade do sistema para um objeto mais limitado, de modo a proteger informação sensível e desnecessária;
- **Helper:** componente com funções auxiliares que são normalmente repetidas inúmeras vezes ao longo da aplicação, sendo assim concentradas neste componente;
- **Mongoose:** componente que é responsável pela comunicação com a base de dados do MongoDB.

5.3.2.2 Vista de Processo

Mais uma vez é nesta vista analisadas as interações existentes entre os diversos componentes referidos anteriormente, através de uma vista de processos. Para o *Back End* são analisados casos que se relacionem com os diferentes tipos de pedidos REST.

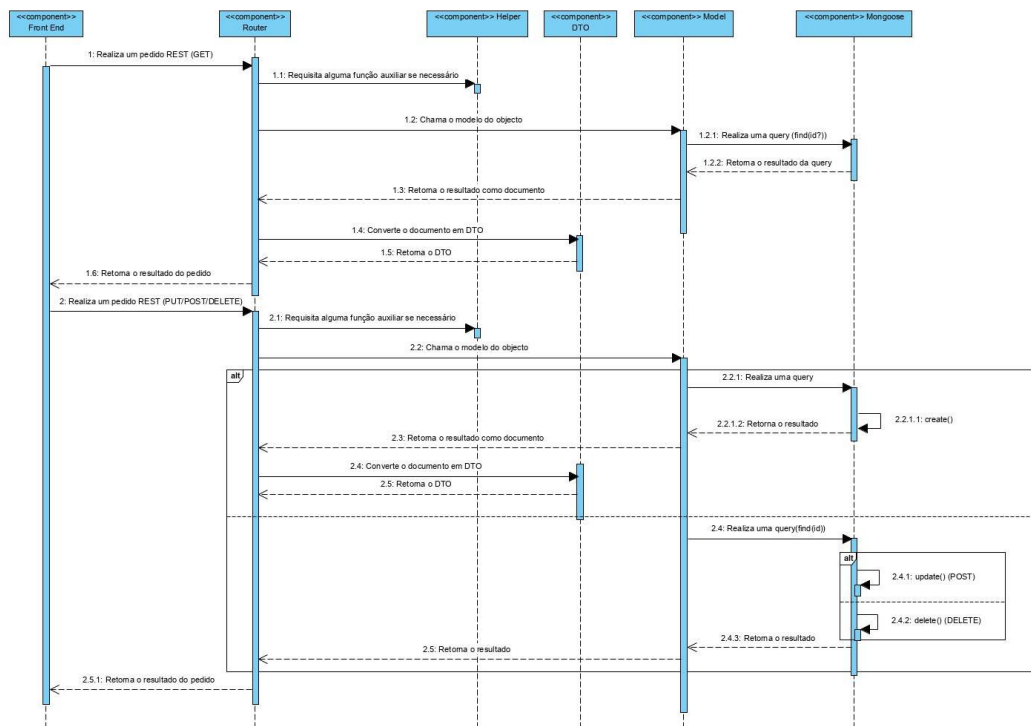


Figura 22 – Vista de Processos do *Back End* do *Front Office* e do *Back Office*

5.3.3 Competition API

5.3.3.1 Vista Lógica

Relativamente a nova API construída, a sua estrutura apresenta algumas diferenças em relação ao *Back End* do *Front Office* e do *Back Office*. A estruturação desta nova API foi desenhada de uma forma diferente, procurando retirar toda a dependência do componente *Router*. Na estrutura da Figura 21, toda a lógica é efetuada neste mesmo componente, fazendo com que a API se encontre bastante dependente do mesmo.

Para além da nova estrutura é ainda necessário introduzir novos componentes, neste caso o *Socket.IO*, devido a necessidade de um novo meio de comunicação com a API através do uso de *WebSocket*.

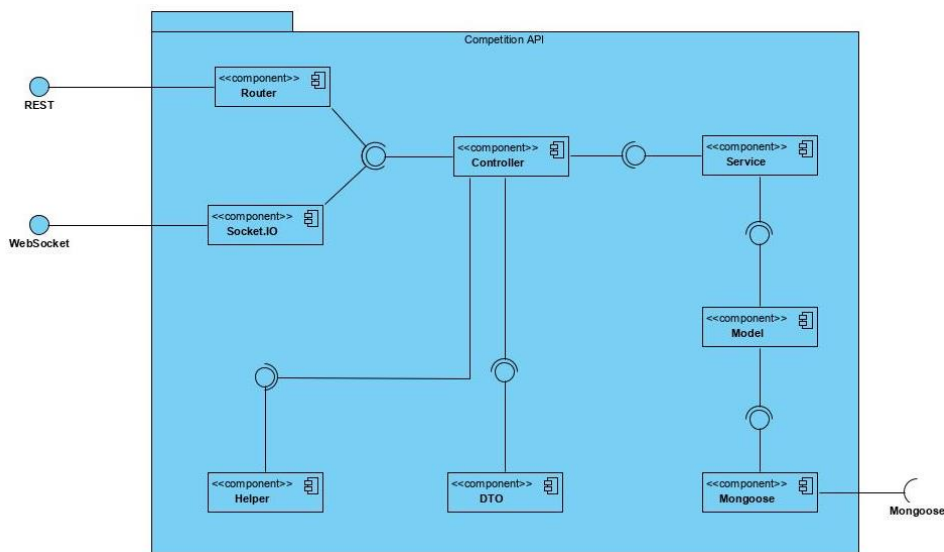


Figura 23 - Vista Lógica da *Competition API*

Assim, analisando a Figura 23 são observados os seguintes componentes:

- **Router:** componente que redireciona todos os pedidos recebidos através do uso do protocolo REST, para as respetivas funções que lidam com a lógica pretendida, existentes no *Controller*;
- **Socket.IO:** componente responsável pela interceção de pedidos recebidos através de WebSocket, redirecionando os mesmos para a função indicada, presente no *Controller*;
- **Controller:** componente essencial que direciona o fluxo de dados entre dois componentes, entre o *Router* ou o *Socket.IO* e o *Service*;
- **Service:** componente que detém todas as funções utilizadas para a ligação com o modelo de dados do sistema. A sua implementação permite a reutilização de funções evitando assim a duplicação de código;
- **Model:** componente que representa uma entidade do sistema, normalmente através de um ficheiro;
- **DTO:** componente onde se aplica uma transformação de uma entidade do sistema para um objeto mais limitado, de modo a proteger informação sensível e desnecessária;
- **Helper:** componente com funções auxiliares que são normalmente repetidas inúmeras vezes ao longo da aplicação, sendo assim concentradas neste componente
- **Mongoose:** componente que é responsável pela comunicação com a base de dados do MongoDB.

5.3.3.2 Vista de Processos

No caso desta nova API, a sua maior diferença encontra-se na utilização dos componentes *Controller* e *Service* nas interações. Como referido anteriormente estes dois novos componentes aumentaram a modularização da API e a maior reutilização de funções dentro da mesma.

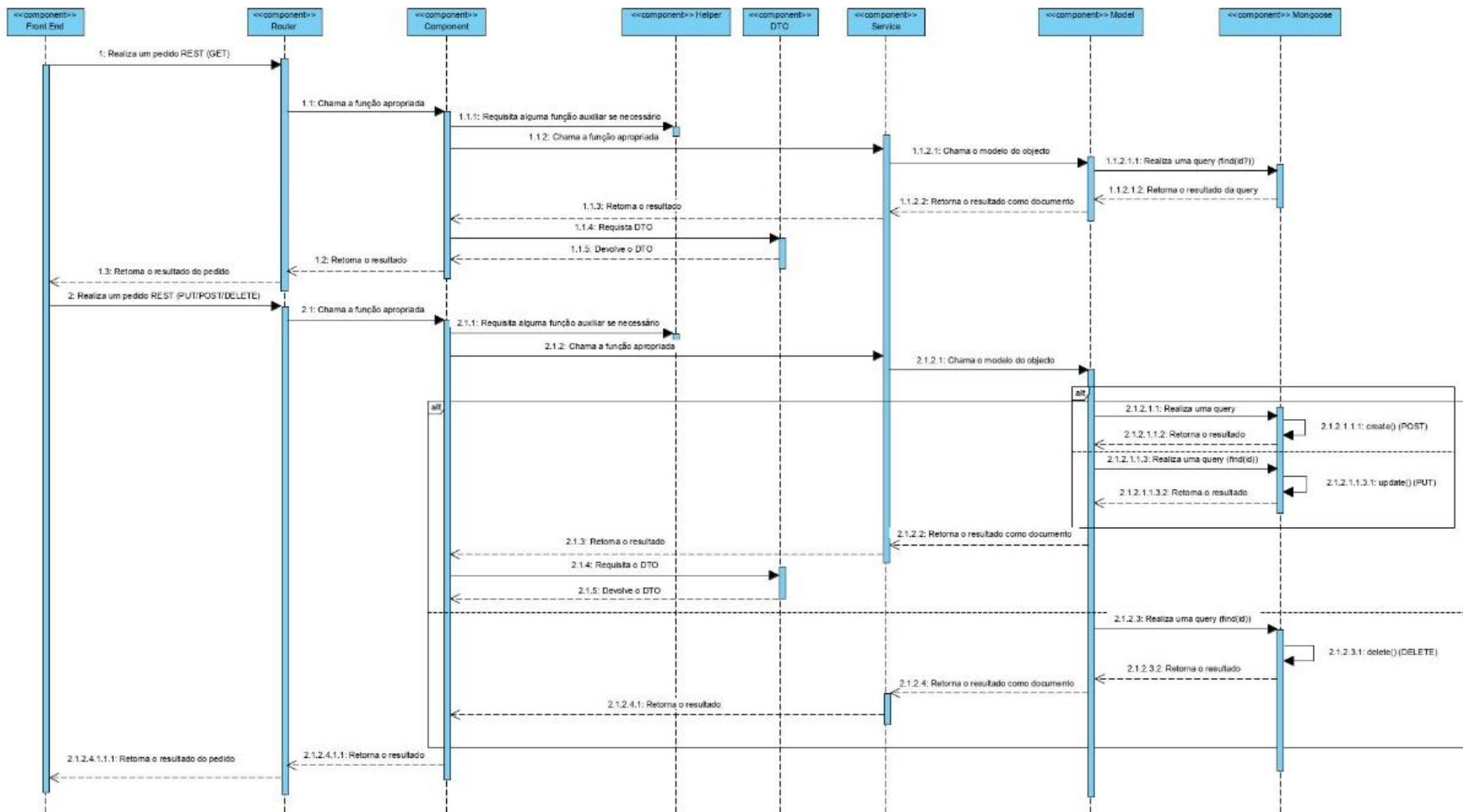


Figura 24 – Vista de Processos da *Competition* API

5.4 Nível 4 (de classes/código)

Esta secção captura pequenas partes dos componentes identificados no nível 3, dando especial detalhe ao componente da base de dados ainda não analisado.

O modelo da base de dados permite demonstrar a estrutura lógica da base de dados, e fundamentalmente determinar as entidades que são armazenadas na mesma, sendo que normalmente o modelo mais popular utilizada um formato relacional.

No entanto o eGYM utiliza uma base de dados não relacional, o MongoDB, sendo por isso utilizado um modelo diferente. Em vez das habituais tabelas que representam entidades do sistema, num modelo não relacional utilizam-se coleções que por sua vez agrupam um número infinito de documentos que simbolizam essas entidades do negócio.

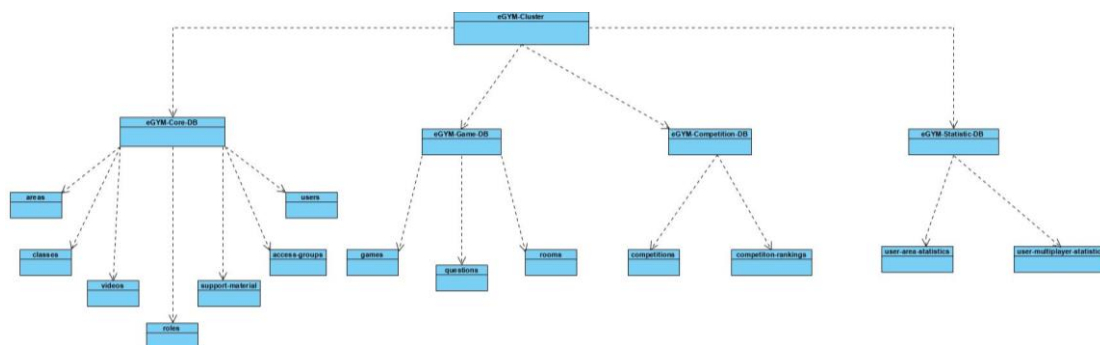


Figura 25 – Estrutura da base de dados não relacional

Através da Figura 25 é possível visualizar que o modelo da base de dados apresenta três níveis distintos:

- **Cluster:** no primeiro nível encontra-se o *cluster*. O *cluster* representa o “local” onde as bases de dados são implantadas;
- **Databases:** no segundo nível encontram-se todas as bases de dados existentes num dado *cluster*. No eGYM cada base de dados diferente tem normalmente um propósito específico associado, como a criação de uma base de dados dedicada à estatística;
- **Coleções:** no terceiro nível encontram-se as coleções. É neste local que são guardados todos os documentos, sendo que estes documentos refletem normalmente entidades do negócio. Por exemplo a coleção “users” armazena todos os documentos que representam utilizadores no sistema.

6 Implementação

No presente capítulo são abordadas técnicas de desenvolvimento utilizadas na reengenharia do sistema da presente dissertação.

Numa primeira secção procede-se à descrição tecnológica das *frameworks* e bibliotecas utilizadas que se considerem pertinentes. Para cada uma destas é efetuada uma descrição mais detalhada e técnica da mesma, utilizando alguns excertos de código que permitam demonstrar os processos utilizados tendo sempre em consideração um exemplo.

Numa segunda e última secção são demonstradas as técnicas de implementação utilizadas para a construção de testes desenvolvidos ao longo da reengenharia do sistema.

6.1 Descrição Tecnológica

Tendo em consideração a diversidade de *frameworks* e bibliotecas utilizadas na reengenharia do sistema, a presente secção encontra-se dividida em subsecções individuais, reservadas para cada *framework* e biblioteca utilizadas que se considerem pertinentes.

Na análise de cada uma destas são utilizados exemplos reais de implementações efetuadas no sistema, através do uso de excertos de código, que permitam ajudar a entender os processos tecnológicos utilizados. Não se pretende que estes processos se encontram totalmente explicados e detalhados para cada uma das *frameworks* e bibliotecas utilizadas, no entanto é esperado que os exemplos apresentados permitam de uma forma simples, demonstrar como é efetuada a implementação através do sistema.

6.1.1 Node.js

O Node.js é uma ferramenta *open-source* que permite desenvolver aplicações *JavaScript*, em que o código é interpretado em tempo real no lado do servidor. Permite ainda a criação de

aplicações escaláveis e rápidas, devido a sua capacidade de manipulação de múltiplas conexões em simultâneo. Para o sistema eGYM, todas as API desenvolvidas utilizam o Node.js como ferramenta de desenvolvimento.

Considere-se o desenho descrito para o servidor das competições, denominado de *Competition* API (cf. secção 5.3.3). Depois de receber um pedido HTTP por parte uma aplicação *Front End*, seja esta o *Front Office* ou o *Back Office*, o servidor redireciona este pedido para a sua respetiva *route* tendo em conta o URL.

```
const competition = require('./app/routes/competition');  
app.use('/api/competition', competition);
```

Figura 26 - Redireccionamento de um pedido para uma *route*, na *Competition* API

Neste ficheiro reservado para as rotas da entidade competição, procede-se à seleção do respetivo controlador e da sua respetiva função tendo em conta os parâmetros adicionais presentes no URL do pedido. Considere-se o pedido para a visualização das competições existentes no sistema, através da rota *"/*". A rota reencaminha assim este pedido para o seu controlador, *"competitionController"* e a sua respetiva função *"getCompetitions"*.

```
const competitionController = require('../controllers/competition');  
const competitionRankingController = require('../controllers/competition-ranking');  
  
// middleware to use for all requests  
router.use(function (req, res, next) {  
  next();  
});  
  
router.route('/')  
  .get(competitionController.getCompetitions)  
  .post(competitionController.createCompetition)  
  .put(competitionController.editCompetition)  
  .delete(competitionController.deleteCompetitions)  
  
router.route('/active-competitions/user/:userID')  
  .get(competitionController.getActiveCompetitions)  
  
router.route('/upcoming-competitions/user/:userID')  
  .get(competitionController.getUpcomingCompetitions)  
  
router.route('/schedule-competitions')  
  .get(competitionController.getScheduleCompetitions)  
  
router.route('/:id')  
  .get(competitionController.getCompetitionByID)  
  
router.route('/:id/register-user/:userID')  
  .put(competitionController.registerUser)  
  
router.route('/:id/ranking')  
  .get(competitionRankingController.getCompetitionRanking)  
  .post(competitionRankingController.createCompetitionRanking)  
  .put(competitionRankingController.updateUserRanking)  
  
module.exports = router;
```

Figura 27 – Rotas disponíveis para a entidade competição, na *Competition* API

Para cada função existente no controlador e para a última mencionada, existe um tratamento de dados e uma comunicação com o respetivo serviço. No caso de exemplo, o controlador requisita os dados ao serviço “*competitionService*” e a sua função “*getCompetitions*”.

```
/* Async Helper */
var asyncMiddleware = fn =>
  (req, res, next) => {
    Promise.resolve(fn(req, res, next))
      .catch(next);
  };

exports.getCompetitions = asyncMiddleware(async (req, res, next) => {
  const data = await competitionService.getCompetitions();
  return res.status(200).send({ success: true, message: 'Competitions successfully queried!', data: data })
})
```

Figura 28 - Controlador e respetiva função para o retorno das competições existentes, na *Competition* API

Por sua vez o serviço requisita toda a informação existente na base de dados através do uso dos modelos do sistema, sendo que estes modelos representam normalmente entidades de negócio. No caso de exemplo enunciado, o serviço requisita uma pesquisa de todas as competições no sistema, através da utilização da *query “find()”* disponibilizada pelo *Mongoose*.

```
const Competition = require('../models/competition');

exports.getCompetitions = async function () {
  return await Competition.find();
}
```

Figura 29 – Serviço e respetiva função para a procura de competições existentes, na *Competition* API

Cada modelo é construído através da utilização de um *schema*. Um *schema* representa uma estrutura de um documento em formato JSON, que permite definir a forma e o conteúdo de todos os documentos daquela coleção. São aqui definidos todos os parâmetros existentes numa entidade, como é o caso de uma competição, assim como o seu tipo e restrições adicionais.

```

var mongoose = require('../databases/competitionDB');
mongoose.set('debug', true);

var Schema = mongoose.Schema;
var CompetitionSchema = new Schema({
  name: {
    type: String,
    required: true
  },
  startDate: {
    type: Date,
    required: true
  },
  endDate: {
    type: Date,
    required: true
  },
  registrationStartDate: {
    type: Date
  },
  registrationEndDate: {
    type: Date
  },
  maxRegisteredUsers: {
    type: Number
  },
  accessGroups: [{
    type: mongoose.Schema.Types.ObjectId
  }],
  registeredUsers: [{
    type: mongoose.Schema.Types.ObjectId
  }],
  schedule: [{...
  }],
  createdAt: {
    type: Date,
    default: Date.now()
  }
});

module.exports = mongoose.model('Competition', CompetitionSchema);

```

Figura 30 - *Schema* de uma competição, na *Competition* API

Assim que uma específica *query* termine sobre um dado modelo, o seu fluxo de direção é efetuado no sentido inverso até chegar ao controlador, onde toda a informação pode ser formatada e trabalhada, como por exemplo a transformação de um objeto puro para um DTO. Por fim o pedido é retornado ao cliente, através de uma resposta onde se indica um código HTTP de resposta, assim como um objeto que contém uma representação do sucesso do pedido, a informação requerida e uma mensagem associada (Figura 24).

6.1.2 Angular

O Angular é uma *framework open-source* baseada na linguagem *TypeScript* para o desenvolvimento de aplicações web dinâmicas, sendo a *framework* de eleição para o desenvolvimento das plataformas de *Back Office* e *Front Office*. A sua estrutura assenta em três elementos essenciais (cf. secção 5.3.1): *template* (i); *component* (ii); *service* (iii).

Antes de atingir algum destes três elementos o Angular tem de traduzir e redirecionar as navegações de um utilizador para o seu respetivo *component*. Este tratamento é conseguido através da utilização do módulo, *Router*, onde se pode definir os *components* de cada página tendo em conta o seu URL e os seus parâmetros adicionais.

```

const routes: Routes = [
  {
    path: '', component: BasepageLayoutComponent,
    children: [
      { path: '', component: BasepageComponent },
    ]
  },
  {
    path: 'app', component: BasepageLayoutComponent,
    children: [
      { path: '', redirectTo: 'home', pathMatch: 'full' },
      { path: 'home', component: HomeComponent, canActivate: [AuthGuard] },
      { path: 'profile', component: ProfileComponent, canActivate: [AuthGuard] },
      { path: 'areas-preview', component: AllAreasComponent, canActivate: [AuthGuard] },
      { path: 'area/:id', component: AreaComponent, canActivate: [AuthGuard] },
      { path: 'area/:areaID/class/:classID/game/:gameID', component: GameComponent, canActivate: [AuthGuard] },
      { path: 'multiplayer-games-preview', component: AllGamesComponent, canActivate: [AuthGuard] },
      { path: 'multiplayer-game/:id/room/:roomID', component: GameMultiplayerComponent, canActivate: [AuthGuard], canDeactivate: [CanDeactivateGuard] },
      { path: 'competitions-preview', component: AllCompetitionsComponent, canActivate: [AuthGuard] },
      { path: 'competition/:competitionID', component: CompetitionComponent, canActivate: [AuthGuard] },
      { path: 'competition/:competitionID/:review', component: CompetitionComponent, canActivate: [AuthGuard] },
      { path: 'competition/:competitionID/question/:questionID', component: CompetitionQuestionComponent, canActivate: [AuthGuard], canDeactivate: [CanDeactivateGuard] },
      { path: 'competition/:competitionID/question/:questionID/:review', component: CompetitionQuestionComponent, canActivate: [AuthGuard], canDeactivate: [CanDeactivateGuard] }
    ]
  }
];

```

Figura 31 – Excerto de código onde se verifica a definição dos *components* para cada URL, no *Front Office*

Quando um utilizador efetua uma navegação para uma dada página, seja esta efetuada através de a introdução de um URL específico ou de um clique de navegação, o Angular carrega o seu respetivo *component*. Nesta primeira chamada são definidas todas as variáveis iniciais de utilização, que permitam o carregamento de informação para a interface do utilizador.

```

@Component({
  selector: 'app-competition',
  templateUrl: './competition.component.html',
  styleUrls: ['./competition.component.css']
})
export class CompetitionComponent implements OnInit, AfterViewInit {

  dataLoaded: boolean = false;
  competitionStarted: boolean = false;
  competitionEnded: boolean = false;

  competitionID: string;
  competition: Competition;

  competitionQuestions: Question[];
  competitionRanking: any[];

  displayedColumns =
  ['user', 'questionsAnswered', 'attempts', 'points', 'time'];
  dataSource: MatTableDataSource<any>;

  competitionTimer: number = 0;
  competitionInterval;
  ...

  currentDate: Date = new Date();
  competitionStartingDate: Date;
  competitionEndingDate: Date;

```

Figura 32 - Excerto de código com a definição das variáveis de um *component*, no *Front Office*

Assim que efetuado este processo, a interface do utilizador é atualizada com o respetivo *template*. Os *templates* podem ser caracterizados como uma forma de dizer ao Angular como

deve desenhar numa interface este *component*, sendo estes normalmente construídos através do uso linguagem HTML e CSS.

```

1 <div class="container">
2   <div class="competition-div" *ngIf="dataLoaded" style="margin: 20px">
3     <div class="competition-flex-container" fxLayout="column" fxLayoutGap="20px">
4       <div class="competition-name-info" fxFlex="" fxLayoutAlign="row">
5         <div fxLayout="column">
6           <h1>{{competition.name}}</h1>
7           {{isCompetitionRunning ? 'Time remaining' : (currentDate < competitionStartingDate) ? 'Starts in' : 'Time remaining'}}
8           <h3>{{competitionTimer | minuteSeconds: 'dd:hh:mm:ss'}}</h3>
9         </div>
10      </div>
11
12     <div class="competition-questions" fxFlex="" fxLayoutGap="20px">
13       <mat-expansion-panel *ngFor="let schedule of competition.schedule" style="width: 100%;">
14         <mat-expansion-panel-header>
15           <mat-panel-title class="expansion-panel-title">
16             <mat-icon>today</mat-icon>
17             <b>{{schedule.date | date: 'dd/MM/yyyy HH:mm:ss'}}</b>
18           </mat-panel-title>
19         </mat-expansion-panel-header>
20         <div fxFlex="" fxLayout="row" fxLayout.lt-md="column" fxLayoutGap="20px"
21           fxLayoutAlign="center center">
22           <div fxLayout="column" fxLayoutGap="20px" *ngFor="let question of schedule.questions">
23             <div>
24               <b>Question: </b><a [MathJax]=retrieveQuestion(question.questionID).questionText</a>
25             </div>
26             <div>
27               <b>Maximum Attempts: </b> {{question.maxAttempts ? question.maxAttempts : 1}}
28             </div>
29             <div>
30               <b>Available until: </b>
31               {{question.closingTime ? (question.closingTime | date: 'dd/MM/yyyy HH:mm:ss') : (competition.endDate | date: 'dd/MM/yyyy HH:mm:ss')}
32             </div>
33             <mat-divider></mat-divider>
34             <div>
35               <b>Attempts: </b> {{retrieveUserAttempts(question.questionID)}}
36             </div>
37             <div>
38               <b>Score: </b>
39               {{retrieveUserScore(question.questionID) + '/' + retrieveQuestion(question.questionID).points}}
40             </div>
41             <button mat-raised-button color="primary" style="margin-top: 20px;"
42               (click)="openQuestion(question)" [disabled]="!competitionStarted">
43               <mat-icon>visibility</mat-icon>
44               {{!competitionStarted ? 'Open Question' : questionAvailable(question) ? 'Open Question' : 'Review Question'}}
45             </button>
46           </div>
47         </div>
48       </mat-expansion-panel>
49     </div>
50

```

Figura 33 - Excerto de código do *template* para a página de uma dada competição, no *Front Office*

Qualquer interação que o utilizador efetue na interface é normalmente capturada pelo seu respetivo *component*. É nesse *component* que se procede ao tratamento da informação e dos eventos lançados pelo utilizador na interface, como por exemplo o clique no botão para a abertura de uma respetiva questão de uma competição, visualizado na linha de código 42 da Figura 33, através do evento “*click()*”.

```

/**
 * Navigates to Competition-Question component.
 *
 * Called when User clicks on "Open Question" button.
 * @param question - selected Question
 */
openQuestion(question: any) {
  if ((this.competitionEnded) || !this.questionAvailable(question)) this.router.navigate(['/app/competition/' + this.competitionID + '/question/' + question.questionID + '/review'])
  else this.router.navigate(['/app/competition/' + this.competitionID + '/question/' + question.questionID])
}

```

Figura 34 - Excerto de código de um *component* onde se valida a navegação para uma respetiva questão de uma competição, no *Front Office*

Através do uso de uma técnica denominada de *Data Binding*¹⁰, o *component* consegue também influenciar a informação oferecida ao utilizador pelo *template*. Considere-se a linha de código 42 presente na Figura 33 que controla a ativação ou desativação de um dado botão

¹⁰ Técnica que permite a propagação de alterações no *template* de forma automática através do uso de variáveis de controlo, existentes no *component*.

consoante o resultado de uma variável. Se esta variável “*competitionStarted*”, que identifica se uma dada competição já se encontra a decorrer ou não, do tipo *boolean*¹¹ se encontrar positiva o botão é ativado, caso contrário o botão é desativado impedindo assim o utilizador de abrir uma questão sem que a competição se encontre a decorrer.

Para além da manipulação dos eventos lançados pelo utilizador através da interface, o *component* comunica ainda com vários *services*. Normalmente estes *services* distinguem-se dos *components* através da sua reutilização e da sua modularidade. Idealmente os *components* devem apenas tratar da validação e da vinculação dos dados entre a aplicação e o seu utilizador, enquanto que um *service* deve oferecer serviços reutilizáveis como comunicações com servidores remotos para a requisição de dados, a validação de autenticações de utilizadores ou o registo de novas entradas para a consola.

```
export class CompetitionService {  
  
  private competitionURL = environment.competitionURL;  
  
  constructor(  
    private http: HttpClient,  
    private authenticationService: AuthenticationService  
  ) { }  
  
  /**  
   * Request Headers  
   */  
  getHeaders(params?) {  
    let headers = new HttpHeaders({  
      | 'x-access-token': this.authenticationService.userInfo.token  
    });  
  
    let httpOptions = {  
      | headers: headers  
    };  
  
    if (params) httpOptions['params'] = params;  
    return httpOptions;  
  }  
  
  /**  
   * Requests all Competitions on Database  
   */  
  getCompetitions(): Observable<Result> {  
    return this.http.get<Result>(`${this.competitionURL}/`, this.getHeaders())  
  }  
  
  /**  
   * Requests a given Competition from the Database  
   * @param competitionID - Competition ID, as string  
   */  
  getCompetitionByID(competitionID): Observable<Result> {  
    return this.http.get<Result>(`${this.competitionURL}/${competitionID}`, this.getHeaders());  
  }  
}
```

Figura 35 – Excerto de código de um *service* dedicado as competições, no *Front Office*

¹¹ Tipo de dado primitivo que possui apenas dois valores, *false* ou *true*, que representam em linguagem computacional o 0 ou 1 respetivamente.

Para utilizar estes métodos oferecidos pelo *service*, no caso anterior das competições, um *component* necessita apenas de injetar a dependência deste *service* no próprio *component*. Assim que definida, o *component* pode agora chamar qualquer método deste *service*, através do uso da variável ao qual a dependência se encontra registada.

```
constructor(private activatedRoute: ActivatedRoute, private router: Router, private competitionService: CompetitionService,
private questionService: QuestionService, private authenticationService: AuthenticationService) { }

ngOnInit(): void {
  this.activatedRoute.paramMap.subscribe(params => {
    this.competitionID = params.get('competitionID');

    const competitionInfoRequest = this.competitionService.getCompetitionByID(this.competitionID);
    const competitionRanking = this.competitionService.getCompetitionRanking(this.competitionID);
    forkJoin([competitionInfoRequest, competitionRanking]).subscribe(result => {
      if (result[0] && result[0].success && result[1] && result[1].success) {
        this.competition = result[0].data;
        this.competitionStartingDate = new Date(this.competition.startDate);
        this.competitionEndingDate = new Date(this.competition.endDate);
        if (params.get('review')) {
          this.competitionEnded = true;
          this.competitionStarted = true;
        }
      } else {
        if (this.currentDate > this.competitionStartingDate) this.competitionStarted = true;
        if (this.currentDate > this.competitionEndingDate) this.competitionEnded = true;
      }
    });
  });
}
```

Figura 36 – Excerto de código de um *component* onde se verifica a injeção de um *service* assim como a sua utilização, no *Front Office*

6.1.3 WebSocket & Socket.IO

Em sistemas onde se pretenda comunicação em tempo real com o servidor, é necessário encontrar ferramentas e protocolos de comunicação que consigam atingir estas finalidades.

Uma das soluções mais utilizadas globalmente é a utilização de WebSockets. Estes WebSockets representam um protocolo de comunicação, que fornecem canais de comunicação bidirecionais numa única conexão TCP.

Apesar de ser um protocolo que utiliza uma conexão TCP, é importante perceber que este tipo de comunicação é diferente de uma comunicação HTTP. No entanto todos os pedidos WebSocket são recebidos e interpretados por servidores HTTP, necessitando apenas de comunicar ao servidor a necessidade de realizar um *upgrade* na comunicação de HTTP para WebSocket no seu cabeçalho.

Tal como o HTTP, o WebSocket oferece dois tipos distintos de comunicação através da sua especificação URL: ws (WebSocket) e wss (WebSocket Secure) que são utilizadas respetivamente para comunicações descriptadas e encriptadas.

Sendo necessário efetuar um tratamento dos dados recebidos no lado do servidor e tendo em conta que os servidores do sistema eGYM se encontram desenvolvidos numa tecnologia Node.js (cf. secção 6.1.1), que assenta numa linguagem JavaScript, foi necessário encontrar uma biblioteca que permitisse a utilização destas comunicações em tempo real nos servidores.

Uma solução bastante popular é o Socket.IO. O Socket.IO é uma biblioteca *JavaScript* para aplicações web, que permite tal como os WebSockets a comunicação bidirecional entre um cliente e um servidor, sendo esta é tanto executada no lado do servidor como no lado do cliente.

Apesar da sua implementação funcionar como um “encapsulamento” da biblioteca WebSocket, é importante realçar que uma ligação Socket.IO não pode estabelecer ligação com uma simples ligação WebSocket, nem vice-versa. No entanto, se necessário, o Socket.IO permite que uma comunicação seja automaticamente atualizada em qualquer momento para uma simples conexão WebSocket, como por exemplo em navegadores web mais antigos que não suportem a utilização de Socket.IO.

Toda a comunicação entre os clientes e o servidor é efetuada através de uma arquitetura orientada a eventos. Sendo assim sempre que necessário, um cliente ou servidor emite um evento com um nome específico e um conjunto de dados, e do outro lado o servidor ou um cliente consegue capturar esse evento específico através do seu nome.

```
/** Socket.IO connect */
connect(user: any): void {
  this.socket = io(this.socketIORUL, { autoConnect: false, query: { userID: user.id, userName: user.firstName + ' ' + user.lastName } });
  this.socket.open();
}

/** Socket.IO listen */
listen(eventName: string) {
  return new Observable((subscriber) => {
    this.socket.on(eventName, (data) => {
      // console.log("received event", this.socketIORUL, eventName, data)
      subscriber.next(data);
    })
  })
}

/** Socket.IO emit event */
emit(eventName: string, data: any) {
  // console.log("emitting event", this.socketIORUL, eventName, data);
  this.socket.emit(eventName, data);
}

/** Socket.IO disconnect */
disconnect(): void {
  this.socket.disconnect();
}
```

Figura 37 – Serviço Socket.IO, no *Front Office*

Uma das utilizações do Socket.IO na reengenharia do sistema eGYM, acontece na introdução dos jogos multijogador no sistema. Como referido anteriormente (cf. secção 3.3.3), é esperado que o sistema permita que um número de utilizadores distintos se possam conectar a um dado jogo em tempo real e competir entre si.

Considere-se o emparelhamento de diferentes utilizadores numa sala de jogo. Quando um utilizador pretende participar num jogo multijogador, abre uma nova conexão com servidor através do uso de WebSocket e Socket.IO, lançando de seguida um novo evento “*searchGameRoom*”.

```

searchOpponent(): void {
  this.wsService.connect(this.data.user);
  this.wsService.emit('searchGameRoom', {
    'userID': this.data.user.id,
    'userName': this.data.user.firstName + ' ' + this.data.user.lastName, 'gameID': this.data.game._id
  });
  this.searchingOpponent = true;
}

```

Figura 38 - Excerto de código para uma pesquisa de sala de jogo, no *Front Office*

O servidor assim que capturar este evento, procede a introdução do utilizador num algoritmo personalizado para o emparelhamento de utilizadores que procurem o mesmo jogo.

```

* @param {*} socket - user socket
* @param {*} gameID - game ID
*/
var findGameRoom = async function (socket, gameID) {
  const filteredGameRooms = queue.filter(x => x.gameID === gameID);
  if (filteredGameRooms && filteredGameRooms.length > 0) {
    // Game Room for the given Game ID already exists, so we need to check if the User can join
    if (filteredGameRooms[0].users.length === MAX_GAME_PLAYERS - 1) {
      // If this User is added to the Game Room, we need to check if the room will be full afterwards, hence the -1
      const room = filteredGameRooms[0].room;
      socket.join(room);
      rooms[socket.id] = room;

      // Retrieve current Game Room User list
      let users = [];
      users.push(socketUsers[socket.id]);
      filteredGameRooms[0].users.forEach((elementSocket, index) => {
        users.push(socketUsers[socket.id])
      })

      // Remove this specific Game Room from the Queue
      console.log("Game Room was full, removed from the Queue!");
      queue = queue.filter(x => x.gameID !== gameID);
      return createRoomForUsers(gameID, users);
    } else {
      // Game Room is up but not full, so we add this User socket
      const room = filteredGameRooms[0].room;
      socket.join(room);
      rooms[socket.id] = room;
      console.log("New User added to an existing Game Room!");

      // Retrieve current Game Room User list
      filteredGameRooms[0].users.push(socket);
      let users = [];
      filteredGameRooms[0].users.forEach(element => {
        users.push(socketUsers[element.id]);
      })
      return users;
    }
  } else {
    // First User to queue for this specific Game, so a new Game Room is created
    const roomString = crypto.randomBytes(20).toString('hex');
    socket.join(roomString);
    rooms[socket.id] = roomString;

    queue.push({ users: [socket], gameID: gameID, room: roomString });
    console.log("New queue created!");
    return [socketUsers[socket.id]];
  }
}

```

Figura 39 – Algoritmo de emparelhamento de jogadores, na *Game API*

Se o seu emparelhamento for bem-sucedido, o servidor emite um novo evento para o *Front Office* indicando a sala de jogo em que o utilizador se encontra, assim como os seus utilizadores.

```

socket.on('searchGameRoom', async (data) => {
  // Calls a matchmaking algorithm that searches for an appropriated Game Room. If none is found User is placed on the Queue and creates a new Room
  const gameRoom = await findGameRoom(socket, data.gameID);
  if (gameRoom) {
    // If a Game Room already exists, all current members are notified that a new User has joined
    const room = rooms[socket.id];
    if (room) {
      SocketIo.in(room).emit('userJoined', { users: gameRoom })
    }
  }
})

```

Figura 40 - Excerto de código para a captura de pedidos de emparelhamento, na *Game API*

O *Front Office* assim que capturar este evento, procede ao seu tratamento de dados e atualiza visualmente a lista de jogadores da sala de jogo, reproduzindo ainda um som simbólico que permita alertar os seus utilizadores da entrada de um novo utilizador.

```

// Server event captured, when an User joins the Game Room that this User currently belongs to
this.wsService.listen('userJoined').subscribe((res: any) => {
  res.users.filter(x => x.userID !== this.currentUser.id).forEach(element => {
    this.multiplayerGameHelper.push({ userID: element.userID, userName: element.userName, questionsAnswered: 0, ready: false, })
  });
  this.noChangesTimer = 0;
  if (this.multiplayerGameHelper.length > 1) {
    this.roomChangesSound.play();
  }
})

```

Figura 41 – Excerto de código para a captura do evento da entrada de um novo utilizador na sala de jogo, no *Front Office*

O utilizador pode ainda em qualquer momento abandonar uma sala de jogo, através do botão apropriado. Se assim acontecer a plataforma do *Front Office* emite um novo evento para o servidor, denominado de “*leaveGameRoom*”, identificando o utilizador que acabou de abandonar a sala, procedido de uma desconexão da comunicação Socket.IO com o servidor.

```

cancelSearch(): void {
  this.searchingOpponent = false;
  this.wsService.emit('leaveGameRoom', { 'userID': this.currentUser.id, 'gameID': this.data.game_id })
  this.wsService.disconnect();
}

```

Figura 42 - Excerto de código para o lançamento do evento aquando do abandono de uma sala de jogo, no *Front Office*

Assim que o servidor capturar este novo evento, “*leaveGameRoom*”, o servidor remove o utilizador da sua respetiva sala de jogo, notificando prontamente todos os utilizadores que se encontravam na mesma sala para além do utilizador que acabou de sair.

```

socket.on('leaveGameRoom', async (data) => {
  console.log('User ' + data.userID + ' exited from a Game Room');
  // Removes User from the Game Room
  const temp = await removeUserFromGameRoom(socket);
  if (temp) {
    // Notifies other Users in the same Game Room about it
    SocketIo.in(temp.room).emit('userLeft', { users: temp.users });
  }
})

```

Figura 43 - Excerto de código para a captura do evento de abandono de uma sala de jogo, na *Game API*

A plataforma do *Front Office* assim que receber este novo evento, “*userLeft*”, simplesmente procede a atualização da listagem de jogadores numa sala de jogo. Esta atualização visual é também acompanhada de um efeito sonoro, permitindo assim notificar os utilizadores que não tenham a página do *Front Office* como foco principal.

```
this.wsService.listen('userLeft').subscribe((res: any) => {
  this.multiplayerGameHelper = this.multiplayerGameHelper.filter(x => res.users.map(y => y.userID).includes(x.userID));
  this.roomChangesSound.play();
});
```

Figura 44 – Excerto de código para a atualização dos utilizadores numa sala de jogo depois de algum abandonar, no *Front Office*

Assim que o jogo se inicie, o servidor emite o evento “*gameRoomReady*” fornecendo notificando assim os seus utilizadores do início do jogo.

```
socket.on('createGameRoom', async (data) => {
  // Game Room creation triggered since its full or Users agreed to start without being full
  const room = rooms[socket.id];
  const gameRoom = await createRoomForUsers(data.gameID, data.users.map(x => ({ userID: x })));
  if (gameRoom) {
    SocketIo.in(room).emit('gameRoomReady', { room: gameRoom });
  }
});
```

Figura 45 – Excerto de código que representa o início de um jogo multijogador, na *Game API*

Quando capturar este evento, o *Front Office* termina o processo de emparelhamento, guardando a informação da sala de jogo recebida do servidor, dando assim início ao jogo multijogador.

```
// Server event captured, when the Online Game is ready and about to start
this.wsService.listen('gameRoomReady').subscribe((res: any) => {
  this.gameRoom = res.room;
  clearInterval(this.searchTimer)
  setTimeout(() => {
    this.dialogRef.close(res.room);
  }, 2000)
});
```

Figura 46 – Excerto de código que demonstra a captura do evento que marca o início de um jogo multijogador, no *Front Office*

6.1.4 *HttpInterceptor (Caching)*

Sendo esperado a utilização do sistema por diversos utilizadores num mesmo período e considerando o número extensivo de pedidos necessários a efetuar aos servidores para cada utilizador, surge a necessidade de procurar soluções que tentem evitar ao máximo o sobre carregamento e por consequente a falha dos servidores. É fácil de compreender que esta sobrecarga dos servidores tem um crescimento exponencial, com o crescimento do número de utilizadores que utilizem a plataforma num preciso momento.

Uma das soluções encontrada e implementada, apresentando um caracter mais técnico, foi a introdução de *caching* . *Caching* representa o processo de guardar um subconjunto de dados, numa camada de armazenamento físico denominada de *cache* . Estes dados guardados nesta camada de armazenamento podem ser acedidos em altas velocidades, permitindo assim que os pedidos efetuados pelos utilizadores sejam respondidos o mais rapidamente possível.

A introdução de *caching* num sistema deste género oferece inúmeros benefícios entre os quais:

- **Performance** : evitando a leitura de dados do servidor, neste caso das bases de dados, o acesso ao mesmo torna-se infinitamente mais rápido, melhorando assim a *performance* do sistema.
- **Redução de carga no Back End** : com maior parte dos dados guardados em *cache* , existe uma menor necessidade de efetuar pedidos ao *Back End* .
- **Redução de custos** : em sistemas de larga escala, como é o caso, a introdução de *caching* permite aos seus administradores reduzirem custos monetários em relação aos servidores.

Um método bastante popular de implementação de *caching* com a *framework* utilizada no desenvolvimento do sistema, o *Angular* , são os **HttpInterceptors** . Estes **HttpInterceptors** como o termo indicam são intercetores de pedidos HTTP, fazendo com que todos os pedidos que o sistema efetue ao seu *Back End* passe por este mesmo antes de chegar ao servidor.

Obviamente que não é esperado que o sistema consiga oferecer *caching* de pedidos que tenham o objetivo de atualizar ou inserir dados na base de dados. São assim filtrados todos os pedidos **GET** ¹² que o sistema execute, sendo os restantes ignorados e redirecionados para os seus normais fluxos de ação.

```
constructor(private cacheService: HttpCacheService, private notificationSnackbarService: NotificationSnackbarService) {  
  }  
  
  intercept(request: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {  
    // Ignores not GET requests  
    if (request.method !== "GET") {  
      return next.handle(request);  
    }  
  }  
}
```

Figura 47 - Excerto de código de um *HttpInterceptor* , no *Front Office*

Quando perante um pedido **GET** ¹² o intercetor procura na *web cache* pela existência desse mesmo pedido. Caso o pedido existe em *cache* , o seu resultado é rapidamente acedido e retornado para o utilizador. Em caso contrário o pedido não tem outra hipótese que não seguir o seu fluxo original para o respetivo servidor, no entanto assim que retornado é imediatamente adicionado a *cache* , permitindo assim que futuros pedidos semelhantes utilizem esta informação temporária.

¹² Denominação para um tipo de pedido HTTP que pretende requisitar ou ler dados de uma base de dados.

```

// Verifies if a given request has a cached response. This search is based on the request URL params
const cachedResponse: HttpResponse<any> = this.cacheService.cache.get(request.urlWithParams);
if (cachedResponse) {
  // Returns the cached response when it exists
  return of(cachedResponse);
}

// If no cached response exists, the requests is forwarded
return next.handle(request).pipe(
  tap(event => {
    if (event instanceof HttpResponse) {
      // Cache this response when a response is received
      this.cacheService.cache.set(request.urlWithParams, event);
    }
  })
)

```

Figura 48 - Excerto de código de uma verificação na *cache* pelo pedido efetuado num *HttpInterceptor*, no *Front Office*

No entanto a informação guardada em *caching* deve ser temporária e não persistente. O eGYM como plataforma de conteúdos didáticos, encontra-se em constante alteração desde a introdução de novos conteúdos até a atualização de conteúdos existentes. Como tal, a persistência de dados em *caching* seria um grande problema para a plataforma, podendo “bloquear” utilizadores de observar novas alterações nos conteúdos e atividades.

Para resolver esta situação, toda a *cache* existente no sistema é eliminada ao fim de dez segundos, permitindo assim que uma resposta de um dado pedido esteja disponível durante dez segundos para ser acedida através da *cache*. Assim que este período expirar, a resposta é existente na *cache* é atualizada com a nova resposta recebida pelo servidor.

```

export class HttpCacheService {
  // Cache Map <Request URL, Response>
  cache = new Map<string, any>();

  // Clears cache after 10sec
  constructor() {
    setInterval(() => {
      this.cache.clear();
    }, 10000)
  }
}

```

Figura 49 - Excerto de código com o *reset* da *cache* ao fim de dez segundos, no *Front Office*

Para pedidos que se consideram extremamente importantes, ou que se saiba à priori da sua alteração de dados é possível adicionar uma *tag* ao cabeçalho do pedido, denominado de “*reset-cache*”, forçando assim caso exista a eliminação da *cache* existente de um dado pedido, obtendo como tal a informação diretamente do servidor.

```

if (request.headers.get('reset-cache')) {
  if (this.cacheService.cache.get(request.urlWithParams)) {
    this.cacheService.cache.delete(request.urlWithParams);
  }
}

```

Figura 50 - Excerto de código com o pedido de eliminação de *cache* para um dado pedido, no *Front Office*

Esta técnica de *caching* encontra-se implementada na sua totalidade apenas no *Front Office*. Apesar de o *Back Office* estar preparado para tal, se assim se pretender num futuro próximo, considerou-se que por enquanto a implementação de *caching* não se alinhava com a finalidade do *Back Office*. O *Front Office* por outro lado, com um número muito superior de utilizadores concorrentes esperados, assim como a sua maior diversidade e o maior número de pedidos paralelos necessários, tornou-se assim no candidato ideal para a implementação desta solução.

6.1.5 Localização & Internacionalização (i18n)

Tendo em consideração a finalidade do sistema eGYM assim como o seu público alvo, torna-se necessário que o sistema se encontre preparado para ser disponibilizado em diferentes locais, com diferentes linguagens e com formatos de dados específicos.

A internacionalização, também conhecida no mundo informático pelo acrónimo i18n do termo inglês *Internationalization*, representa um processo de preparação e projeção de uma aplicação que possa ser usada em diferentes locais do mundo através de diferentes linguagens. A localização representa o processo de construção efetiva de diferentes versões da aplicação para diferentes locais, através da tradução de textos para diversos idiomas e formatação de dados tendo em conta as especificidades de cada local.

Um local representa e identifica uma região (como um país) do mundo, em que os seus habitantes falam um determinado idioma ou variante desse mesmo idioma. Assim para cada local existe normalmente uma variação de diversas variáveis como a sua moeda, o seu fuso horário, a sua formatação de datas, as suas unidades de medidas, entre outros...

Felizmente a *framework* escolhida para o desenvolvimento das plataformas de *Front Office* e *Back Office*, o Angular (cf. secção 6.1.2), oferece um suporte para a introdução destes conceitos nas suas aplicações, através de um módulo embutido do i18n denominado de *localize*.

Antes de proceder à análise técnica da introdução de i18n na *framework* é importante realçar que todo esta seleção de locais assenta na utilização de um identificador de localidade único, do inglês *unicode local identifier*. Esta lista de identificador é baseada numa lista de códigos **ISO 639-2**, que representa uma biblioteca de padrão internacional para códigos de

representação de nomes de idiomas. Para o Português de Portugal considera-se o código “pt-PT”, enquanto que um Inglês Americano é reconhecido através do “en-US”.

Assim que adicionado a *framework*, este módulo permite a marcação de texto que se pretenda traduzir para diferentes locais através do uso do atributo `i18n`.

```
<div i18n>An e-learning platform developed to help students attending undergraduate courses on Engineering improve their Math skills, where logic & thinking meet the best learning and teaching practices.</div>
```

Figura 51 - Marcação de texto através do uso do atributo `i18n`, no *Front Office*

É ainda possível para além desta marcação, atribuir para cada uma destas um identificador específico, permitindo assim que textos com o mesmo identificador apenas necessitem de ser traduzidos uma única vez, através do uso dos caracteres “@@”.

```
<mat-label i18n="@@password-label">Password</mat-label>
```

Figura 52 – Marcação do texto com um identificador específico através do uso do atributo `i18n`, no *Front Office*

Assim que terminada a marcação de textos pretendidos para tradução, é feita uma extração do ficheiro com o idioma de origem, neste caso inglês, através da utilização do módulo `i18n`. Neste novo ficheiro são apresentadas todas as marcações existentes na aplicação com o atributo `i18n`.

```
<trans-unit id="password-label" datatype="html">
  <source>Password</source>
  <context-group purpose="location">
    <context context-type="sourcefile">app/components/login-dialog/login-dialog.component.html</context>
    <context context-type="linenumber">22</context>
  </context-group>
</trans-unit>
```

Figura 53 - Excerto de código com a marcação do texto "Password", no *Front Office*

Para os restantes idiomas pretendidos que o sistema ofereça, é necessário efetuar uma cópia do ficheiro da linguagem original, acrescentando-se um sufixo relativo a nova língua, como por exemplo “pt” para Português. Neste novo ficheiro é possível aqui fazer as respetivas traduções tendo em conta o texto original. Cada texto original é determinado pela utilização da tag “*source*”, enquanto que a tradução respetiva é efetuada dentro da tag “*target*”.

```
</trans-unit><trans-unit id="181b01e5a9dcf8e11a9272b8b69ce18e0a4610a3" datatype="html">
  <source>An e-learning platform developed to help students attending undergraduate courses on Engineering improve their Math skills, where logic & thinking meet the best learning and teaching practices.</source>
  <target>Uma plataforma de e-learning desenvolvida para auxiliar os alunos que frequentem cursos de Engenharia a melhorar as suas capacidades Matemáticas, onde a lógica e o pensamento encontram as melhores práticas de ensino e aprendizagem.</target>
  <context-group purpose="location">
    <context context-type="sourcefile">app/components/basepage/basepage.component.html</context>
    <context context-type="linenumber">141</context>
  </context-group>
```

Figura 54 - Tradução de um excerto de texto da linguagem original para Português, no *Front Office*

Depois de efetuada todas as traduções pretendidas, é necessário informar a aplicação das localizações que esta suporta. Para isso são definidos os locais que a aplicação suporta, sendo através da definição de um local padrão, “*sourceLocale*”, e os respetivos locais suportados, “*locales*”, no ficheiro de configuração da aplicação.

```
    "e-gym": {  
      "root": "",  
      "sourceRoot": "src",  
      "projectType": "application",  
      "i18n": {  
        "sourceLocale": "en-GB",  
        "locales": {  
          "pt": "src/locale/locale.pt.xlf"  
        }  
      }  
    },
```

Figura 55 - Definição dos locais suportados pela aplicação no ficheiro, no *Front Office*

Depois de definidos os locais que a aplicação suporta é apenas necessário dizer à mesma que esta deve utilizar localização no seu arranque. Como tal é definido uma nova opção “*localize*” como verdadeira (*true*), no ficheiro de configuração da aplicação. É ainda possível para casos de desenvolvimento específicos para um dado local, definir nesta opção um conjunto finito de locais como por exemplo “*fr*”, permitindo assim que a aplicação apenas se encontre preparada para oferecer localização para a língua Francesa.

```
    "build": {  
      "builder": "@angular-devkit/build-angular:browser",  
      "options": {  
        "localize": true,  
        "aot": true,  
        "outputPath": "dist",  
        "index": "src/index.html",  
        "main": "src/main.ts",  
        "tsConfig": "src/tsconfig.app.json",
```

Figura 56 – Definição do uso de localização numa aplicação no seu ficheiro de configuração, no *Front Office*

Quando a aplicação inicia, esta necessita de saber que localização deve utilizar tendo em conta a respetiva localização ou opção de um dado utilizador, procurando assim por uma referência para o identificador do local (*LOCALE_ID*) no sistema. Para o caso do eGYM é considerado que a linguagem padrão a oferecer em primeiras utilizações é o Inglês (“en-GB”), enquanto que em utilizações futuras é utilizado a opção que o utilizador selecionar, sendo esta guardada no próprio *browser* através do uso da variável *localStorage*. Esta variável é bastante útil, sendo aqui armazenada informação através do uso de pares chave/valor, sem uma data de expiração, podendo assim ser reutilizada e acedidas no dia seguinte ou semana seguinte, através de um identificador específico.

```
{ provide: LOCALE_ID, useValue: localStorage.getItem('locale') || 'en-GB' },
```

Figura 57 – Determinação do local a utilizar para o utilizador, no *Front Office*

6.2 Testes

A presente secção descreve a metodologia utilizada para a construção dos testes implementados no sistema, assim como técnicas utilizadas para a automatização da sua utilização.

6.2.1 Metodologia

O desenvolvimento de testes é sempre um passo importante na construção de novos sistemas, tendo em conta os seus objetivos de detetar possíveis falhas nas aplicações desenvolvidas que não tenham sido detetadas.

Assentando num sistema já previamente existente, os testes utilizados nesta nova reengenharia do sistema partilham os mesmos testes anteriormente desenvolvidos. Naturalmente que com as entradas e saídas de novas entidades no sistema, alguns testes teriam de ser desenvolvidos de raiz enquanto que outros deixariam de ter qualquer utilidade para o sistema.

Foram assim para esta reengenharia do sistema reutilizados e construídos testes de integração. Estes testes agrupam normalmente diferentes módulos do sistema num grupo, expondo possíveis falhas nas interações entre os componentes e as interfaces dos sistemas integrados.

Para as API expostas do sistema, como por exemplo o *Back End* do *Back Office*, foram construídos testes que permitissem testar diferentes operações do tipo CRUD sobre as entidades existentes no sistema. Por exemplo para uma área são desenvolvidos testes que permitem verificar se um utilizador não autenticado, um utilizador com autenticação expirada, um utilizador com acesso inválido e um utilizador com autenticação e acesso válido conseguem executar uma pesquisa sobre uma área através do seu identificador (Figura 58).

```

describe('/GET :id', () => {
  it('it should not GET an area by id without token', (done) => {
    let curr = url + areaA._id;
    chai.request(server)
      .get(curr)
      .end((err, res) => {
        mlog.log("boddy-----" + res.body);
        res.should.have.status(403); // Code error for no token provided
        done();
      });
  });

  it('it should not GET an area by id without a valid token', (done) => {
    let curr = url + areaA._id;
    chai.request(server)
      .get(curr)
      .set('x-access-token', tmpToken)
      .end((err, res) => {
        res.should.have.status(500); // Code error for invalid token
        done();
      });
  });

  it('it should not GET an area by id without a valid access', (done) => {
    let curr = url + areaA._id;
    chai.request(server)
      .get(curr)
      .set('x-access-token', basic.token)
      .end((err, res) => {
        res.should.have.status(401); // Code error for invalid access
        done();
      });
  });

  it('it should GET an area by id with valid token and access', (done) => {
    let curr = url + areaA._id;
    chai.request(server)
      .get(curr)
      .set('x-access-token', boss.token)
      .end((err, res) => {
        res.should.have.status(200); // OK HTTP GET Code for successfull query
        res.body.should.be.an('object');
        done();
      });
  });
});

```

Figura 58 – Excerto de código para os diferentes testes para uma área, no *Back End* do *Back Office*

Todos estes testes são efetuados sobre uma base de dados externa, evitando assim o sobrecarregamento dos servidores existentes com processos de testes e a introdução de dados temporários nas primeiras. Como tal na configuração das ligações as bases de dados do sistema, é necessário considerar a ligação a uma base de dados diferente para quando o sistema se encontra em ambiente de teste.

```

if (process.env.NODE_ENV == 'test') {
  console.log("Test Enviroment created !")
  mongoConn.connect(config.DBTest, { useNewUrlParser: true, useUnifiedTopology: true });
} else {
  mongoConn.connect(config.eGYMDB, { useNewUrlParser: true, useUnifiedTopology: true });
}

```

Figura 59 – Configuração das ligações às bases de dados, no *Back End* do *Back Office*

É importante realçar que todos os testes desenvolvidos utilizam uma *framework* denominada de *Mocha*. Esta *framework*, especialmente desenvolvida para tecnologias Node.js, permite executar de uma forma simples e assíncrona testes de integração. Em conjunto com o *Mocha* é utilizada a biblioteca de testes *Chai*, que oferece uma linguagem mais expressiva e com um estilo bem mais legível de testes assertivos (Figura 58).

6.2.2 *Continuous Delivery / Continuous Integration*

A utilização de *Continuous Delivery / Continuous Integration* permite que alterações efetuadas no código de uma aplicação possam ser testadas e implantadas de uma forma automatizada.

Para tal e aproveitando a *feature* já existente no repositório de controlo de versões, são utilizadas *pipelines* que permitem a automatização deste processo, sendo estas denominadas de *Bitbucket-Pipelines*.

Para tal é apenas necessário introduzir um ficheiro na raiz de cada aplicação, que contenha a configuração desejada.

```

# This is a sample build configuration for JavaScript.
# Check our guides at https://confluence.atlassian.com/x/14UWN for more examples.
# Only use spaces to indent your .yaml configuration.
# -----
# You can specify a custom docker image from Docker Hub as your build environment.
image: node:10.15.0

pipelines:
  branches:
    master:
      - step:
          caches:
            - node
          script: # Modify the commands below to build your repository.
            - npm install
            - chmod 0777 ./node_modules/.bin/mocha
            - npm test
            - ssh -T egym@e-gym.dei.isep.ipp.pt "cd deployment_scripts;/sh egym_statistic_api.sh"

```

Figura 60 – Ficheiro de configuração da *pipeline*

Na Figura 60 é possível observar a estrutura de configuração de uma dada *pipeline*, neste caso específico para a *Statistics* API. Antes de se configurar a *pipeline* é necessário decidir qual a imagem do ambiente a utilizar, sendo que no caso da figura anterior é utilizado a versão 10.15 do Node.

Com a versão escolhida, procede-se a configuração da respetiva *pipeline*. Para a reengenharia deste sistema, esta técnica é apenas utilizada no *branch master*. Foi utilizado um modelo de *Git branching*¹³ onde todas as alterações efetuadas localmente são introduzidas no *branch develop*, sendo depois estas diluídas no *branch master* assim que estiverem preparadas para serem introduzidas nos servidores.

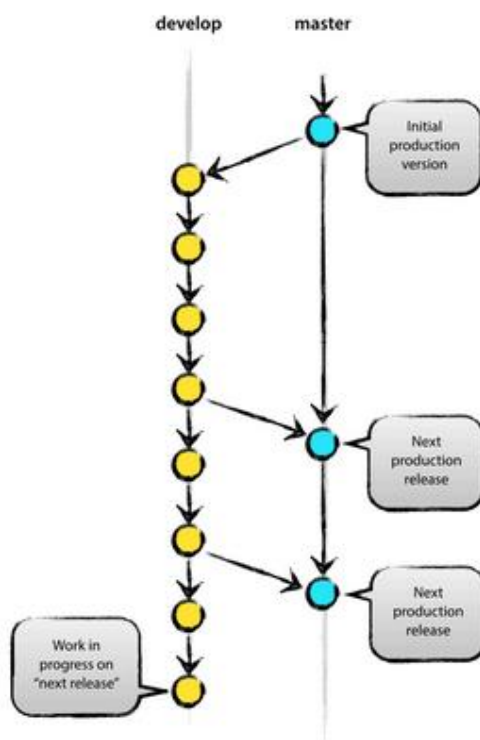


Figura 61 – *Git branching* simplificado (Driessen, 2010)

Sendo assim, antes de se introduzirem estas alterações nos servidores, é importante validar se estas mesmas conseguem ultrapassar os testes existentes. Como tal e analisando mais uma vez a Figura 60 é possível verificar que antes de se proceder a implantação das mudanças é utilizado o *script "npm test"*. Este é o *script* dá início ao processo de testes, seguindo uma estrutura semelhante à apresentada na secção anterior.

Só se os testes forem concluídos com sucesso, é que a *pipeline* avança para a próxima linha do *script*, sendo que esta é nessa próxima linha que se procede à implantação das alterações no servidor. No caso do eGYM esta implantação é realizada através do uso de *ssh* e através de ficheiros de implantação preparados, que automatizam o processo de alteração de versões.

¹³ Estratégia/técnica que permite dividir um sistema de controlo de versões em diferentes *branches*, cada uma delas com um propósito específico.

7 Resultados Obtidos

No presente capítulo são demonstrados todos os resultados obtidos, através do auxílio de imagens representativas das interfaces do sistema desenvolvido, funcionando assim como um “manual” de utilização do sistema.

Sendo assim numa primeira secção são demonstrados todos os resultados obtidos sobre os objetivos descritos na secção 1.3.

Reserva-se ainda uma segunda e última secção para a demonstração de processos relevantes na implementação do sistema, que não se encontrem enunciados na secção 1.3, mas que se representem como funcionalidades que melhorem a qualidade geral do sistema como um todo.

7.1 Funcionalidades Principais

7.1.1 Configurabilidade e extensibilidade de atividades e conteúdos

Uma das principais limitações existentes no sistema anteriormente desenvolvido centrava-se com a inexistência de uma configurabilidade e extensibilidade de atividades e conteúdos existentes na plataforma. Esta limitação tornava todo o processo de criação e gestão dos conteúdos e atividades bastante moroso e ineficiente, sendo sempre necessário proceder a alteração de todos os conteúdos ou até mesmo ao *upload* dos mesmos ficheiros.

Relativamente as atividades existentes na plataforma, o sistema anterior apresentava-se bastante limitado, oferecendo apenas a possibilidade de criação, edição e remoção dos seguintes tipos: **áreas (disciplinas da Engenharia), módulos (aulas) e jogos didáticos.**

Nesta nova iteração de desenvolvimento procurou-se introduzir uma maior variedade de atividades e conteúdos existentes na plataforma.

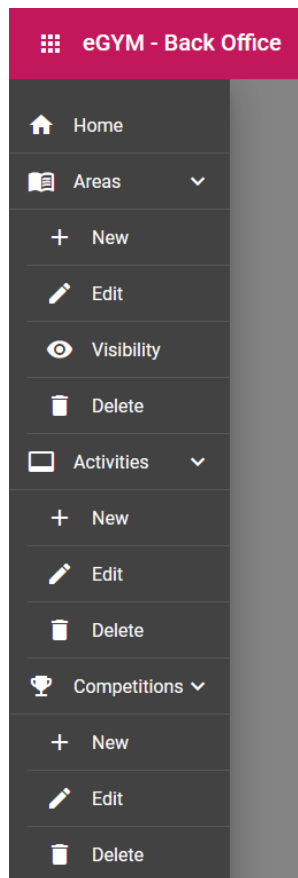


Figura 62 - *Sidebar* de navegação do *Back Office*

Devido a grande variedade de conteúdos e atividades existentes na plataforma, cada uma destas será de seguida distinguida e retratada numa subsecção dedicada, procurando para cada uma destas descrever e exemplificar todas as implementações efetuadas sobre a configurabilidade e extensibilidade das mesmas.

7.1.1.1 Área

Na fase inicial de desenvolvimento do sistema e persistindo na sua última iteração, o processo de criação de uma Área apresentava-se bastante simples e estático oferecendo poucas possibilidades de configurabilidade e extensão aos seus utilizadores.

Na sua criação, uma Área aceitava um pequeno número finito de parâmetros como o seu nome, descrição, iniciais e os professores que integravam esta área.

Figura 63 - Criação de uma área em sistemas anteriores (João Fernandes and José Barros, 2018)

Esta simples implementação acarretava bastante problemas para o sistema na sua globalidade. Tornava-se impossível de no ato de criação definir à partida os conteúdos e atividades existentes na mesma, sendo para isso necessário editar individualmente cada conteúdo.

Nesta nova abordagem ao sistema, procurou-se combater estas dificuldades, oferecendo aos seus utilizadores um maior grupo de parâmetros e possibilidade no ato de criação e edição de uma Área.

Assim, para além dos parâmetros anteriormente definidos o utilizador pode no seu processo de criação e edição definir novos parâmetros como:

- **Grau de Educação:** representa o grau de educação da determinada “cadeira”.
- **Área de Estudo:** representa a área de estudo da determinada “cadeira”.

Com a introdução destes novos parâmetros, o sistema fica preparado para num futuro se necessário se expandir para outros tipos de conteúdos que não estejam diretamente relacionados com as Engenharias e/ou as Matemáticas. Atualmente e como descrito na análise do sistema, o eGYM foca-se apenas no estudo de disciplinas relacionadas com as áreas da Matemática no ensino superior de Engenharia. No entanto com a definição destes dois novos parâmetros, o sistema pode num futuro próximo avançar para um conceito mais amplo, no caso de novos graus de educação como o ensino básico e para novas áreas de estudo como as Línguas.

Estes parâmetros são também configuráveis, deixando alguma liberdade para os seus utilizadores no processo de definição de novos tipos. Atualmente é apenas permitido o ramo da Matemática e os graus de ensino superior (Licenciatura e Mestrado) como opções, no entanto com o uso do botão de configuração, podem ser acrescentados, em qualquer momento, novos tipos.

Area Form

Area Information | Area Teachers | Area Classes | Area Online Games

Name *
Algebra Linear

Description *
Studies linear equations

Initials *
ALGAN

Education Degree *
Bachelor's Degree

Field of Study *
Mathematics

> Next

Figura 64 - Exemplo de criação de uma Área, no *Back Office*

Para além destes, o utilizador pode agora no seu processo de criação também definir as diferentes aulas (módulos), podendo para cada uma destas definir também o número de pontos necessários para desbloquear esse conteúdo. Esta pequena alteração na estrutura do processo de criação permite assim de uma maneira simples e eficaz, oferecer uma extensibilidade de todos os módulos existentes no sistema. É importante salientar que apesar de estes poderem agora ser estendidos por todas as áreas, a sua ligação com as diferentes áreas diferencia sempre na definição do número de pontos necessários, sendo que estes pontos são apenas validos para aquela Área criada/editada e não para todas que contenham esse dado módulo.

Area Form

Area Information | Area Teachers | Area Classes | Area Online Games

+ Add Class

Class: Introduction to Matrices | Points Required: 0

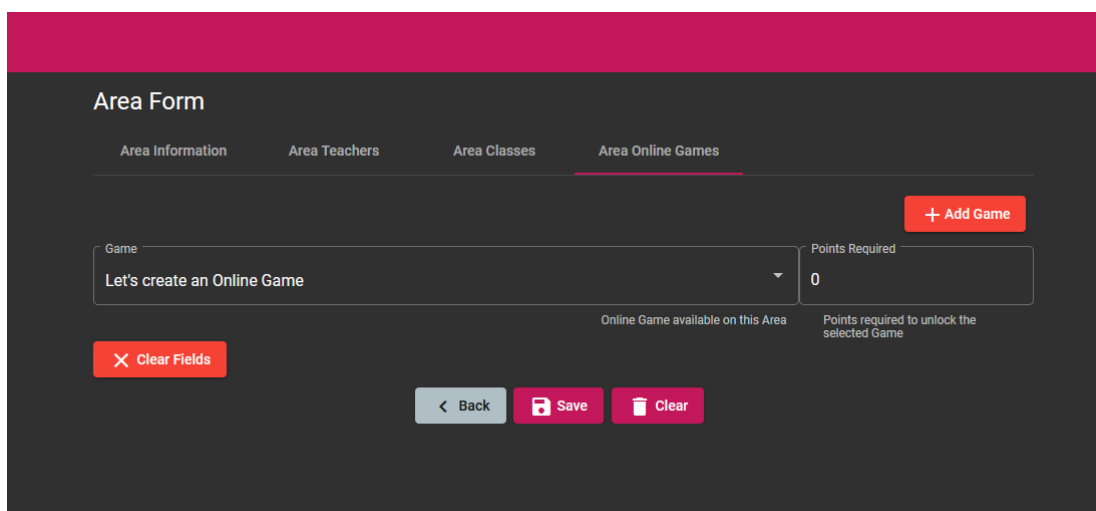
Class: Another Class | Points Required: 10

Clear Fields | Delete Option

< Back | > Next

Figura 65 - Definição dos módulos de uma Área, no *Back Office*

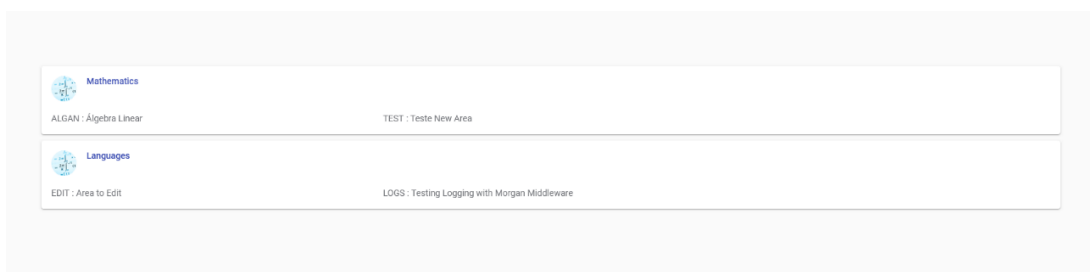
Da mesma forma no seu processo de criação podem também ser definidos os jogos multijogador existentes, assim como os pontos de Área necessários para desbloquear esse mesmo jogo. Mais uma vez, com esta simples alteração no processo de criação, é atingido o objetivo de extensibilidade dos conteúdos do sistema.



The screenshot shows a web interface titled "Area Form" with a dark background. At the top, there are four tabs: "Area Information", "Area Teachers", "Area Classes", and "Area Online Games" (which is selected). Below the tabs, there is a red button labeled "+ Add Game". The main form area contains a "Game" dropdown menu with the text "Let's create an Online Game" and a "Points Required" input field with the value "0". Below the input fields, there are two small text labels: "Online Game available on this Area" and "Points required to unlock the selected Game". At the bottom of the form, there are three buttons: a red "Clear Fields" button, a grey "Back" button, a red "Save" button, and a red "Clear" button.

Figura 66 - Definição dos jogos multijogador de uma Área, no *Back Office*

Assim que criadas, estas áreas estariam disponíveis para os seus utilizadores no *Front Office*.



The screenshot shows a list of areas in the "Front Office". The list is organized into two sections: "Mathematics" and "Languages". Under "Mathematics", there are two items: "ALGAN - Algebra Linear" and "TEST - Teste New Area". Under "Languages", there are two items: "EDIT - Area to Edit" and "LOGS - Testing Logging with Morgan Middleware".

Figura 67 – Listagem das áreas existentes por área de estudo, no *Front Office*

Depois de selecionada a Área pretendida, toda a sua informação é apresentada seguindo uma estrutura baseada em módulos. Cada módulo encontra-se dividido nos diferentes tipos de conteúdos existentes, apresentando para cada um deles os seus conteúdos e uma pequena informação visual sobre se o conteúdo já foi consumido pelo utilizador e para o caso dos jogos, o número de pontos obtido. Cada tipo de conteúdo pode ser expandido, através de um clique, apresentando toda a informação pertinente do mesmo.

Para além disto, esta interface apresenta ainda no canto superior direito o número de pontos obtidos na Área, assim como o seu progresso a nível de pontos para atingir a totalidade de pontos disponíveis.

Por fim é ainda também apresentado ao utilizador uma pequena secção dedicada ao histórico de conteúdos consumidos pelo utilizador, separando o histórico dos jogos e dos conteúdos estáticos (vídeo-aulas e documentos).

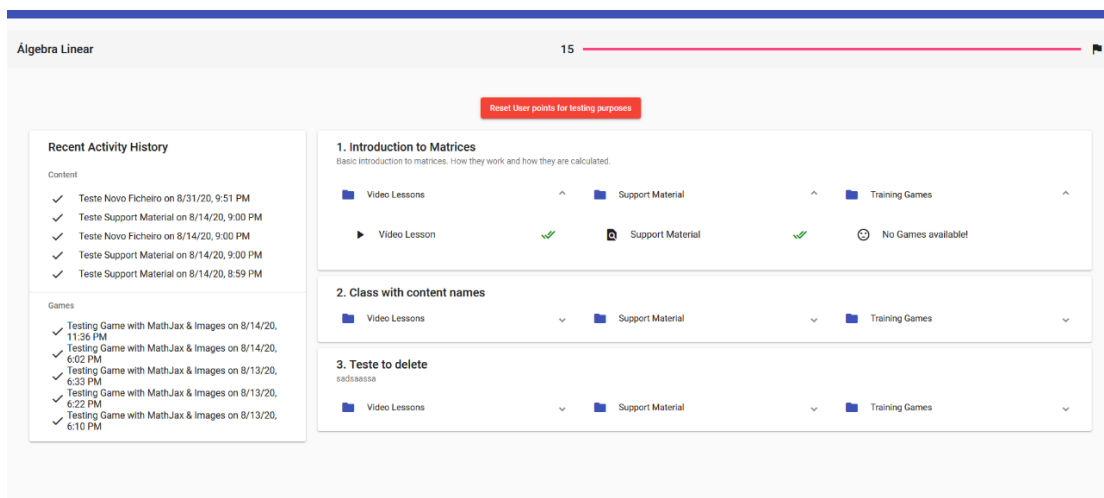


Figura 68 - Apresentação de uma Área, no *Front Office*

No caso de o utilizador não possuir o número de pontos necessários para desbloquear os conteúdos (sistema de progressão, analisado anteriormente na secção 3.2.1), a interface alerta o utilizador para tal, bloqueando qualquer tipo de acesso a esse conteúdo até o número de pontos ser alcançado.

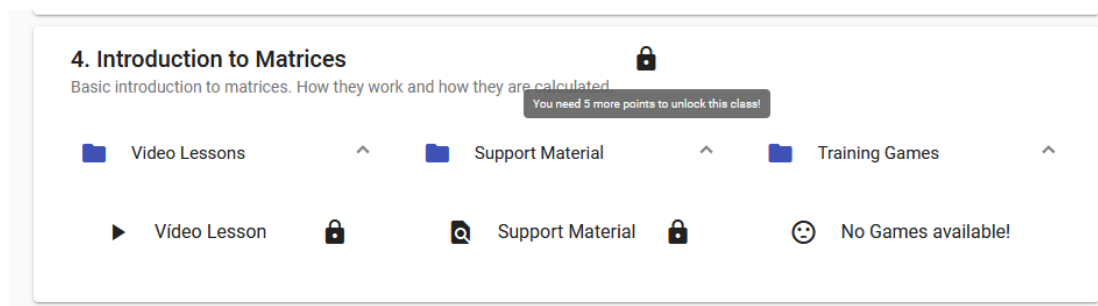


Figura 69 – Alerta do número insuficiente de pontos para desbloquear um conteúdo, no *Front Office*

Quanto ao seu processo de edição, este é um processo bastante semelhante ao da sua criação, variando apenas na necessidade de selecionar a Área a editar previamente. Depois de selecionada toda a sua informação é carregada para os formulários, agilizando assim todo o seu processo de edição.

Area
ALGAN - Álgebra Linear

Select one Area to edit!

Area Form

Area Information Area Teachers **Area Classes** Area Online Games

+ Add Class

| Class | Points Required |
|--------------------------|-----------------|
| Introduction to Matrices | 0 |
| Teste to delete | 5 |
| Class with content names | 0 |

Class available in this Area Points required to unlock this Class

X Clear Fields **Delete Option**

X Clear Fields **Delete Option**

X Clear Fields **Delete Option**

< Back **> Next**

Figura 70 – Processo de edição de uma Área, no *Back Office*

Pode ainda ser observado, através da Figura 70, que a Área selecionada “ALGAN – Álgebra Linear” possui três distintos módulos, ou aulas, e as suas respetivas pontuações de desbloqueamento diferenciadas. É assim possível observar que dois dos três módulos se encontram disponíveis para os utilizadores desde o seu início, tendo em conta que necessitam de zero pontos, enquanto que uma terceira “Teste to delete” apenas estará desbloqueada aos seus utilizadores assim que estes atingirem cinco pontos na respetiva Área.

7.1.1.2 Módulos

Relativamente às aulas existentes no sistema, foram encontrados problemas semelhantes às áreas nos processos de criação e de edição. Mais uma vez, o sistema anteriormente desenvolvido oferecia um conjunto de parâmetros limitados e estáticos que impossibilitavam de certa forma a sua extensão ao longo de toda a plataforma.

Dando seguimento à construção implementada para as áreas, o sistema oferece agora para além dos parâmetros necessários como o sumário da aula e a sua descrição, a possibilidade de definir em qualquer processo de criação, as áreas as quais este dado módulo pertence.

Content Type
Class

Select one type of content that you wish to create!

Class Form

Class Information Class Vídeo Lessons Class Support Material Class Games **Class Areas**

+ Add Area

Area: ALGAN: Algebra Linear Points Required: 0

Areas where this Class is present Points required to unlock this Class

X Clear Fields

Area: LOGS: Testing Logging with Morgan Middleware Points Required: 5

Areas where this Class is present Points required to unlock this Class

X Clear Fields **Delete Option**

< Back **Save** **Clear**

Figura 71 – Definição das áreas para as quais esta aula (módulo) se encontra presente, no *Back Office*

Outra alteração significativa passou pela introdução de jogos didáticos (jogos de único jogador) dentro de cada aula. Anteriormente o sistema apenas disponibiliza nos módulos, conteúdos como as vídeo-aulas e os documentos de suporte. Depois de múltiplas análises do negócio e conversas entre as partes interessadas, foi decidido que seria o mais correto que cada aula pudesse oferecer vários jogos didáticos com dificuldades diferentes, que atribuísem um número de pontos assim que fossem completos, promovendo assim novamente uma extensibilidade de conteúdos.

Content Type
Class

Select one type of content that you wish to create!

Class Form

Class Information Class Vídeo Lessons Class Support Material **Class Games** Class Areas

+ Add Game

Game: Testing Game with MathJax & Images Points Awarded: 20

Game to be included in this Class Points awarded for completing this Game

X Clear Fields

< Back **> Next**

Figura 72 – Definição dos jogos didáticos de uma aula (módulo), no *Back Office*

A maior melhoria em todo este processo recai, no entanto, sobre a extensibilidade das vídeo-aulas e dos documentos de suporte inseridos dentro de um módulo. Ao contrário dos anteriores, este tipo de conteúdo assenta na utilização de ficheiros de áudio ou PDF, sendo por isso uma operação que acarrete algum peso ao nível do servidor.

Em versões anteriores o sistema já se encontrava preparado para lidar com estes ficheiros, fazendo um *upload*¹⁴ dos mesmos para uma CDN¹⁵, no entanto cada vez que um módulo fosse criado ou editado os ficheiros seriam novamente enviados para a CDN, mesmo que estes já tivessem sido utilizados em outro módulo, fazendo assim com que todo este processo fosse ineficaz e que não promovesse a extensibilidade destes conteúdos.

Como tal grande parte do esforço recaiu assim na reformulação deste subprocesso específico. Seria necessário encontrar uma solução que conseguisse reutilizar este tipo de conteúdos na plataforma, evitando assim o sobre carregamento do servidor e da CDN com o *upload*¹⁴ de ficheiros repetidos.

Sendo assim, a nova solução implementada passou a considerar estes ficheiros utilizados como entidades do sistema, permitindo assim a sua configurabilidade e extensibilidade. Atualmente aquando da criação ou edição de um módulo, é possível de uma forma independente fazer o *upload*¹⁴ dos ficheiros, necessitando apenas de o nome a disponibilizar e do próprio ficheiro, como é o caso das vídeo-aulas.

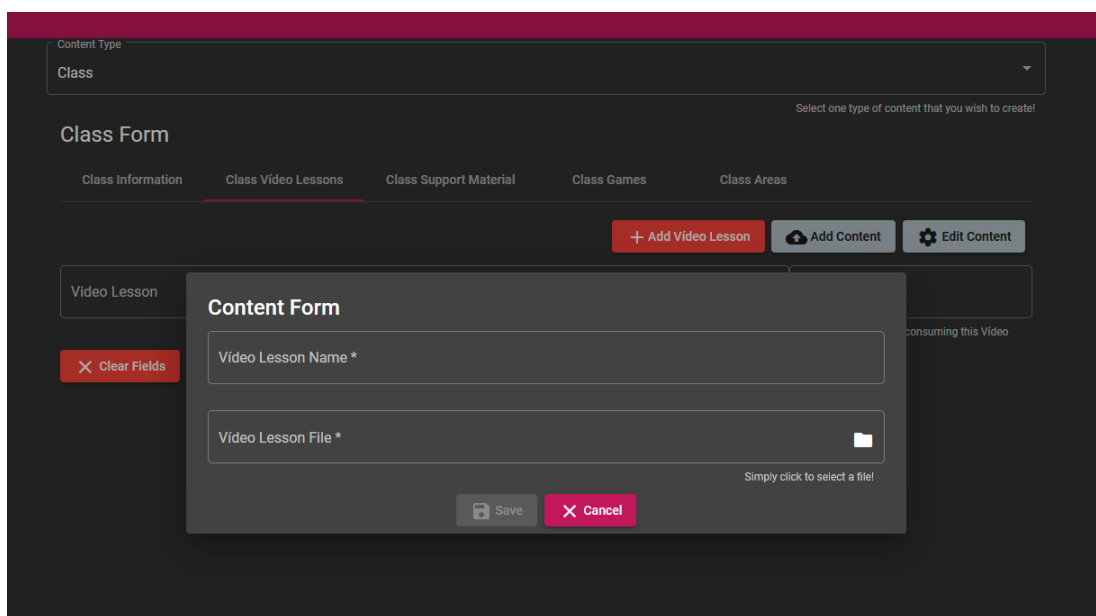
The image shows a screenshot of a web application interface. At the top, there is a 'Content Type' dropdown menu set to 'Class'. Below this is a 'Class Form' section with several tabs: 'Class Information', 'Class Video Lessons' (which is active), 'Class Support Material', 'Class Games', and 'Class Areas'. In the 'Class Video Lessons' tab, there are three buttons: '+ Add Video Lesson' (red), 'Add Content' (grey), and 'Edit Content' (grey). A modal window titled 'Content Form' is open in the foreground. It has a 'Clear Fields' button (red) on the left. The modal contains two input fields: 'Video Lesson Name *' and 'Video Lesson File *'. The 'Video Lesson File' field has a folder icon and the text 'Simply click to select a file!'. At the bottom of the modal are 'Save' and 'Cancel' buttons.

Figura 73 - Processo de criação de uma vídeo-aula fornecendo o nome e o ficheiro, no *Back Office*

Depois de criados, estes conteúdos podiam agora ser escolhidos através de uma simples seleção, podendo assim ser reutilizados em qualquer momento e para qualquer módulo, necessitando apenas de definir para além do conteúdo o número de pontos atribuídos a um estudante pelo consumo do mesmo.

¹⁴ Envio

¹⁵ *Content Delivery Network*, traduzida para Rede de Distribuição de Conteúdo, que funciona como um grupo de servidores distribuídos que armazenam e entregam conteúdo através da internet.

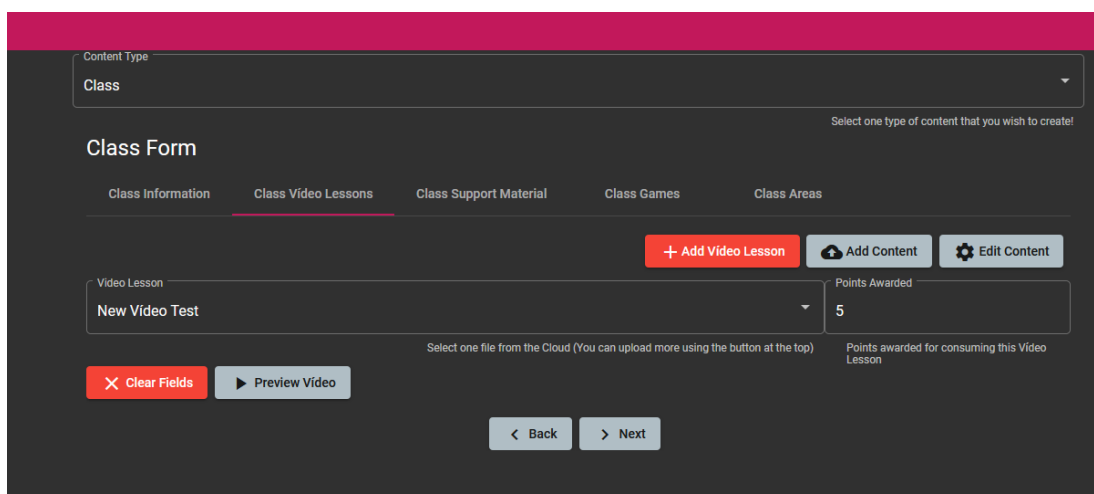


Figura 74 – Seleção de uma vídeo-aula num módulo depois de esta ser criada no sistema, no *Back Office*
 Da mesma maneira que um material de suporte, slides ou exemplos de exercícios, podem ser criados em qualquer momento.

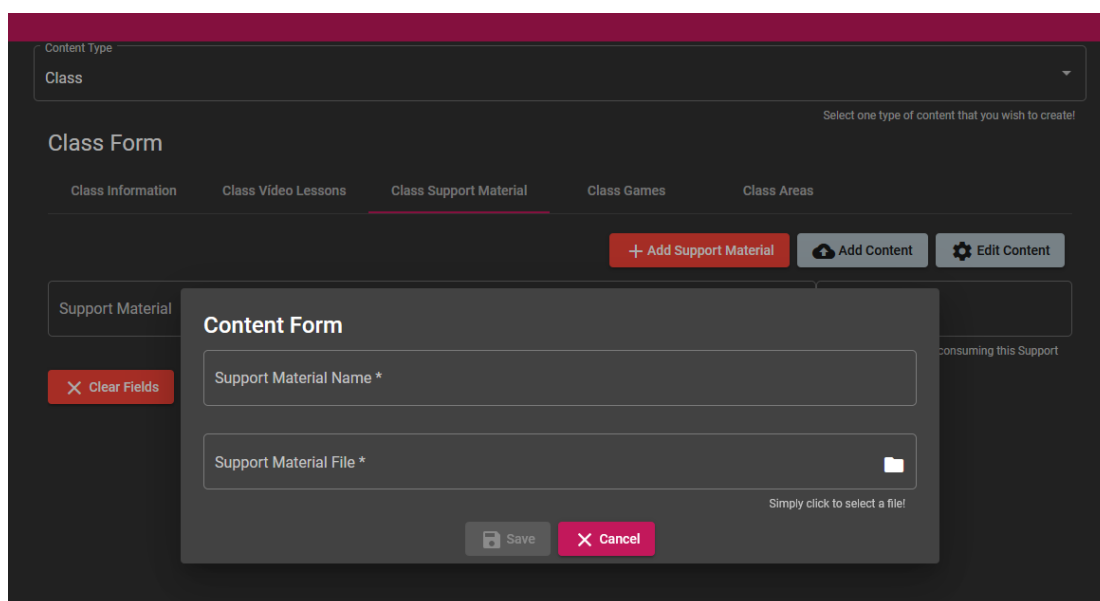


Figura 75 – Processo de criação de um material de suporte fornecendo o nome e o ficheiro, no *Back Office*
 Da mesma maneira que as vídeo-aulas, estes depois de criados podem ser seleccionados da mesma forma.

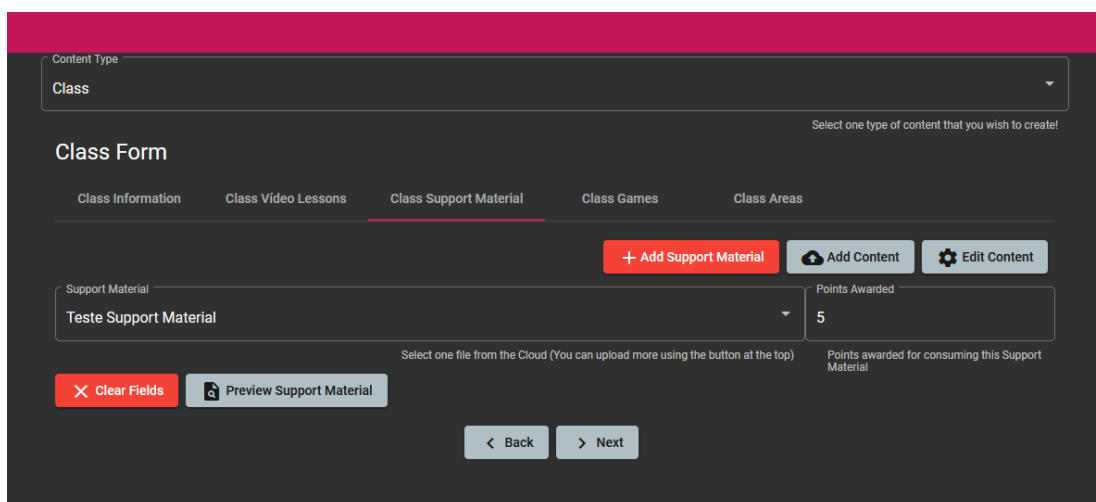


Figura 76 – Seleção de um material de suporte num módulo depois de este ser criado no sistema, no *Back Office*

Depois de criados, estes módulos passam a estar disponíveis dentro das suas áreas. Como referido inicialmente, cada módulo dispõe de três tipos de conteúdos: vídeo-aulas, material de suporte e jogos didáticos. Cada um destes pode ser acedido através de um simples clique no respetivo local apropriado.

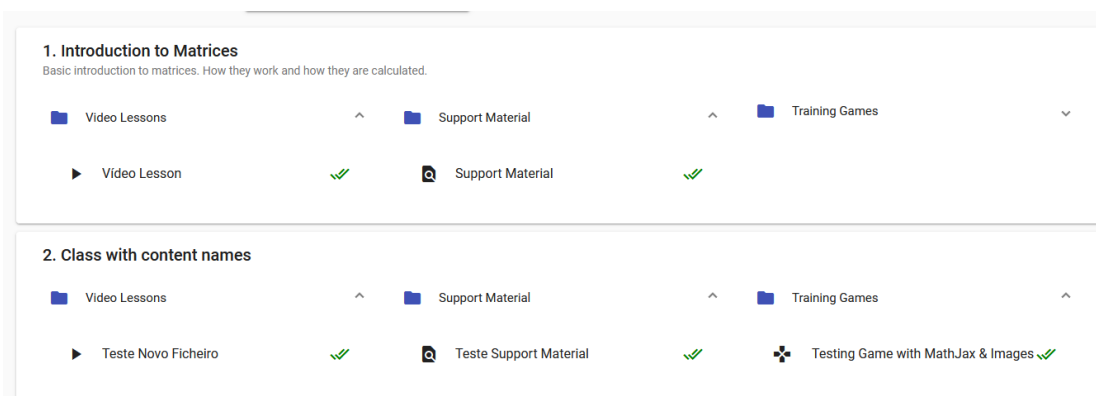


Figura 77 - Representação dos módulos e os seus respetivos conteúdos, no *Front Office*

No caso das vídeo-aulas, se estas estiverem disponíveis para consumo, um pequeno reproduzidor de vídeo é apresentado numa janela de diálogo¹⁶, onde o utilizador pode reproduzir o vídeo livremente oferecendo um conjunto de operações básicas dos leitores de vídeos atuais.

¹⁶ Janelas pop-up que aparecem sobre a janela principal em vez de redirecionar a plataforma para uma nova janela.

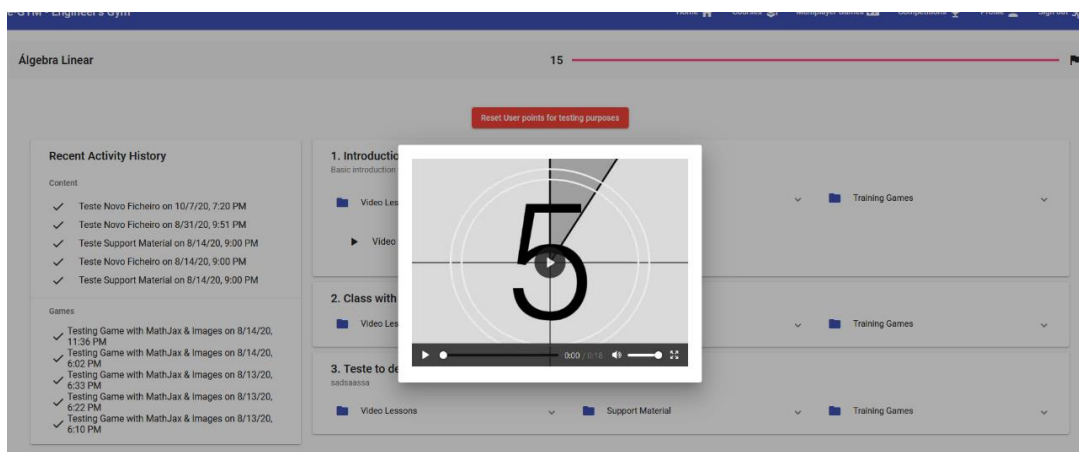


Figura 78 – Reprodução de uma vídeo-aula, no *Front Office*

No caso dos materiais de suporte, a ideologia implementada é a mesma, utilizando novamente as janelas de diálogo¹⁶, só que para este tipo de conteúdo é utilizado um leitor de PDF em vez de um leitor de vídeo.

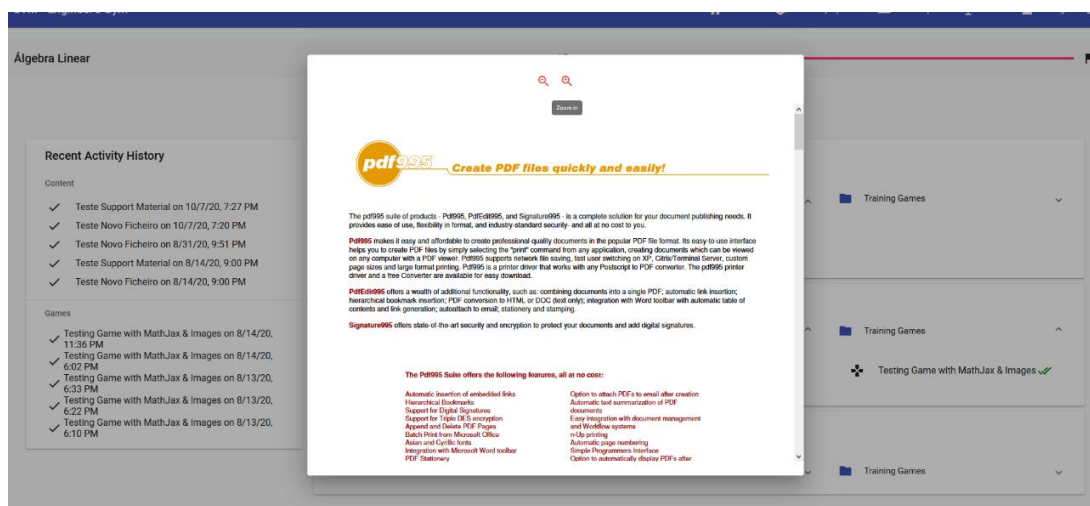


Figura 79 – Visualização de um material de suporte, no *Front Office*

Quando uma vídeo-aula ou um material de suporte seja consumido pela primeira vez, são atribuídos os respetivos pontos ao utilizador na respetiva Área. O histórico de atividade de conteúdos do utilizador, apresentado do lado esquerdo da página, é atualizado e na listagem de conteúdos, o respetivo conteúdo para a contar com uma pequena indicação visual que permite mostrar ao utilizador em qualquer momento que aquele conteúdo já foi visualizado pelo mesmo.

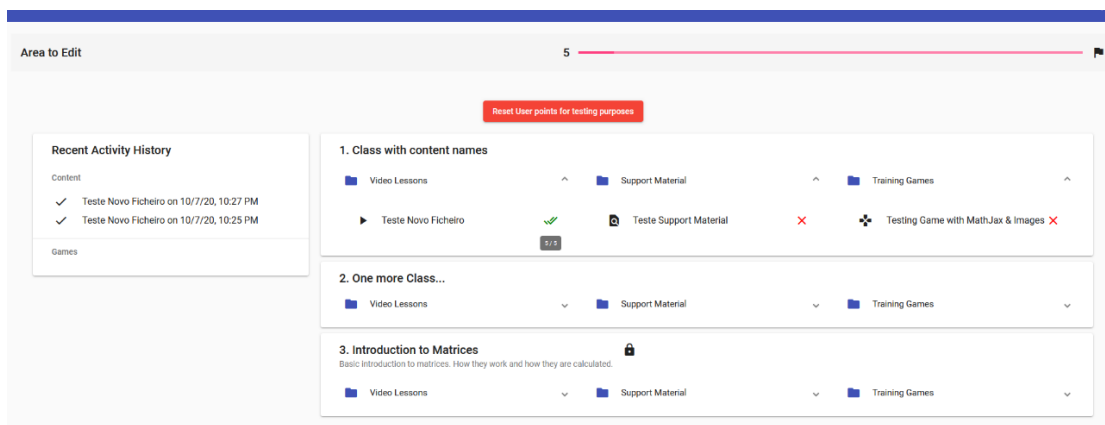


Figura 80 – Informação visual do consumo de um conteúdo de um módulo, no *Front Office*

A edição de um módulo apresenta um processo bastante semelhante à sua criação, necessitando apenas de uma seleção do módulo pretendido para editar antes de visualizar os seus dados e efetuar qualquer tipo de alteração. Depois da sua seleção todos os dados correspondentes são carregados para o respetivo formulário.

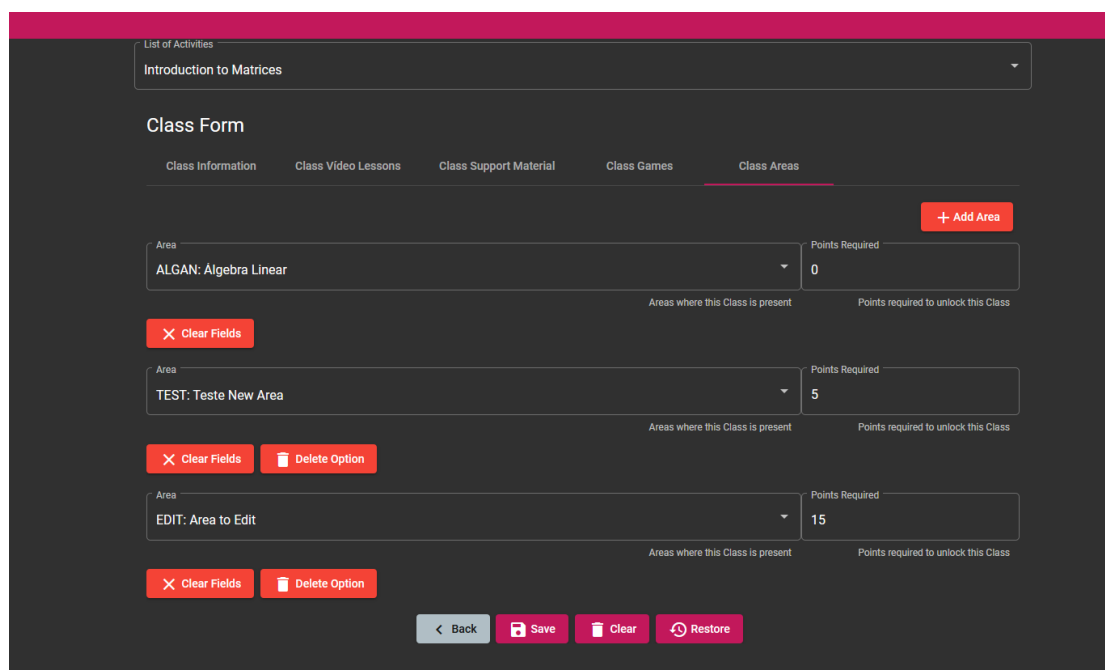


Figura 81 - Processo de edição de um módulo, no *Back Office*

Pode-se ainda visualizar na Figura 81 que o módulo seleccionado “*Introduction to Matrices*” está presente em várias áreas, como o caso de Álgebra Linear, demonstrando mais uma vez em funcionamento, a extensibilidade deste tipo de atividades e conteúdos ao longo do sistema.

7.1.1.3 Jogos didáticos

Quanto aos jogos didáticos foi possível verificar depois de uma análise aos sistemas anteriores, que os seus processos se apresentavam bastante primitivos e limitados. Cada Jogo criado podia apenas pertencer a um dada Área, ficando sempre dependente desta mesma, impedindo assim a sua extensibilidade ao longo do sistema.

CREATE GAME

Area Selection MAT - Matematica

Game Name
Pitagoras

Game Difficulty 1

Game Helps
Fifty Fifty

Time Limit (seconds)
20

| Nº | Question | Change Order | |
|----|---|--------------|--|
| 1 | Qual a formula para calcular a hipotenusa ? | | |

Create Game Cancel

Figura 82 - Criação de um Jogo em sistemas anteriores (Luís Peixoto, 2019)

Para além desta pequena, mas grande limitação no contexto global do sistema, é possível ainda observar através da Figura 82 uma pequena subjetividade no nível de dificuldade dos Jogos. Como tal para esta nova versão desenvolvida foram adicionados quatro novos graus de dificuldade que permitissem melhor aos seus utilizadores compreender as dificuldades concretas de cada jogo, sendo estes:

- **Fácil:** nível de dificuldade mais básico.
- **Normal:** nível de dificuldade mediano.
- **Difícil:** nível de dificuldade acrescido.
- **Extrema:** nível de dificuldade extremo.

Foi ainda decido em conjunto com as partes interessadas que estes tipos de jogos didáticos não teriam daqui em diante qualquer limitação imposta a limite de tempo de jogo, tendo em conta o propósito que este representava para os seus utilizadores. Este tipo de Jogo devia ser aproveitado para os seus utilizadores como um Jogo mais casual, onde o utilizador poderia obter algum *feedback* em relação ao seu progresso sobre os conteúdos estudado e assimilados, sem qualquer veia competitiva ou pressão adicional.

Figura 83 – Exemplo do processo de criação de um jogo didático

Para além disto, uma das suas maiores limitações centrava-se nas questões de um dado Jogo. No sistema antigo cada questão era criada no momento, estando assim apenas disponível para esse mesmo Jogo. O utilizador do *Back Office* poderia assim adicionar quantas questões quisesse para um dado Jogo, no entanto no próximo jogo que criasse teria de criar novamente as mesmas questões se assim o necessitasse.

| Nº | Answer | Percentage | Change Order |
|----|-------------------|------------|--------------|
| 1 | $H^2 = C^2 + C^2$ | 100% | ↑ ↓ |
| 2 | $E = (mc)^2$ | 0% | ↑ ↓ |

Figura 84 - Processo de criação de uma questão no sistema anterior (Luís Peixoto, 2019)

Para resolver esta limitações, as questões passaram a ser consideradas entidades do sistema, podendo assim ser guardadas no sistema aquando da sua criação, permitindo assim a sua reutilização em qualquer momento. O utilizador podia assim seleccionar questões já existentes no sistema, ou através de um simples clique criar uma questão no próprio instante.

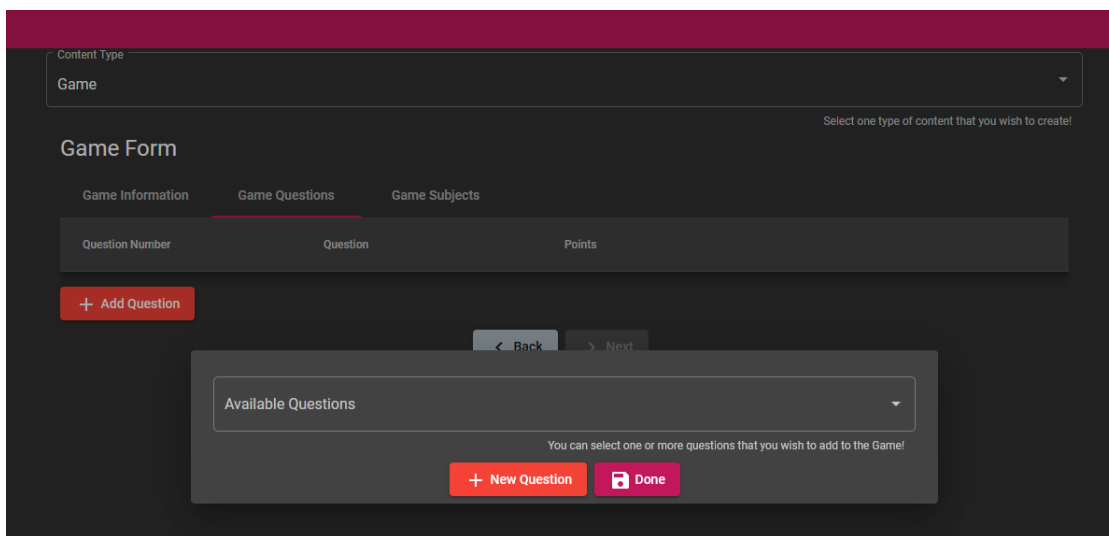


Figura 85 – Seleção e criação de questões, através de uma janela de diálogo, no *Back Office*

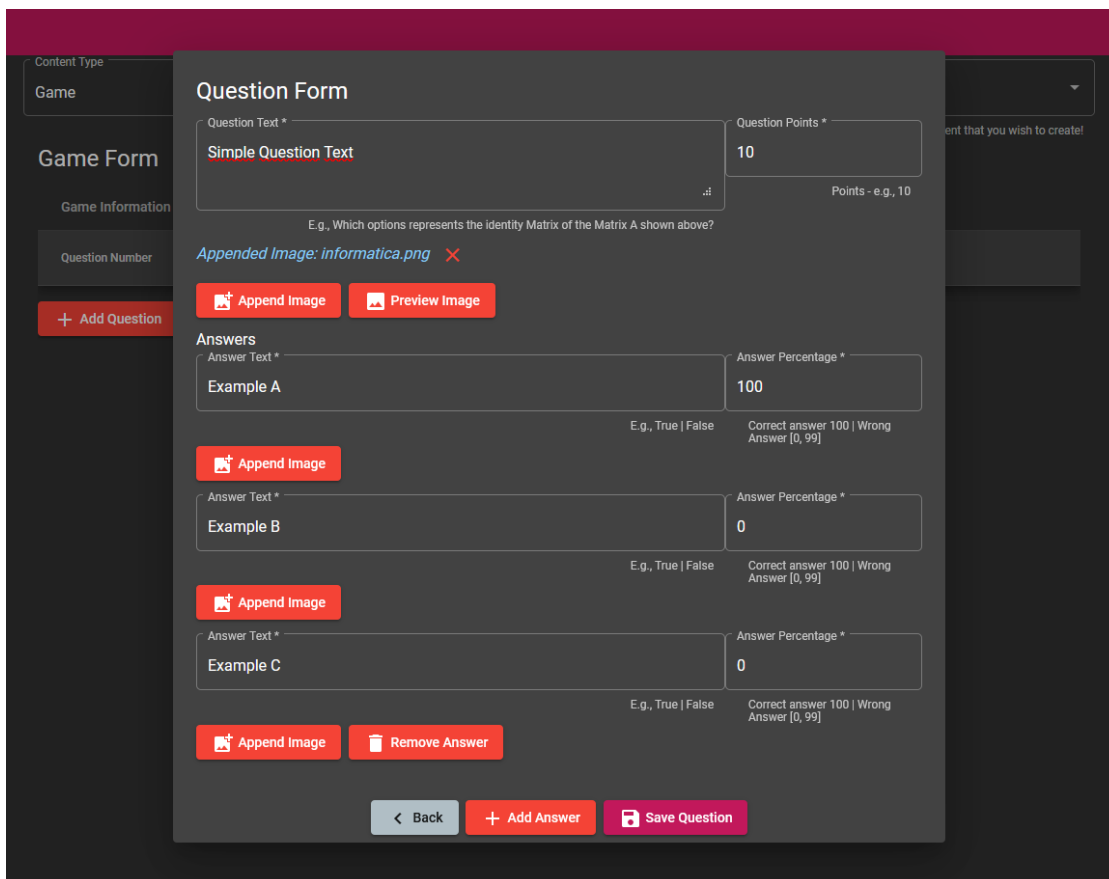


Figura 86 – Exemplo da criação de uma questão, no *Back Office*

Para além disto, o sistema anterior permitia apenas questões em formato de texto, isto é, as questões e as suas respetivas respostas apenas aceitavam texto como possível *input*. Como podemos observar na Figura 86, é permitido agora que os utilizadores do *Back Office* possam livremente anexar imagens ao texto, ou simplesmente utilizar as imagens, oferecendo assim

uma maior configurabilidade das mesmas. O número de respostas de cada questão pode ser expandido ou retraído em qualquer momento, sendo que o número mínimo de respostas necessárias se situa nas duas. É importante também salientar que todos os *inputs* relativos as questões e as suas respostas, suportam expressões Matemáticas desenvolvidas em *LaTeX*.

Depois de criadas, estas questões ficam disponíveis para serem seleccionadas para qualquer Jogo através de uma simples seleção.

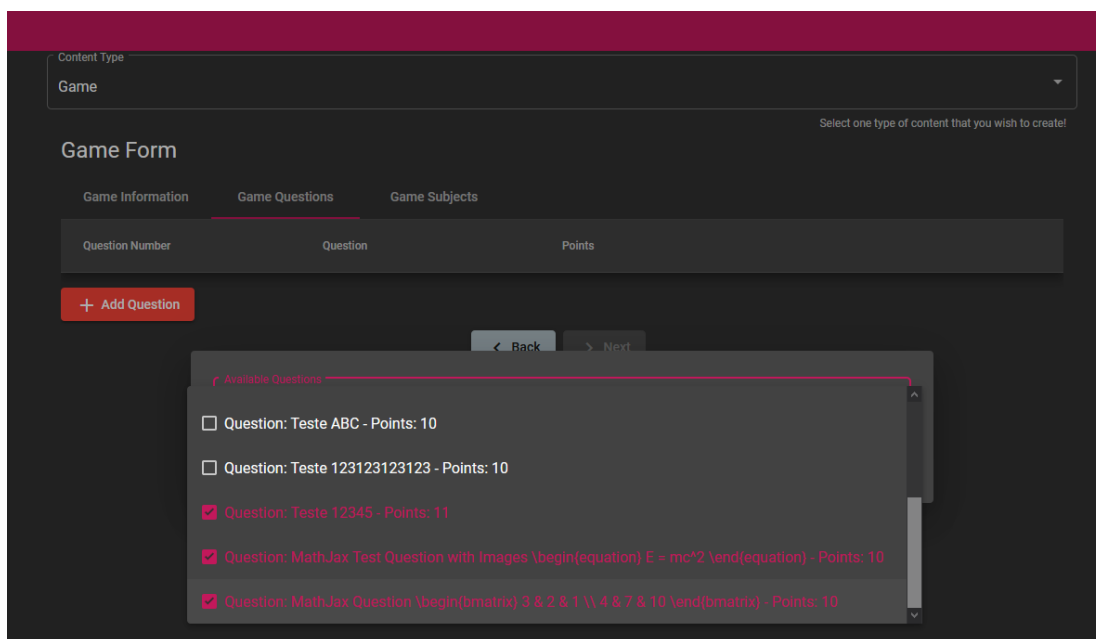


Figura 87 – Seleção de questões previamente criadas para um dado Jogo, no *Back Office*

Assim que seleccionadas e inseridas, a lista de questões pode ser gerida de livre vontade, permitindo a eliminação ou a alteração das posições das questões através do uso dos ícones apropriados.

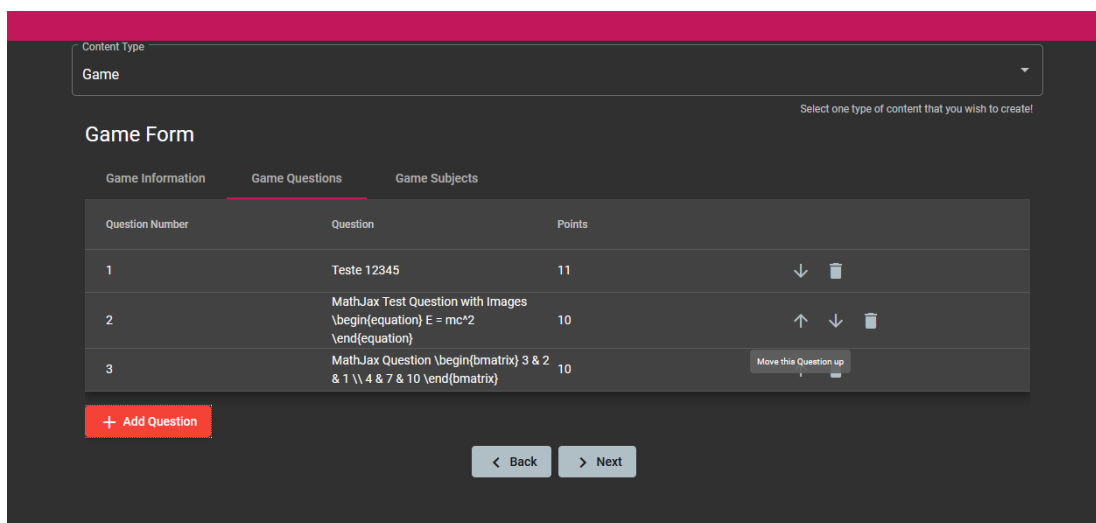


Figura 88 – Gestão das questões existentes num dado Jogo, no *Back Office*

Por fim ao contrário do que sucedido nos sistemas anteriores e enunciado anteriormente, os jogos didáticos estariam agora associados a um ou vários módulos e não a uma dada Área. Assim no seu processo de criação o Jogo poderia ser associado a um número infinito de módulos, sendo que os seus pontos de atribuição seriam pré-preenchidos pelo sistema, como o somatório das pontuações de cada questão.

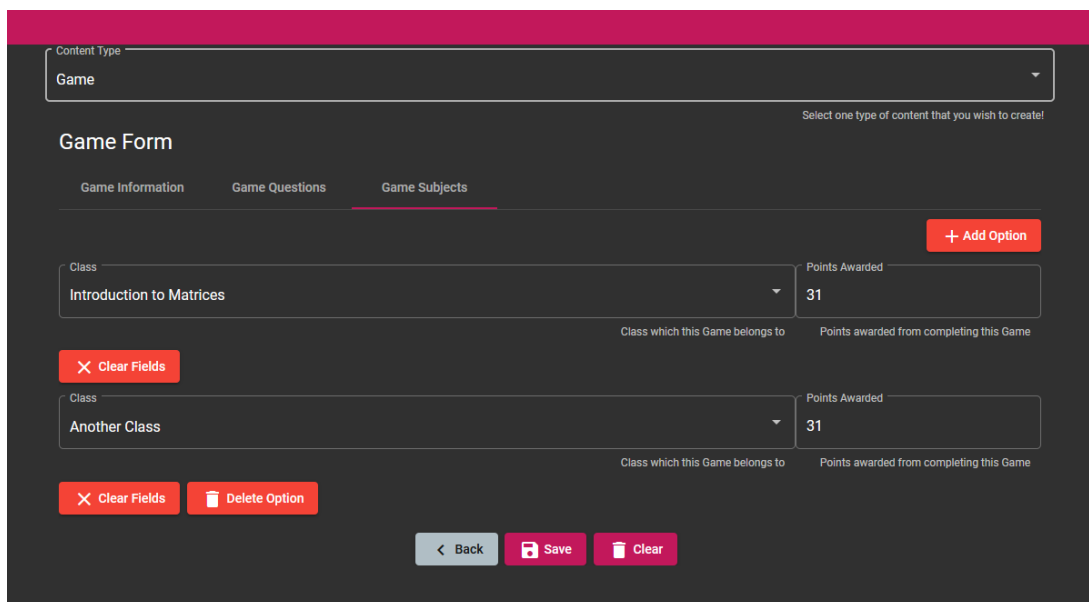


Figura 89 – Definição dos módulos aos quais um dado Jogo pertence, no *Back Office*

Assim que concluído o processo, estes jogos didáticos estariam disponíveis para os seus utilizadores do *Front Office*, dentro de cada módulo respetivo.

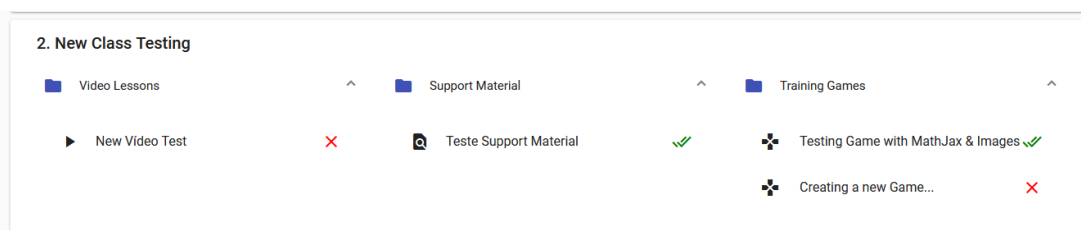


Figura 90 – Listagem dos jogos didáticos de um módulo, sobre “Training Games”, no *Front Office*

Quando o utilizador seleciona um dado jogo didático, é apresentada uma nova página com uma simples janela informativa, informando o utilizador da dificuldade do Jogo selecionado seguida de uma pequena sobre o objetivo do jogo didático, permitindo depois o utilizador avançar para o mesmo Jogo ou recuar se assim o pretender.

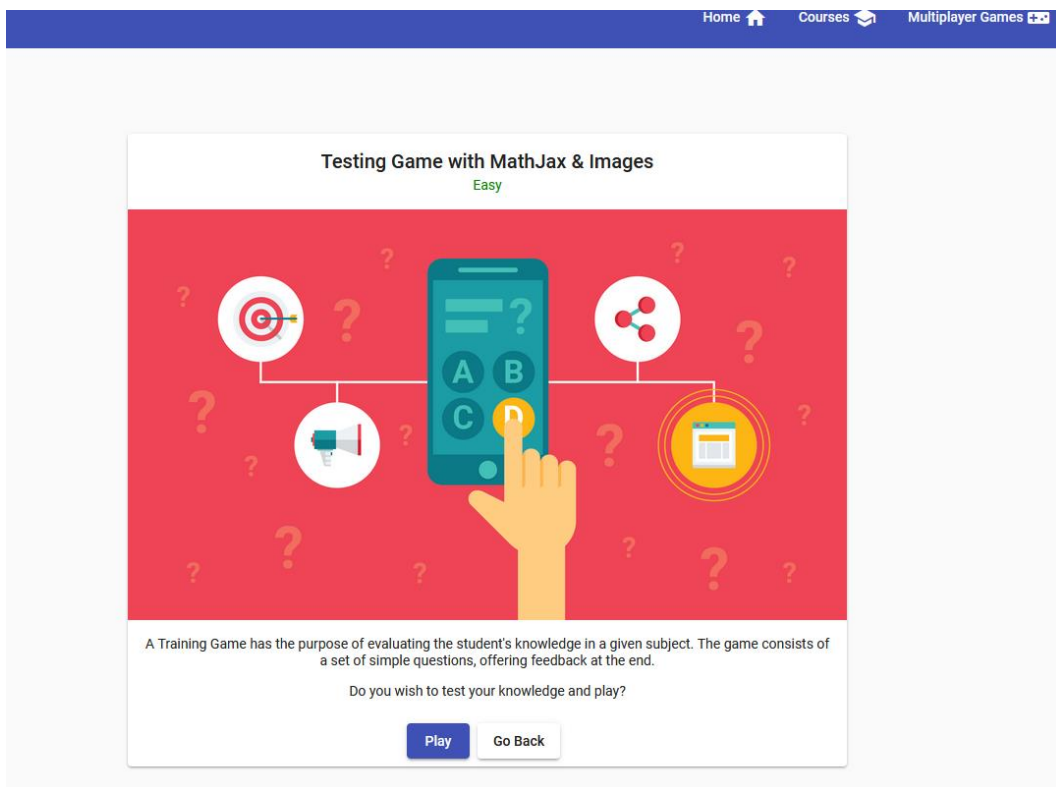


Figura 91 – Janela informativa de um dado jogo didático, no *Front Office*

Assim que iniciado o utilizador é apresentado com uma nova interface onde pode visualizar cada questão individualmente assim como as suas respetivas respostas. Para além de cada questão, o utilizador é ainda informado do número da questão em que se encontra, o tempo decorrido até ao momento e uma pequena barra de progresso que permite guiar o utilizador no seu progresso.

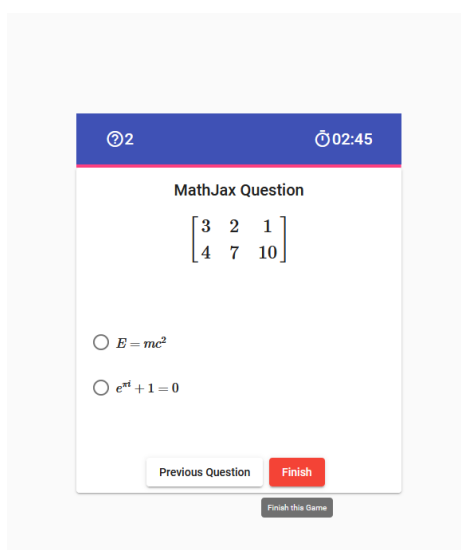


Figura 92 – Interface de um jogo didático, no *Front Office*

Importante realçar que o sistema se encontra preparado para alertar o utilizador em situações em que alguma questão se encontre por resolver, seja esta por opção ou por esquecimento do utilizador.

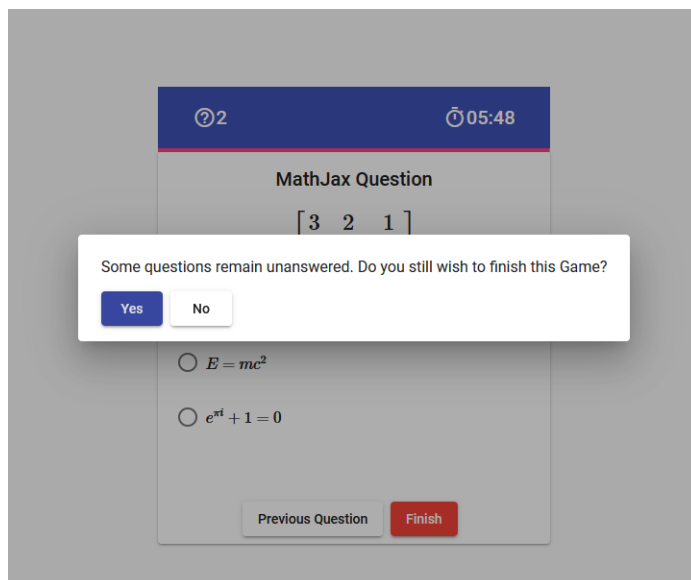


Figura 93 - Mensagem de aviso sobre questão não respondidas, no *Front Office*

Depois de terminado o Jogo, uma nova mensagem é apresentada com a pontuação obtida assim como o número de questões corretas.

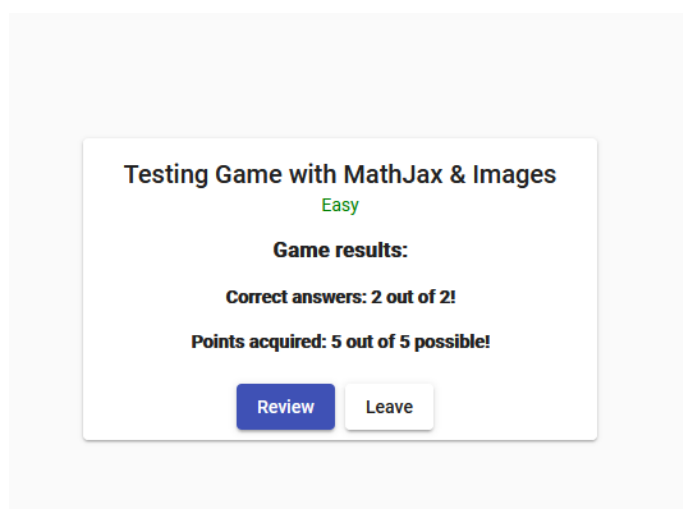


Figura 94 – Mensagem de fim de jogo didático, no *Front Office*

Quando a pontuação obtida não é a total, o utilizador é informado que pode repetir o jogo em qualquer ocasião, para que possa ganhar a totalidade de pontos oferecidos pelo próprio Jogo.

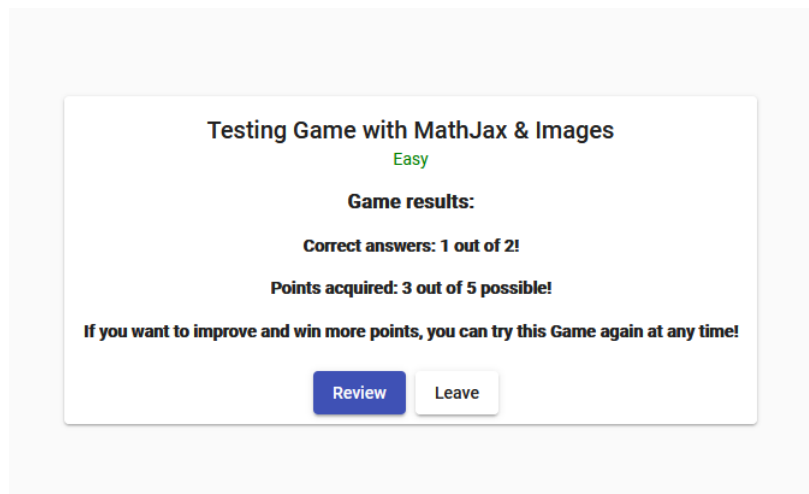


Figura 95 - Mensagem de fim de Jogo quando a pontuação obtida não é a máxima, no *Front Office*

O utilizador pode ainda no final do Jogo fazer uma revisão das suas respostas. No entanto tendo em conta a finalidade deste tipo de Jogos e a possibilidade de repetição em qualquer altura, condiciona a informação disponível na revisão. Como tal na revisão do jogo apenas é informado ao utilizador se a resposta a uma dada questão se encontra correta ou errada, e no caso da ultima opção é omitida a resposta correta, permitindo assim que o utilizador reinicie novamente o Jogo quando desejar sem saber à priori as respostas corretas as questões para as quais errou. É possível ainda observar em cada resposta, para além da sua validação, o número de pontos obtidos pela resposta selecionada.

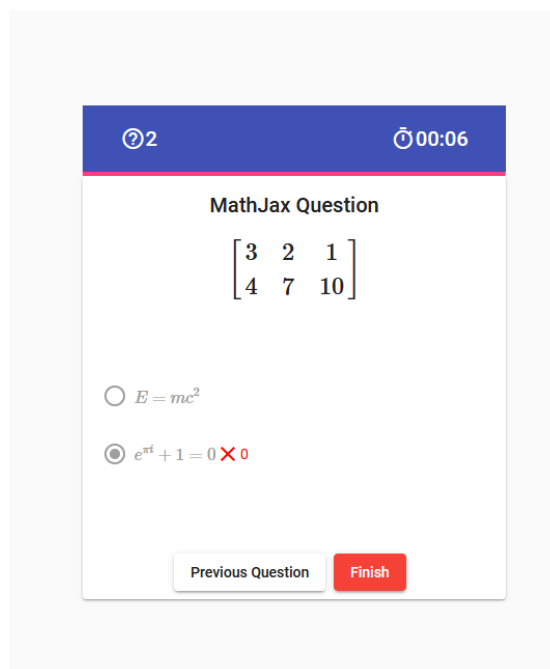


Figura 96 - Revisão de um jogo didático, no *Front Office*

Quando terminado, tal como os outros conteúdos existentes no respetivo módulo, a indicação visual dos pontos alcançados nesse dado Jogo seria atualizada. No entanto esta pontuação apresentada será sempre a pontuação mais alta do utilizador e não a mais recente. Podem coincidir, no entanto, considerou-se pertinente que a melhor pontuação obtida pelo utilizador nesse Jogo seja a pontuação apresentada.

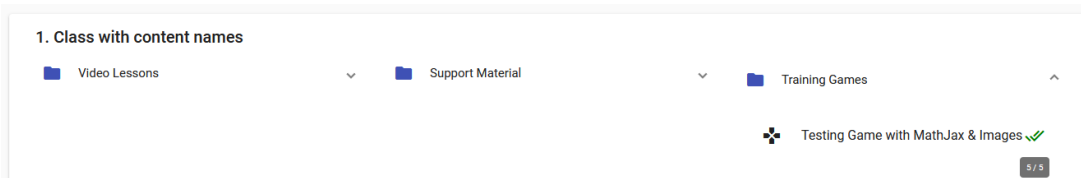


Figura 97 - Indicação visual da melhor pontuação de um utilizador num dado Jogo, no *Front Office*

Quanto ao seu processo de edição no *Back Office*, este mais uma vez segue os moldes do processo de criação, necessitando como esperado uma seleção previa do Jogo a editar.

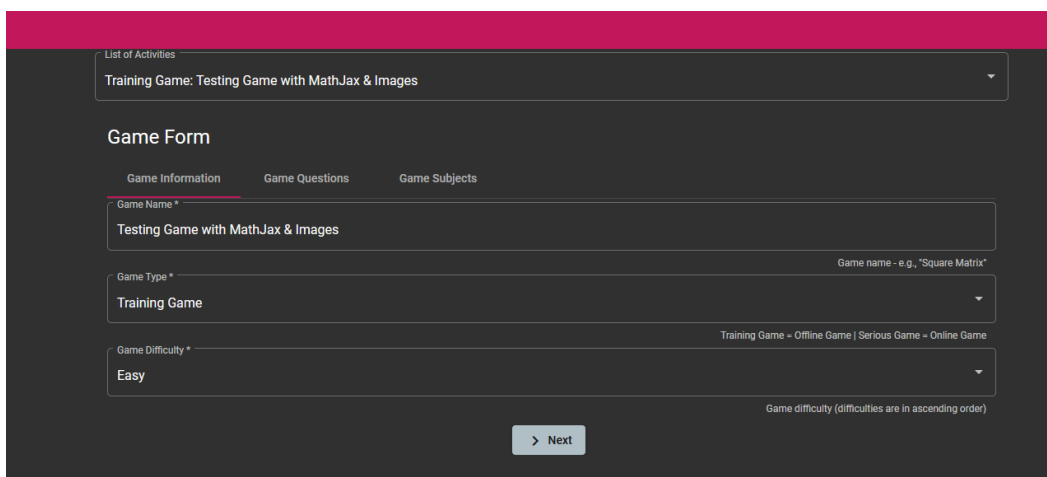


Figura 98 - Edição do Jogo "Testing Game With MathJax & Images", no *Back Office*

7.1.2 Configurabilidade de autenticação e autorização

Para o controlo de autenticações e de autorizações na plataforma foi necessário construir processos de raízes que permitissem atingir os objetivos pretendidos. Infelizmente os sistemas anteriores, apesar de admitirem a existência de "roles" para os utilizadores da plataforma (cf. secção 3.3.2), estes acabaram por não ser desenvolvidos, ficando extremamente limitados.

Antes de implementar este último processo, foi necessário tratar da acessibilidade ao *Back Office*. Tratando-se de uma plataforma de acesso bastante limitada, sendo esperado apenas o acesso de um número reduzido de utilizadores, é importante que a mesma se encontre segura de acessos alheios e não autorizados.

Para isso, apenas os supervisores e administradores do sistema tem a possibilidade de em qualquer momento convidar um utilizador para aceder a plataforma do *Back Office*, bastando apenas indicar o email do utilizador pretendido. Podem ser convidados novos utilizadores, como utilizadores já registados na plataforma do *Front Office*.

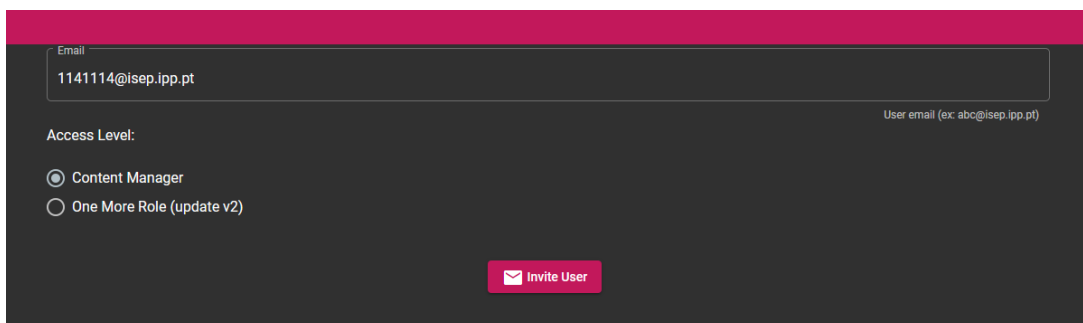


Figura 99 - Convite de um utilizador para aceder ao *Back office*, no *Back Office*

Depois de introduzido o email do respetivo utilizador e do seu nível de acesso, que será abordado em mais detalhe posteriormente, o utilizador convidado recebe um email personalizado, alertando o mesmo para o convite efetuado.

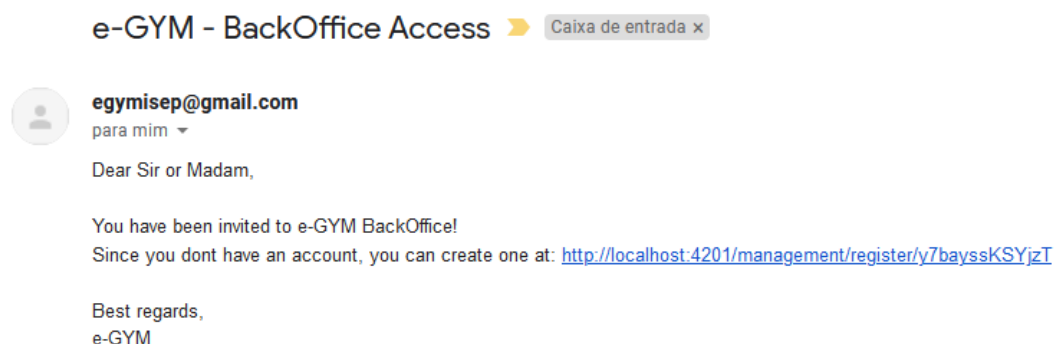


Figura 100 – Email recebido por utilizadores convidados

Para os níveis de acesso, ou os denominados “*roles*”, o processo criado procura desde início apresentar-se bastante dinâmico tendo em conta as especificações da plataforma. Como tal e tendo em conta que esta mesmo se baseia em diferentes tipos de ações, apresentadas numa *sidebar*¹⁷ lateral, tornou-se imperativo desde início manter esta sempre em consideração, pois com a adição ou remoção constante de funcionalidade torna-se importante que estas mesmas pudessem estar logo disponíveis nos processos de autorização.

¹⁷ Barras laterais com finalidades de navegação.

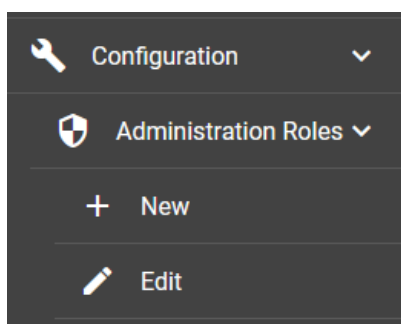


Figura 101 – Navegação para a criação de novas *roles*, no *Back Office*

Para tal é assim permitido aos supervisores do *Back Office*, que tenham autorização para tal, a criação de novos “*roles*”, fornecendo os seguintes parâmetros:

- **Nome:** nome do “*role*”, para que este possa ser reconhecido.
- **Descrição:** pequeno texto que permita descrever o novo “*role*”, de carácter opcional.
- **Propriedades de acesso:** regras de acesso para o dado “*role*”, que dinamicamente carrega todas as opções existentes na *sidebar*.

A screenshot of the 'Administration Roles Configuration' form. It has a dark background with a pink header. The form contains three main sections: 1. 'Name' field with the value 'Content Manager' and a 'Role Name' label. 2. 'Description' field with the text 'This role should be assigned to content managers...' and a 'Role Description' label. 3. 'Access Properties' section with a tree view: 'Areas' (unchecked), 'Activities' (checked) containing 'New', 'Edit', and 'Delete' (all checked), 'Competitions' (unchecked), 'Statistics' (checked) containing 'Overview', 'Area Stats', 'Content Stats', and 'Game Stats' (all checked), and 'Configuration' (unchecked). A pink 'Save' button is at the bottom right.

Figura 102 – Criação de um novo nível de acesso, no *Back Office*

Depois de criados, estes *roles* poderiam ser atribuídos aos utilizadores com acesso ao *Back Office*, sendo necessário apenas seleccionar o utilizador pretendido, assim como o seu novo nível de acesso.

User
admin admin - adminadmin@teste.egym.pt

Email
adminadmin@teste.egym.pt

Access Level:

Content Manager

One More Role (update v2)

Save Changes

Figura 103 – Atribuição de um *role* a um utilizador do *Back Office*, no *Back Office*

Os utilizadores teriam assim as suas opções de navegações laterais, a *sidebar*, restringidas baseadas nas restrições do seu *role*.

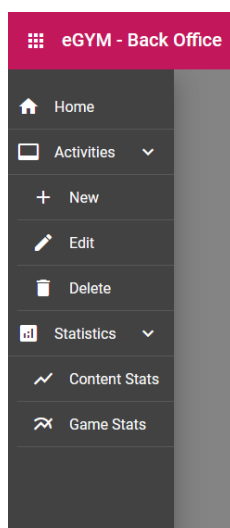


Figura 104 - Navegação lateral restrita, no *Back Office*

Por fim e como medida de proteção dos conteúdos disponibilizados na plataforma do *Front Office*, foi introduzida uma nova identidade denominada de grupo de acesso. Estes grupos de acesso representam grupos de inúmeros utilizadores, permitindo assim de uma forma eficaz limitar o acesso aos conteúdos a um grande número de utilizadores utilizando apenas estas grupos como referência.

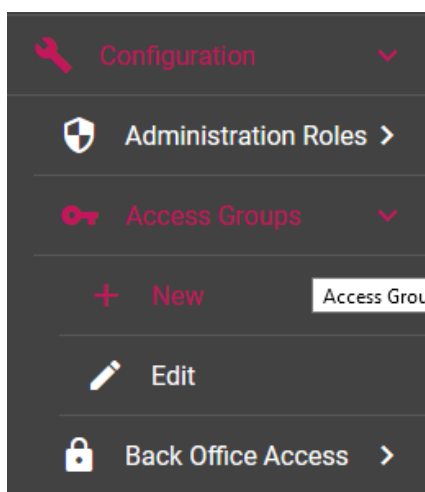


Figura 105 - Grupos de acesso, no *Back Office*

Para a sua criação são apenas necessários os seguintes parâmetros:

- **Nome:** nome do grupo, para efeitos de reconhecimento.
- **Utilizadores:** utilizadores que pertencem a este grupo, podendo ser alterado em qualquer momento.

A form for creating a new access group. It has a dark background with a red header. The form fields are: 'Group Rule Name *' with the value 'Estat Geo UM'; 'Users' with the value 'João Fernandes, Jorge Mendonça, Iurdes babo, Carla Pinto'; and a checkbox for 'Default Group' which is checked. There is a 'Save' button at the bottom right. A tooltip for the 'New' button from the previous image is visible over the 'Users' field. A note at the bottom left says 'Newly registered Users will automatically be assigned to this Group'. A note at the bottom right says 'Select users (You can search by first/last name or email)'. A note at the top right says 'Group Rule Name (e.g., - Freemium Users)'.

Figura 106 - Criação de um Grupo de Acesso, no *Back Office*

Para além destes parâmetros, é ainda possível definir se o grupo de acesso criado é o grupo por defeito. Com esta seleção o grupo criado para a ser considerado pelo sistema como o grupo designado para todos os novos utilizadores registados a partir daquele preciso momento.

Com a criação destes grupos, era assim agora possível limitar o conteúdo observado pelos utilizadores da plataforma do *Front Office*. Os utilizadores da plataforma do *Back Office* poderiam agora criar conteúdos e atividades, tendo em conta um grupo de utilizadores específicos.

Digamos que a plataforma se encontra num futuro próximo a espera de utilizadores estrangeiros. Com estes grupos, é agora facilmente possível desenvolver conteúdos e atividades direcionadas aos últimos, sendo depois necessário apenas definir os grupos que

podem aceder a estes conteúdos. Esta definição aconteceria no seu próprio processo de criação ou então em páginas reservadas para tal.

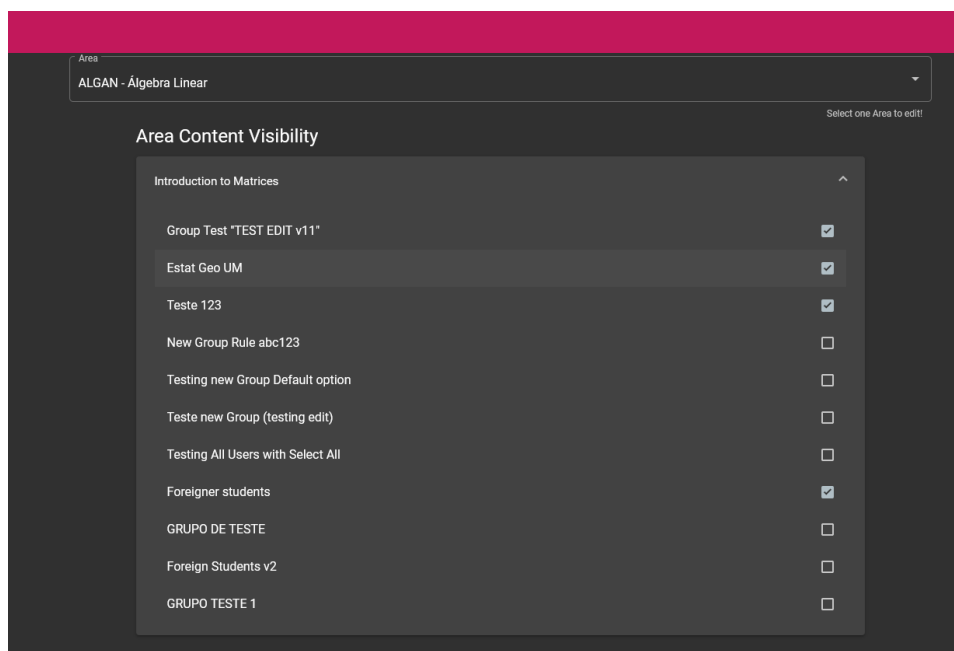


Figura 107 - Definição dos grupos de acesso para um dado módulo dentro de uma Área, no *Back Office*

7.1.3 Inexistência de jogos multijogador

Quanto a implementação dos jogos multijogador, esta demonstrou desde a sua fase inicial, alguma dificuldade acrescida devido a necessidade de sincronização entre todos os jogadores.

Antes de abordar o processo de sincronização é importante descrever o processo de criação destes jogos multijogador. Durante o processo de criação de um Jogo, seja este didático ou de multijogador, é preciso de definir um tipo de jogo, encontrando-se os seguintes tipos disponíveis:

- **Jogo de Treino:** tipo de jogo associado aos jogos didáticos, também conhecidos por Jogos de um único utilizador.
- **Jogo Sério:** tipo de jogo reservado para os jogos multijogador.

Como tal, quando um utilizador pretender criar um Jogo, este deve ter sempre em consideração o tipo de jogo que pretende criar. Como é possível observar na Figura 98, o jogo didático “*Testing Game with MathJax & Images*” tem no seu tipo de jogo o primeiro tipo enunciado anteriormente, neste caso em tradução literal para o Inglês “*Training Game*”.

Para os jogos multijogador o processo utilizado é o mesmo que o utilizados nos jogos didáticos, no entanto o utilizador deve inserir o segundo tipo enunciado anteriormente, ou a sua tradução literal dependente da localização utilizada. Com esta escolha do novo tipo surge ainda um novo parâmetro, denominado de tempo limite, que permite ao utilizador definir um

tempo máximo para o jogo ser completado, em segundos. Esta parâmetro tem como finalidade introduzir alguma “pressão” aos utilizadores que se encontrem a resolver um dado Jogo multijogador, tendo em conta o objetivo destes mesmos. É de realçar que este parâmetro, apesar de presente, não é necessariamente obrigatório. Caso o utilizador assim o pretenda, pode deixar o respetivo campo em branco, fazendo com o que o Jogo multijogador criado não possua nenhum tempo limite.

Content Type
Game

Select one type of content that you wish to create!

Game Form

Game Information Game Questions Game Subjects

Game Name *
Example of an Online Game
Game name - e.g., "Square Matrix"

Game Type *
Serious Game
Training Game - Offline Game | Serious Game - Online Game

Game Difficulty *
Hard
Game difficulty (difficulties are in ascending order)

Game Time Limit
600
Game time limit in seconds (If no value is inputted, no time limit will be considered)

> Next

Figura 108 – Exemplo do processo de criação de um Jogo multijogador, no *Back Office*

A gestão das questões inseridas dentro deste Jogo multijogador segue os mesmos princípios implementados para o jogo didáticos.

Content Type
Game

Select one type of content that you wish to create!

Game Form

Game Information Game Questions Game Subjects

| Question Number | Question | Points | |
|-----------------|--|--------|------|
| 1 | MathJax Test Question with Images $E = mc^2$ | 10 | ↓ 🗑️ |
| 2 | MathJax Question $\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix} & 7 & 10$ | 10 | ↑ 🗑️ |

+ Add Question Delete this Question

< Back > Next

Figura 109 – Gestão das questões existentes num Jogo multijogador, no *Back Office*

No entanto uma diferença existente na criação de um jogo multijogador para um jogo didático passa pela associação de um jogo a diferentes entidades. No caso do Jogo multijogador este

encontra-se inserido dentro de áreas enquanto que o jogo didático se encontra inserido em diferentes módulos (cf. secção 3.1.3).

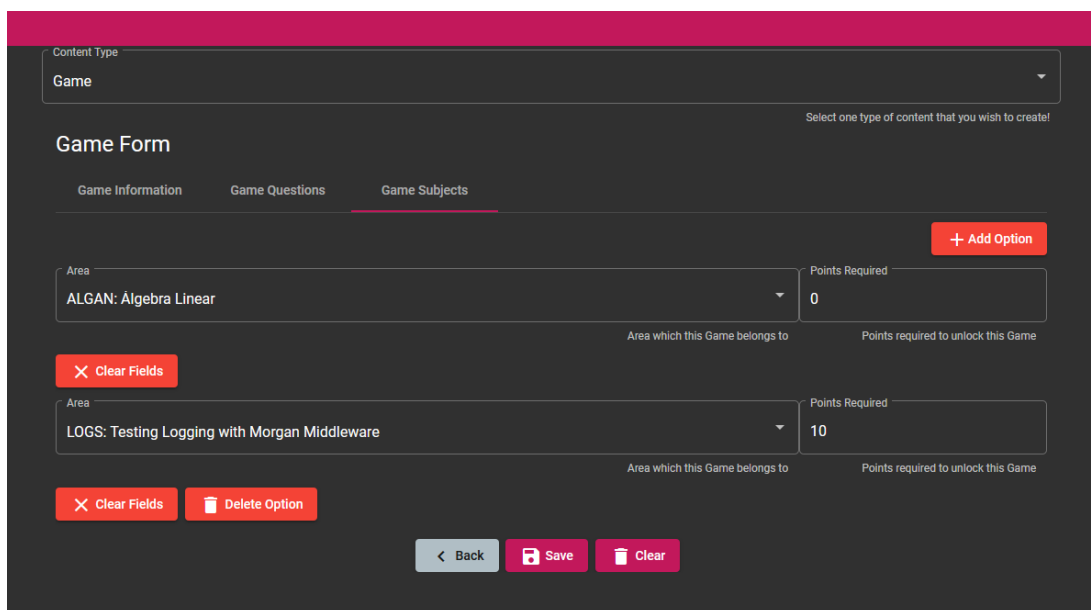


Figura 110 – Definição das áreas para os quais um Jogo multijogador se encontra inserido, no *Back Office*

Depois de criados, este tipo de Jogos passa a estar disponível para os utilizadores do *Front Office* numa página reservada, onde os Jogos se encontram distribuídos pelas áreas existentes e com jogos multijogadores disponíveis.

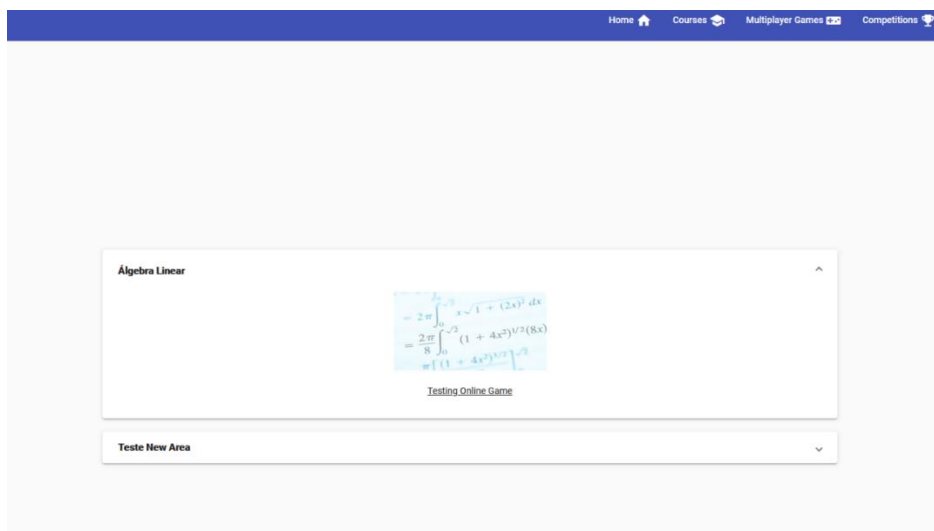


Figura 111 - Apresentação dos jogos multijogador existentes, no *Front Office*

Assim que o utilizador efetuar um clique num dado Jogo multijogador, uma nova janela de diálogo¹⁶ é aberta apresentando alguma informação sobre o Jogo.

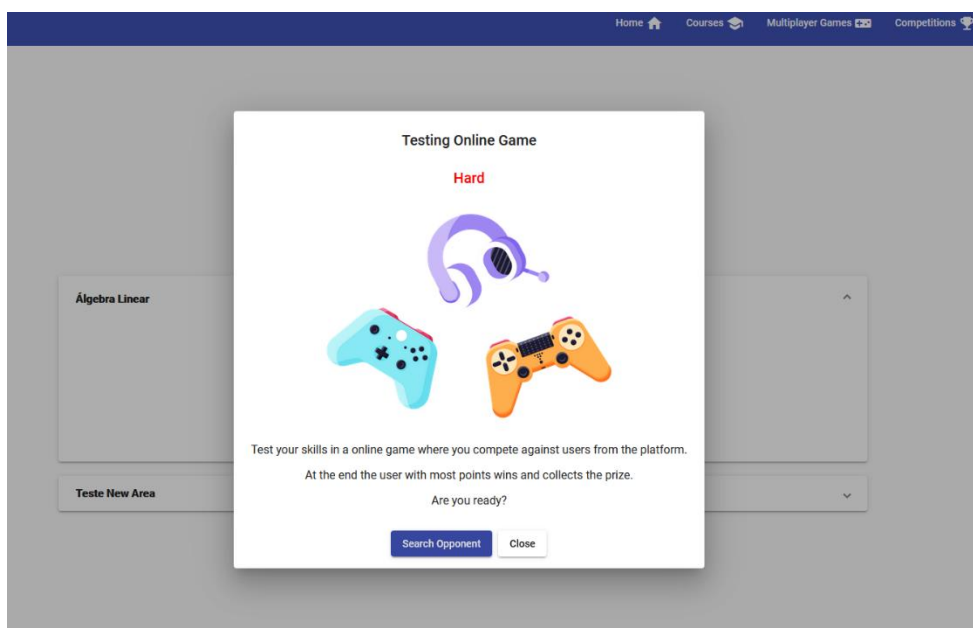


Figura 112 – Informação relativa de um Jogo multijogador, no *Front Office*

Depois de apresentada esta última janela, o utilizador pode iniciar o processo de *matchmaking*¹⁸, onde o utilizador é emparelhado com outros utilizadores que procurem jogar o mesmo Jogo multijogador naquele dado momento.

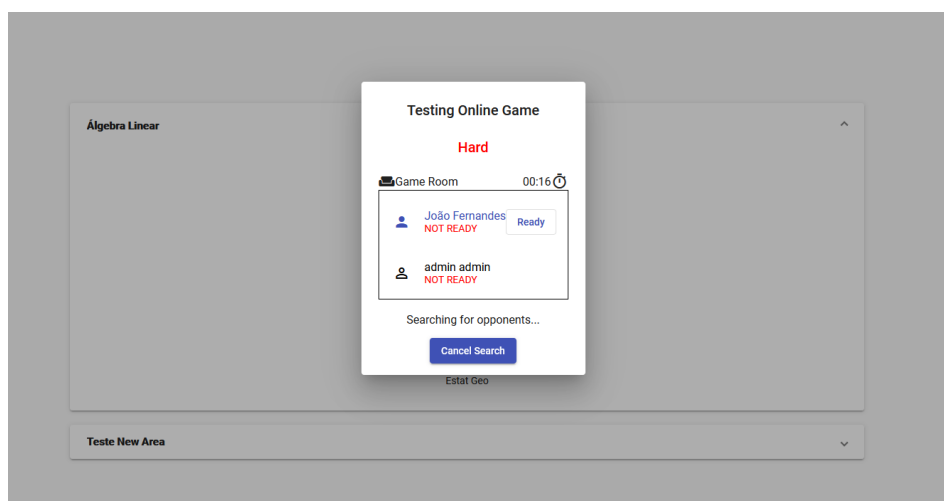


Figura 113 – Emparelhamento entre dois utilizadores para um jogo multijogador, no *Front Office*

Uma das especificações dos jogos multijogador descreve que este tipo de jogos deve ser realizado entre vários jogadores da plataforma, até um limite superior de quatro jogadores. No entanto e prevendo situações em que não seja possível efetuar este emparelhamento de quatro jogadores, o sistema permite que o jogo se inicie quando se encontrem pelo menos

¹⁸ Termo utilizado para a conexão de utilizadores em sessões de jogos online. São normalmente utilizados algoritmos que façam o emparelhamento de diversos jogadores, normalmente em salas de espera.

dois jogadores na sala de jogo. Para isto acontecer, é preciso que o sistema reconheça que num período de tempo não exista qualquer entrada ou saída de jogadores da sala, sendo para já este período quinze segundos, e se caso isso se suceder todos os jogadores da sala de jogo devem concordar com o seu início antecipado, devendo para tal sinalizar os restantes adversários que se encontram preparados para tal utilizado o botão “Ready” visível na Figura 113.

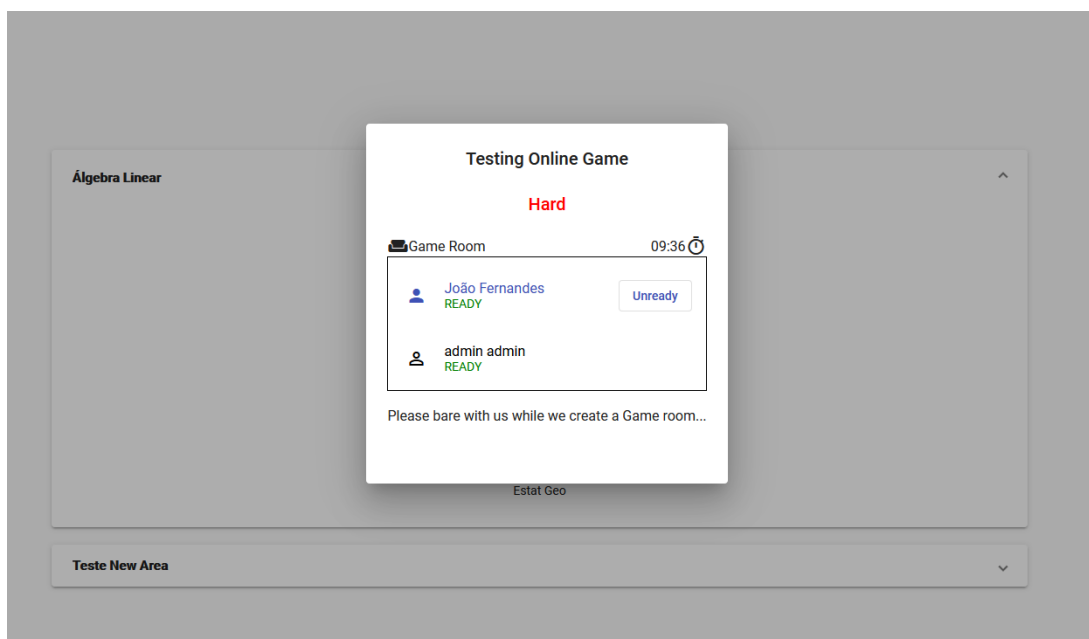


Figura 114 - Criação de uma sala de jogo antecipada por indicação dos seus utilizadores, no *Front Office*

Assim que todos os jogadores se apresentarem como preparados para iniciar o jogo ou assim que a sala de jogo atinja a sua limitação de quatro jogadores, a sala de jogo é criada na base de dados e é apresentado a todos os jogadores uma simbólica contagem de crescente para estes se prepararem para o início do jogo.

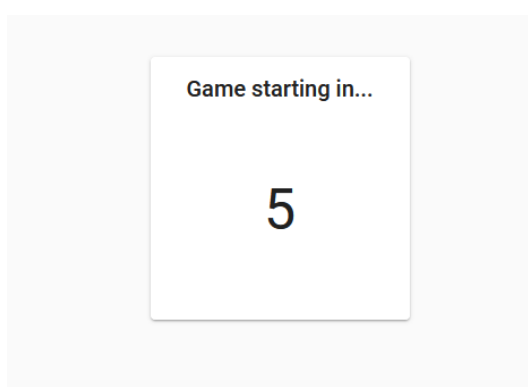


Figura 115 - Contagem decrescente para o início de um jogo multijogador, no *Front Office*

Depois de o jogo dar início, a interface visível é bastante semelhante à utilizada nos jogos didáticos (cf. secção 7.1.1.3) no entanto é possível agora observar uma pequena zona que

acompanha todo o progresso dos utilizadores deste jogo, apresentando para cada um deles uma pequena barra de progresso que identifica o número de questões que um utilizador já respondeu. Para um dado utilizador, todos os seus oponentes são representados pelo ícone de fundo branco, enquanto que o seu próprio progresso é identificado pelo ícone de fundo preto ou pela indicação na barra superior, existente do lado esquerdo. Se o utilizador abandonar a partida mais cedo, um novo ícone de aviso será acrescentado perto da barra de progresso desse utilizador, notificando assim os restantes da sua saída.

Para além desta indicação é ainda possível observar do lado direito, o tempo limite deste mesmo jogo e ao centro o tempo decorrido, sendo que a cor deste altera consoante a sua aproximação ao tempo limite.

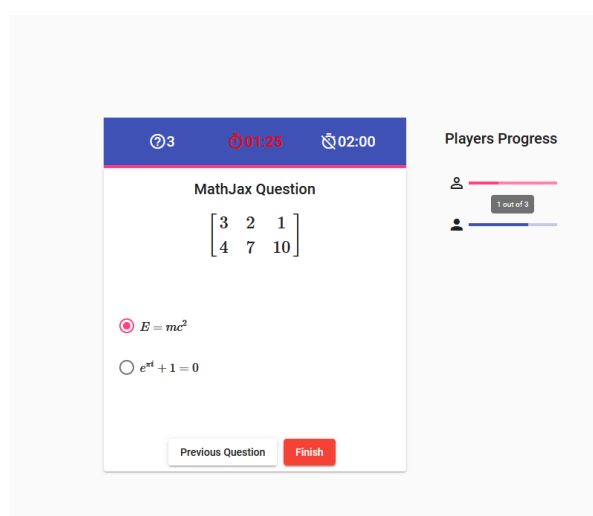


Figura 116 - Interface de um jogo multijogador, no *Front Office*

Quando um dado utilizador termine o seu jogo e ainda não se tenha determinado um vencedor, o *Front Office* apresenta uma simples mensagem interativa a todos os jogadores que tenham terminado o jogo e se encontrem a espera da determinação do resultado.

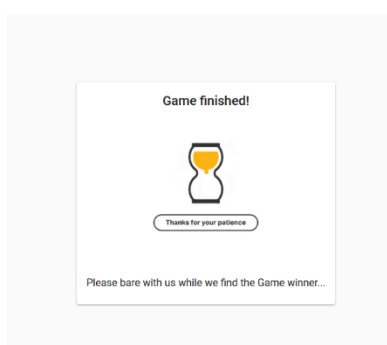


Figura 117 – Mensagem apresentada aos utilizadores que se encontrem a espera da determinação do vencedor, no *Front Office*

Assim que um vencedor for determinado pelo servidor, neste caso pela *Game API*, o *Front Office* procede ao tratamento de dados recebido pelo servidor e apresenta uma lista com a

classificação final, mostrando para cada utilizador o número de pontos alcançados e o tempo utilizado. Esta classificação é acompanhada por uma simples mensagem e de uma imagem alusiva tendo em conta a classificação do mesmo.

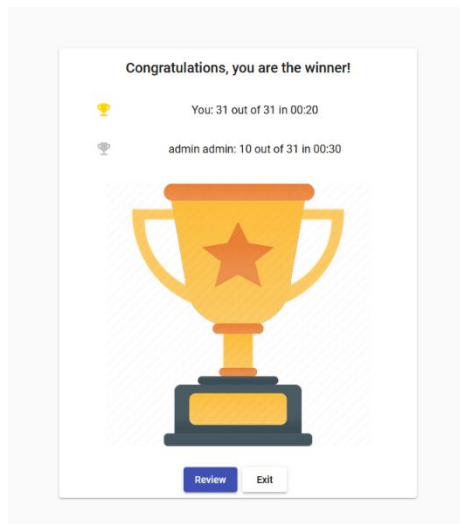


Figura 118 - Mensagem apresentada ao vencedor de um jogo multijogador, no *Front Office*

O utilizador pode depois, tal como nos jogos didáticos, efetuar uma revisão do mesmo. Mais uma vez apenas é apresentada *feedback* se a resposta efetuada pelo utilizador for a correta, deixando assim espaço para o utilizador melhorar a *performance* sem saber a resposta correta no caso de errar.

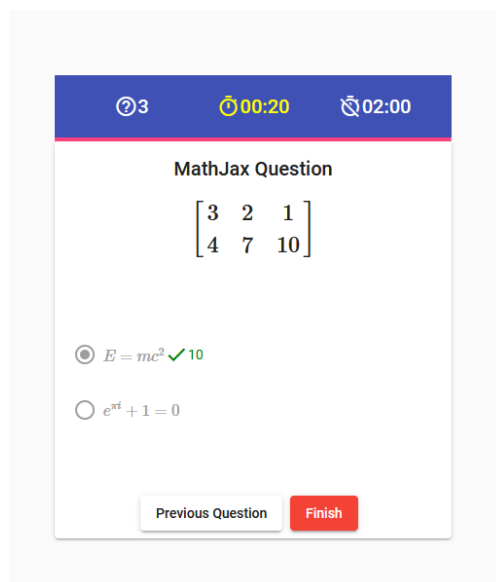


Figura 119 – Revisão de um jogo multijogador, no *Front Office*

7.1.4 Introdução de competições

Para esta limitação, foi necessário construir um subsistema de raiz, tendo em conta que nenhuma das versões anteriores ofereciam a possibilidade aos seus utilizadores de participar em competições.

Quanto aos processos reservados à administração do sistema, neste caso ao processo de criação de uma competição, a sua estrutura é semelhante ao implementado para limitações anteriores. Para uma competição foram deduzidos os seguintes parâmetros:

- **Nome:** nome da competição, para a qual esta será reconhecida.
- **Data de início:** data em que a competição começa, em formato data-hora.
- **Data de fim:** data em que a competição termina, no formato data-hora.
- **Data de início para registo:** data em que o registo para a competição começa, em formato data-hora.
- **Data de fim para registo:** data em que os registos para uma competição terminam, em formato data-hora.

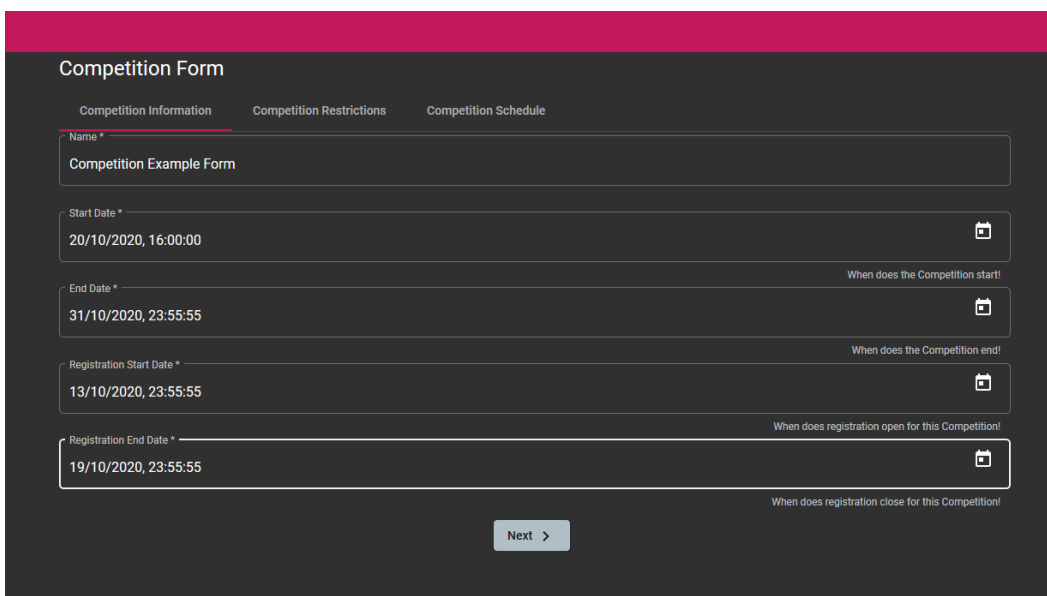


Figura 120 – Exemplo de uma criação de uma Competição, no *Back Office*

Tendo em conta os objetivos de uma Competição no sistema pretendido, assim como o seu significado literal, foi considerado que todos os parâmetros reservados para formatos data deveriam ser estendidos para formatos data-hora. Esta mudança permite assim uma maior especificação dos mesmos, permitindo definir períodos mais específicos. Para agilizar o seu processo de criação todos estes campos apresentam também validações internas sobre possíveis conflitos entre datas, como por exemplo a definição de uma data de fim que seja inferior à sua respetiva data de início.

The screenshot shows a 'Competition Form' with the following details:

- Competition Information** (selected tab)
- Name ***: Competition Example Form
- Start Date ***: 20/10/2020, 16:00:00
- End Date ***: 31/10/2020, 23:55:55
- Registration Start Date ***: A calendar is open for October 2020, with the 13th selected. Below the calendar, the time is set to 23:55:55.
- Next >** button

Figura 121 - Definição dos parâmetros data-hora para uma Competição, no caso específico da data de início para registos, no *Back Office*

Para além destes parâmetros “básicos”, o utilizador pode ainda definir, dois parâmetros adicionais:

- **Número máximos de utilizadores:** número máximos de inscrições permitidas para a Competição.
- **Grupos de acesso:** grupos de utilizadores que podem visualizar e aceder à Competição.

O primeiro destes dois surge na necessidade de oferecer um maior controlo e configurabilidade das Competições, permitindo colocar um teto no número de utilizadores que podem participar numa dada Competição, ou pelo contrário não colocar nenhum limite deixando este valor vazio.

O segundo aparece naturalmente com a implementação dos processos de autorização abordados anteriormente (cf. secção 7.1.2), oferecendo assim um maior controlo sobre os utilizadores que podem visualizar a Competição criada na plataforma *Front Office*.

The screenshot shows the 'Competition Form' interface. At the top, there are three tabs: 'Competition Information', 'Competition Restrictions' (which is active), and 'Competition Schedule'. Below the tabs, there is a form with two main sections. The first section is 'Maximum Amount of Users (Leave empty if no limit exists)' with a text input field containing the number '100'. The second section is 'Access Groups' with a dropdown menu showing 'Estat Geo UM (+1 others)'. A small note below the dropdown says 'You can limit the number of users that can participate in this Competition!'. At the bottom of the form, there are two buttons: '< Back' and 'Next >'.

Figura 122 – Definição de parâmetros adicionais no processo de criação de uma Competição, no *Back Office*

Por fim é ainda necessário definir a calendarização da própria Competição. Para tal é possível definir dias, em formato data-hora mais uma vez, para os quais questões se encontram disponíveis para os seus utilizadores as resolverem. A estrutura idealizada e construída, permite assim aos supervisores definir vários dias de atividade, em que para cada podem ser definidas várias questões com as suas próprias regras. De uma maneira sintetizada a estrutura é seguinte:

- **Data agendada:** define um dia, no formato data-hora, em que existem questões para os utilizadores registados de uma Competição resolver.
 - **Questão:** questão existente para uma dada data agendada, possuindo as suas próprias características e restrições.

The screenshot shows the 'Competition Form' interface, specifically the 'Competition Schedule' tab. At the top, there are three tabs: 'Competition Information', 'Competition Restrictions', and 'Competition Schedule' (which is active). Below the tabs, there is a form with two main sections. The first section is 'Schedule Date *' with a text input field containing '20/10/2020, 18:00:00'. To the right of this field is a red button labeled '+ Add Schedule Entry'. The second section is 'Question *' with a dropdown menu showing 'A probabilidade do aluno chegar atrasado dado que utiliza o m...'. To the right of this dropdown is a red button labeled '+ Add Question Entry'. At the bottom of the form, there are three buttons: '< Back', 'Clear', and 'Save'.

Figura 123 – Definição da calendarização de uma Competição, no *Back Office*

Cada questão incluída dentro de uma data agendada, possui um número de parâmetros configuráveis:

- **Número máximo de tentativas:** valor número que permite restringir o número máximos de tentativas de uma questão. Por defeito o número de tentativas máximas permitidas é igual a 1.

- **Tempo de fecho:** parâmetro data-hora que permite aos utilizadores do *Back Office* definirem um tempo limite para o qual a respetiva questão pode ser respondida. Por defeito se o parâmetro ficar por preencher, o tempo limite para uma questão corresponde a data de fim da Competição.

The screenshot displays the 'Competition Form' interface, specifically the 'Competition Schedule' tab. The form is set against a dark background with a magenta header. At the top, there are three tabs: 'Competition Information', 'Competition Restrictions', and 'Competition Schedule'. A red button labeled '+ Add Schedule Entry' is located in the top right corner. The main form area contains several fields: 'Schedule Date *' with the value '20/10/2020, 18:00:00', a 'Question *' field with the text 'A probabilidade do aluno chegar atrasado dado que utiliza o m...', a 'Number of attempts possible' field with the value '1', and a 'Closing Time' field with the value '21/10/2020, 23:55:00'. There are also two red buttons: '+ Add Question Entry' and a 'Save' button. At the bottom, there are three buttons: '< Back', 'Clear', and 'Save'.

Figura 124 – Configuração de uma questão para uma Competição, no *Back Office*

É naturalmente possível adicionar múltiplas datas agendadas assim como múltiplas questões dentro de cada data agendada. Para tal o utilizador necessita de apenas utilizar os respetivos botões, visualizados na Figura 124 como “*Add Schedule Entry*” e “*Add Question Entry*”.

Figura 125 - Adição de múltiplas datas agendadas e as suas respetivas questões para uma Competição, no *Back Office*

Quanto ao *Front Office*, depois de criada esta fica visível para os utilizadores com grupos de acesso seleccionados durante o seu processo de criação. Na sua página reservada, as Competições existentes encontram-se divididas em três grupos distintos:

- **Minhas Competições:** grupo de Competições para as quais o utilizador se encontra registado.
- **Competições abertas a registo:** grupo de Competições para as quais o utilizador se pode registar. Nesta lista aparecem Competições para as quais o utilizador ainda não se tenha registado e que o dia atual se encontre dentro dos limites de datas para o registo da mesma.
- **Competições agendadas:** grupo de Competições que se encontram agendadas para um futuro próximo, mas que, no entanto, ainda não se encontram abertas a registos de utilizadores.

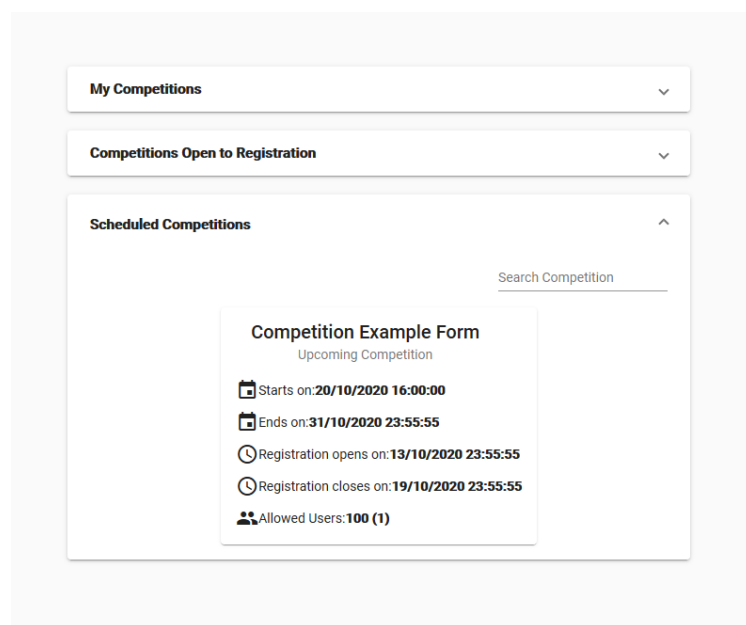


Figura 126 - Listagem de Competições agendadas, no *Front Office*

Para além da impossibilidade de realizar alguma ação, é possível ainda observar um filtro de pesquisa através da Figura 126. Com este filtro o utilizador pode facilmente filtrar Competições pelo seu nome. Neste painel estão apresentadas todas as Competições agendadas para um futuro, mas que, no entanto, ainda não permitam qualquer tipo de ação extra.

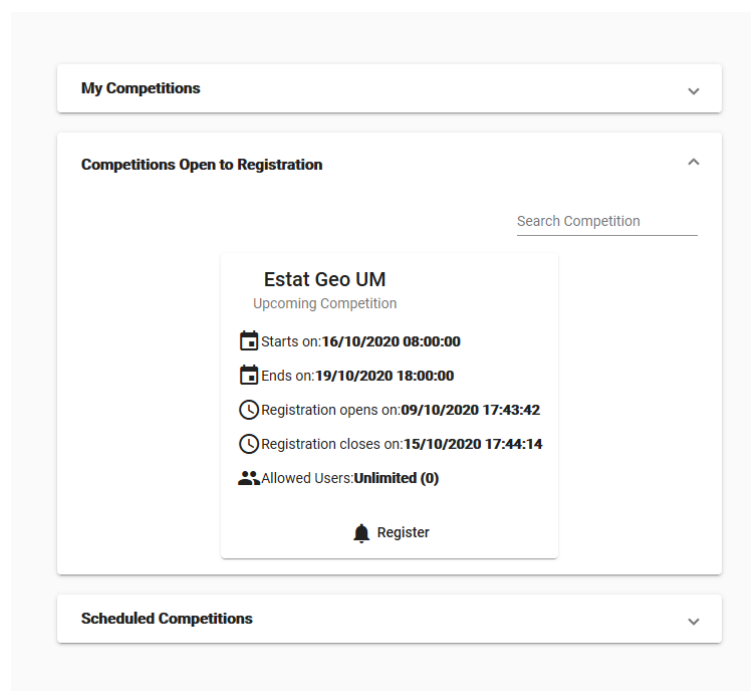


Figura 127 - Listagem de Competição abertas para registo, no *Front Office*

Quanto as Competições abertas para Registo, o utilizador pode aqui através do botão apropriado realizar o seu registo para a mesma. Caso o número de utilizadores registados já iguale o número de utilizadores permitidos, a Competição passa a notificar os seus utilizadores deste caso, impossibilitando registos adicionais através do bloqueio do botão de registo. É sempre possível observar o número de utilizadores já registados, estando este identificado pelos limitadores “()”, como demonstrado na Figura 127.

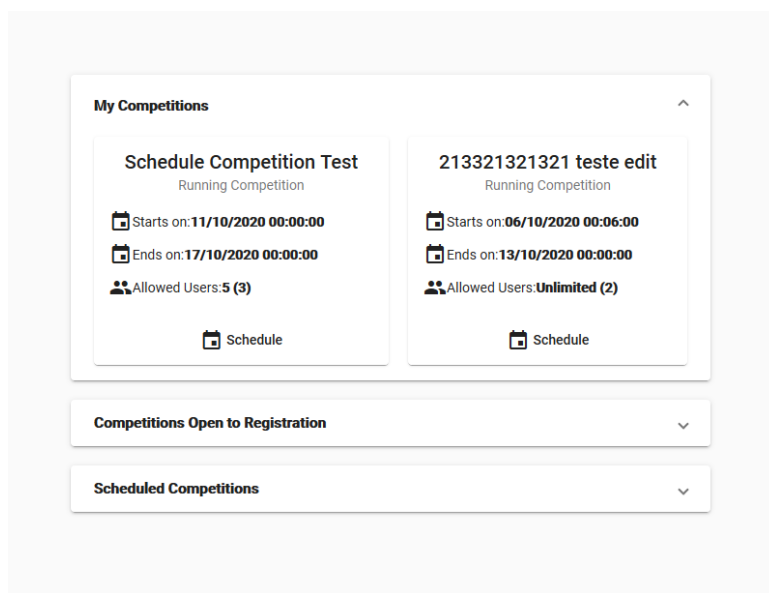


Figura 128 - Listagem de Competições ativas de um utilizador, no *Front Office*

Para as Competições onde o utilizador esteja registado, estas são apresentadas no painel superior. O utilizador pode aqui observar alguma informação básica das mesmas, assim como visualizar a calendarização desta, através do clique no botão “Schedule” visualizado na Figura 128.

Assim que efetuado o clique, o utilizador navega para uma nova página reservada a Competição seleccionada, apresentado dentro desta um maior detalhe de informação. Aqui é possível observar as datas agendadas e as suas respetivas questões.

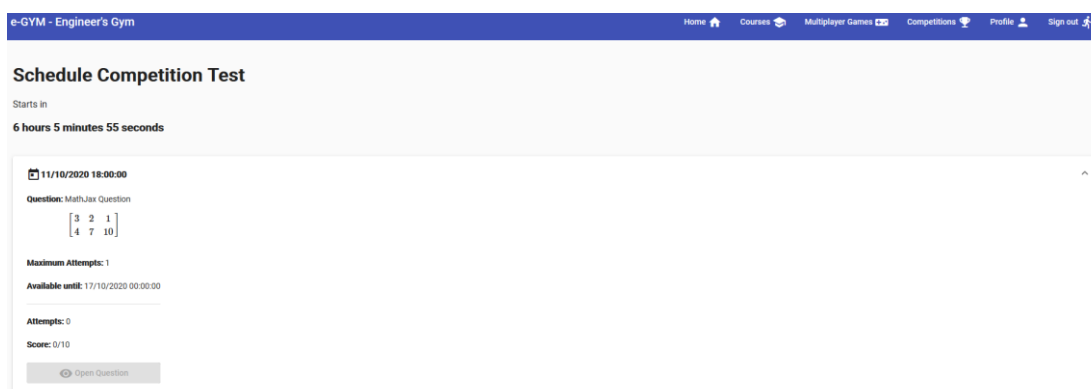


Figura 129 - Calendarização de uma Competição, no *Front Office*

No caso específico da Figura 129 é possível observar uma contagem decrescente para o início da Competição, assim como uma questão agendada para o dia 11/10/2020 às 18:00h. Sobre a questão é ainda possível observar as suas restrições e a performance do utilizador sobre a mesma questão, apresentando o número de tentativas efetuadas e a sua melhor pontuação. Como a Competição ainda não se encontra ativa, o utilizador é impedido de visualizar por completo a dada questão e as suas respetivas respostas.

No caso de a Competição se encontrar em progresso, o botão apresenta uma mensagem e visual diferente, permitindo assim ao seu utilizador proceder à ação específica.

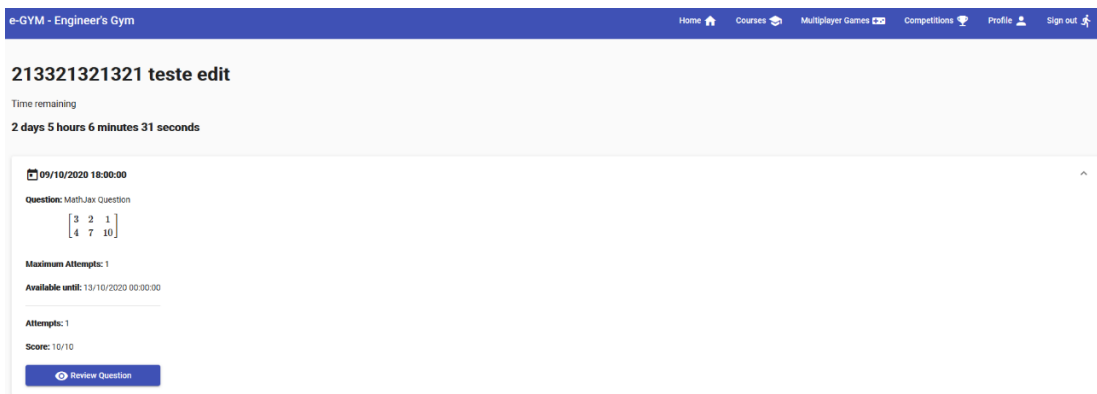


Figura 130 - Competição em progresso, no *Front Office*

Para o exemplo apresentado na Figura 130, o utilizador já esgotou o número de tentativas permitidas e como tal apenas pode rever a sua resposta última tentativa efetuada. Durante essa revisão para além da resposta selecionada, o utilizador pode ver o *feedback* dessa mesma, assim como o tempo utilizado.

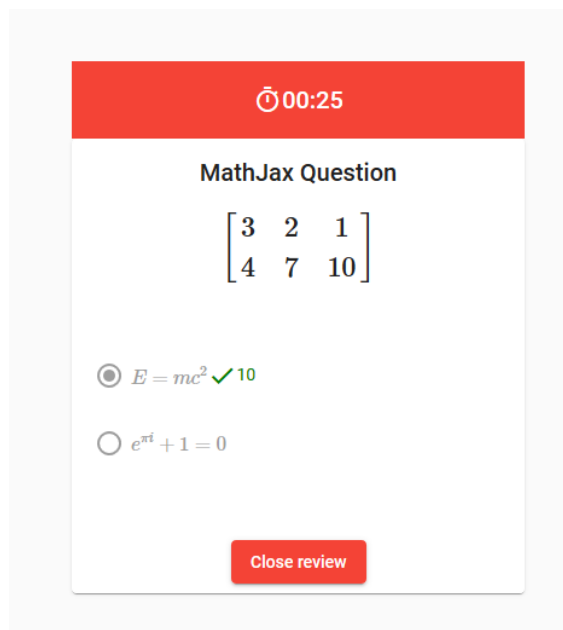


Figura 131 – Revisão de uma questão de uma Competição, no *Front Office*

A página da Competição apresenta ainda uma classificação geral dos utilizadores participantes da mesma. Estando normalmente presente em atividades competitivas e mais concretamente em Competições, este *ranking* classifica os utilizadores utilizando as seguintes condições:

1. Maior número de pontos;
2. Menor número de tentativas;
3. Menor tempo utilizado nas respetivas tentativas;
4. Ordem alfabética;

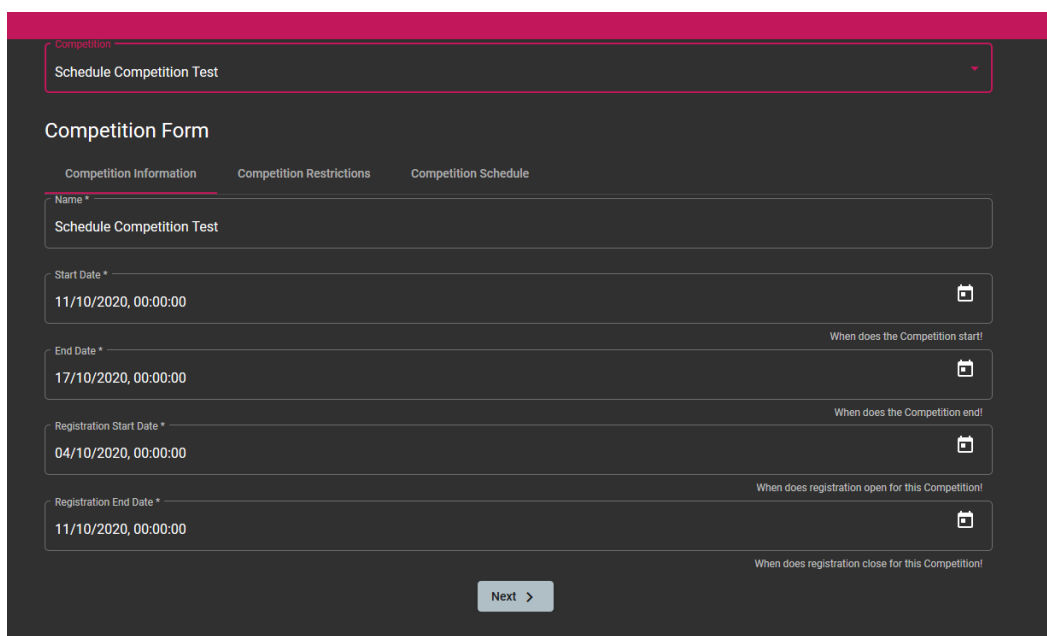
Sempre que se encontre uma igualdade no primeiro critério, é utilizado o seguinte como classificador dos utilizadores, repetindo-se o processo até ao último critério se assim o for necessário.



| User | Questions Answered | Number Of Attempts | Points | Time |
|---------------------|--------------------|--------------------|--------|----------|
| 1. João Costa 🏆 | 1 | 1 | 10 | 00:00:25 |
| 2. João Fernandes 🏆 | 1 | 1 | 10 | 00:00:29 |
| 3. Paulo Pinto 🏆 | 0 | 0 | 0 | 00:00:00 |

Figura 132 - Classificação de uma dada Competição, no *Front Office*

Relativamente a sua edição no *Back Office*, o processo mais uma vez é semelhante aos restantes descritos anteriormente. O supervisor deve selecionar uma Competição que pretende editar e depois de efetuada a sua seleção, toda a informação dessa mesma Competição é carregada para o formulário de edição, podendo depois o utilizador proceder as edições pretendidas.



Competition Form

Competition Information | Competition Restrictions | Competition Schedule

Name *
Schedule Competition Test

Start Date *
11/10/2020, 00:00:00 When does the Competition start!

End Date *
17/10/2020, 00:00:00 When does the Competition end!

Registration Start Date *
04/10/2020, 00:00:00 When does registration open for this Competition!

Registration End Date *
11/10/2020, 00:00:00 When does registration close for this Competition!

Next >

Figura 133 - Exemplo de uma edição de uma Competição, no *Back Office*

7.1.5 Aprofundamento da Estatística existente

Relativamente aos processos de Estatística existentes na plataforma do *Back Office*, estes apresentavam-se bastante simples e limitados, sem a profundidade pretendida para este sistema. Como referido anteriormente (cf. secção 3.3.5) o sistema estava preparado para apenas oferecer aos utilizadores do *Back Office* os seguintes dados estatísticos:

- Número global de utilizadores;
- Número de Áreas;
- Número de Jogos;
- Número de Módulos;
- Número de Questões.

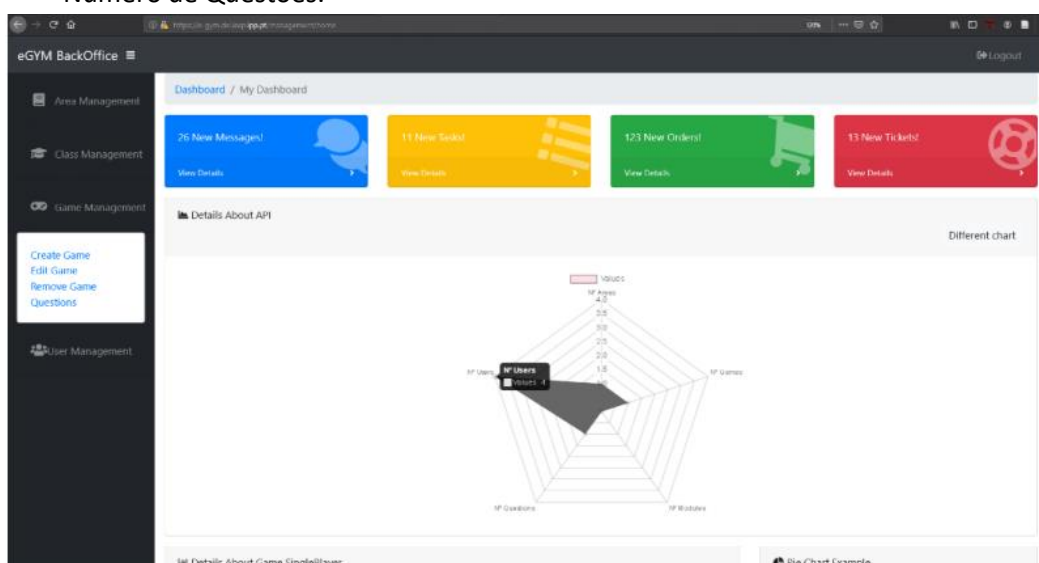


Figura 134 – Estatística disponibilizada na versão anterior do sistema (Luís Peixoto, 2019)

Para a nova iteração do sistema desenvolvido, foi desde cedo acordado na necessidade de expandir este conceito. Foi assim decidido que o novo conceito de Estatística presente no *Back Office* devia permitir obter dados estatísticos sobre todas as entidades relevantes, assim como para cada uma destas aplicar diferentes filtros.

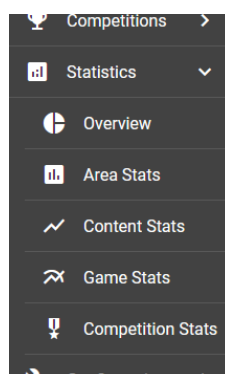


Figura 135 – *Sidebar* de navegação com as opções de Estatística, no *Back Office*

Para o novo sistema, a Estatística oferecida foi agrupada em diferentes grupos:

- **Overview:** dados estatísticos sobre a plataforma em geral;
- **Área:** dados estatísticos relativos às áreas existentes;
- **Conteúdo:** dados estatísticos relativos aos conteúdos existentes;
- **Jogos:** dados estatísticos relativos aos jogos existentes;
- **Competição:** dados estatísticos relativos as competições existentes;

Assim que algumas das opções, representada na Figura 135, fosse selecionada por um utilizador do *Back Office* a plataforma oferecia ao utilizador a escolha de diferentes filtros de pesquisa tendo em conta a opção tomada. Considere-se o exemplo de uma análise estatística da atividade existente no sistema, sendo que esta atividade representa o consumo de conteúdos e atividades no sistema como vídeo-aulas, documentos de suporte ou jogos didáticos:

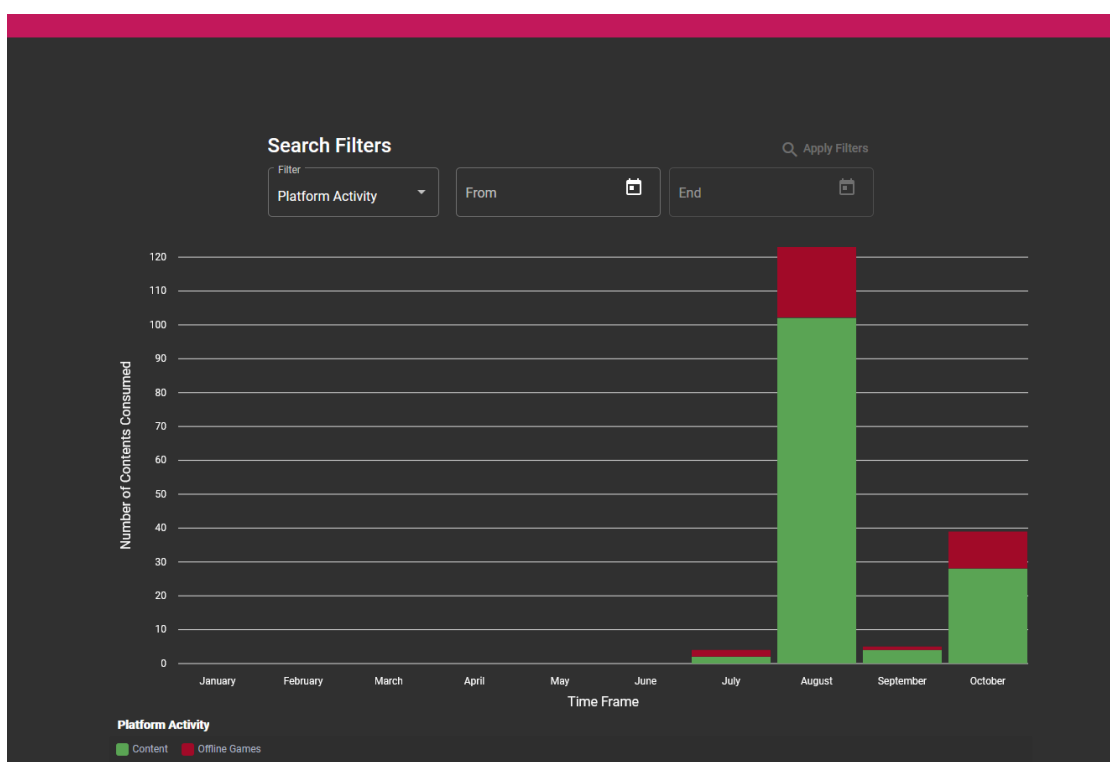


Figura 136 – Estatística relativas a atividade existente no sistema, no *Back Office*

Para além da atividade da plataforma, o utilizador poderia ainda visualizar outros dados interessantes como os intervalos de idade dos utilizadores registados assim como a distribuição dos países dos utilizadores registados:

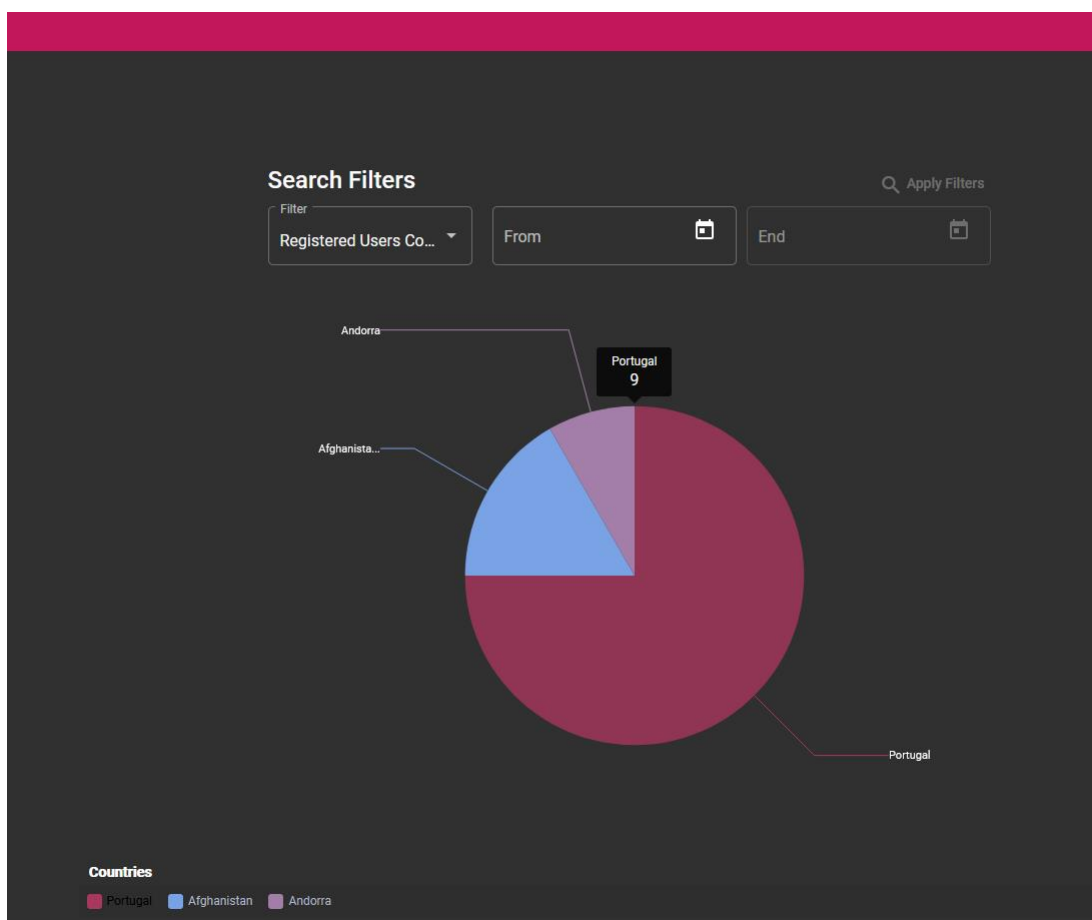


Figura 137 – Visualização estatística da distribuição de países dos utilizadores registados no sistema, no *Back Office*

Para além do filtro de pesquisa utilizado, considerando ainda o exemplo referido anteriormente, o utilizador poderia ainda restringir esta pesquisa introduzindo uma data inicial e uma data final. No caso de o utilizador não inserir nenhuma data inicial e final, são consideradas como data inicial o dia de criação do sistema e como data final o dia atual.

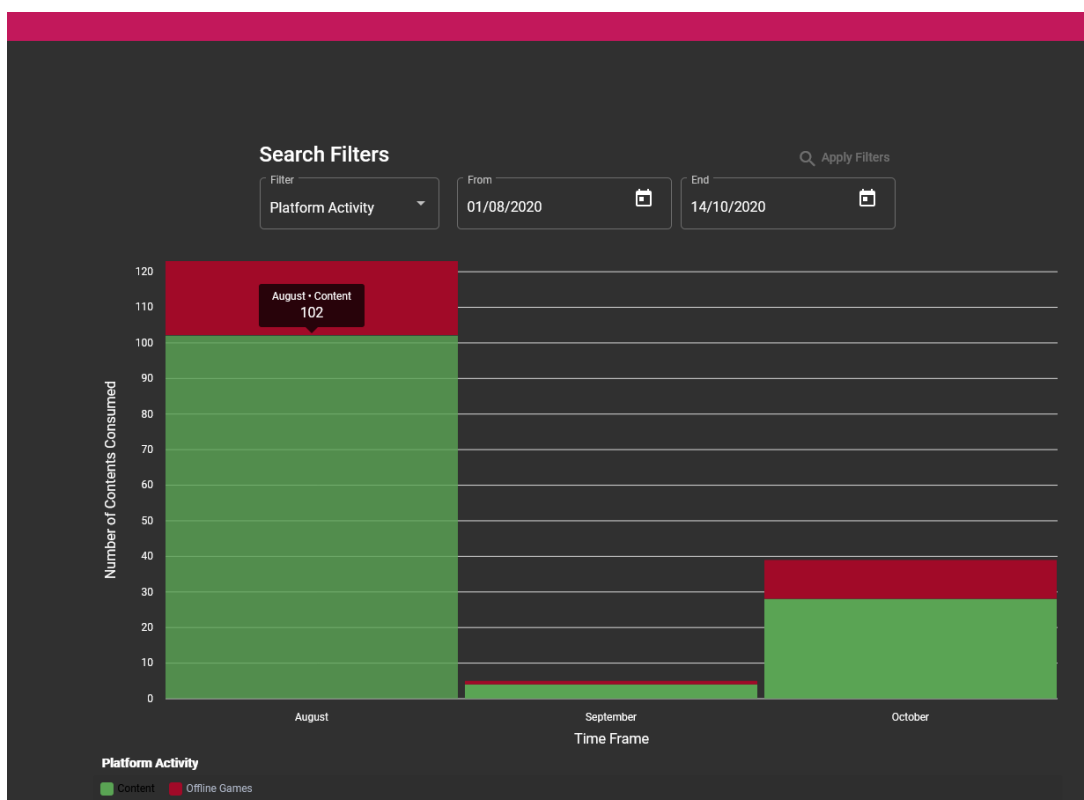


Figura 138 – Pesquisa estatística da atividade existente no sistema entre 01/08/2020 e 14/10/2020, no *Back Office*

No caso de o utilizador pretender consultar algum dado estatístico relacionado com os conteúdos e atividades do sistema, é adicionado um novo filtro de pesquisa que permite ao utilizador selecionar, do tipo de conteúdos selecionado, um número específico destes.

Considere-se que o utilizador do *Back Office* pretende visualizar a popularidade dos conteúdos dentro das áreas existentes. No entanto este pretende visualizar apenas esta mesma popularidade para um grupo restrito de áreas e não para todas. Como tal o sistema permite ao utilizador, para além dos filtros anteriormente apresentados, a possibilidade de selecionar um número de áreas a filtrar, executando assim uma pesquisa mais detalhada:

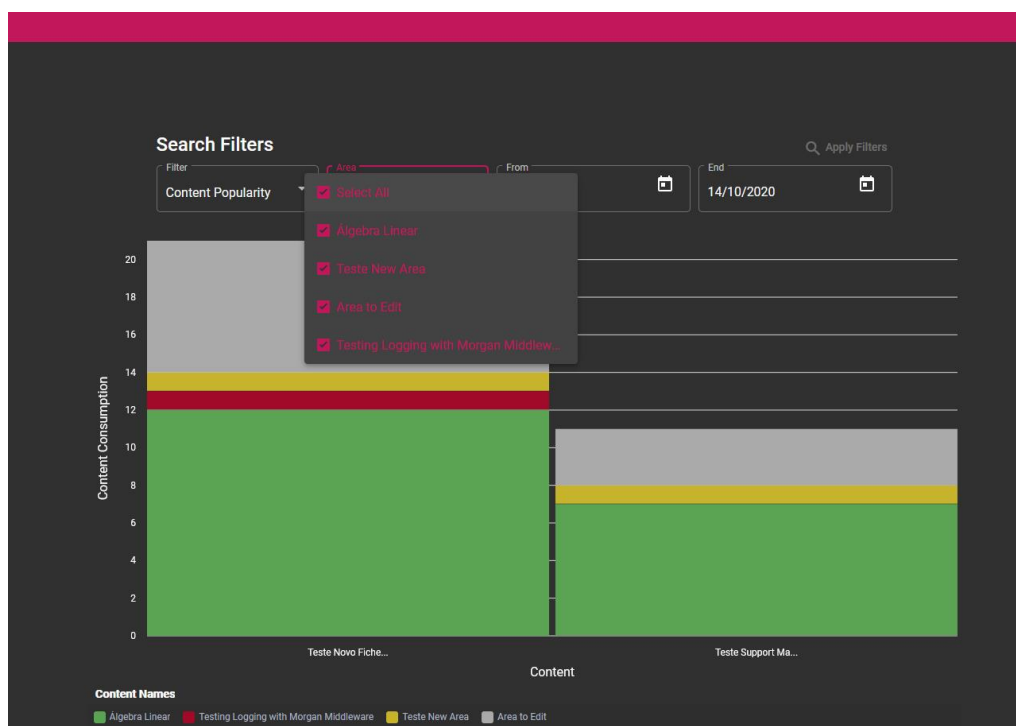


Figura 139 – Consulta estatística da popularidade dos conteúdos existentes nas áreas selecionadas, no *Back Office*

O mesmo é observado na visualização da popularidade dos jogos didáticos, sendo que neste caso de exemplo o utilizador seleciona apenas três das quatro possíveis áreas:

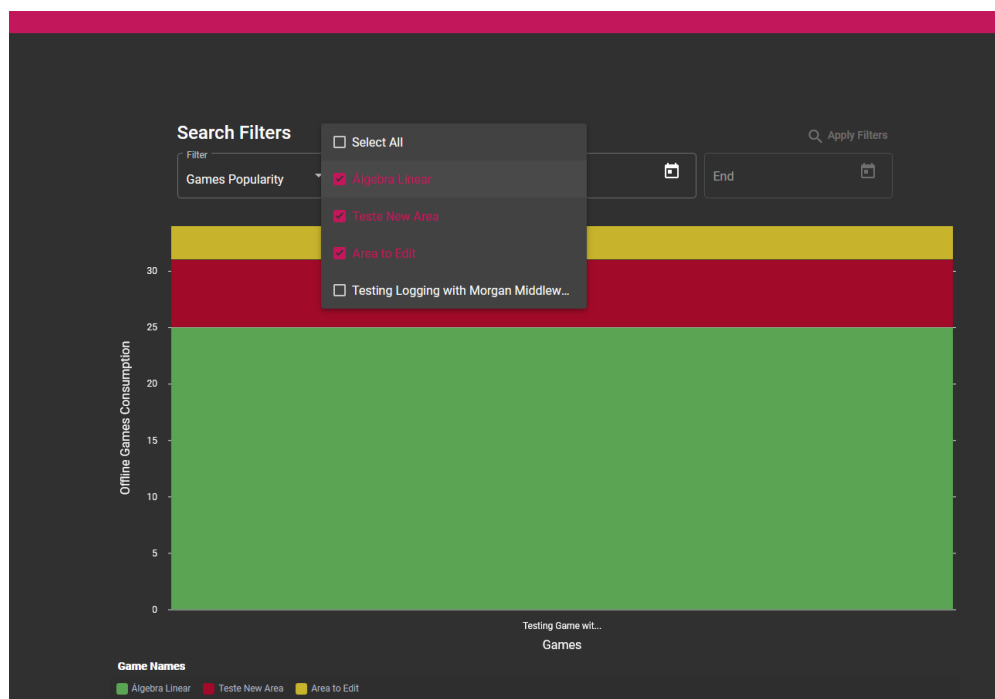


Figura 140 – Consulta estatística da popularidade dos jogos didáticos existentes nas áreas selecionadas, no *Back Office*

Relativamente aos jogos, o utilizador consegue para além de dados de popularidade, observar dados relativamente as taxas de sucesso dos jogos, assim como os tempos médios necessários para terminar o jogo. Estes dois últimos filtros foram desde cedo, considerados como essenciais, pois permitem verificar se a curva de dificuldade de um jogo se encontra adequada para o nível pretendido.

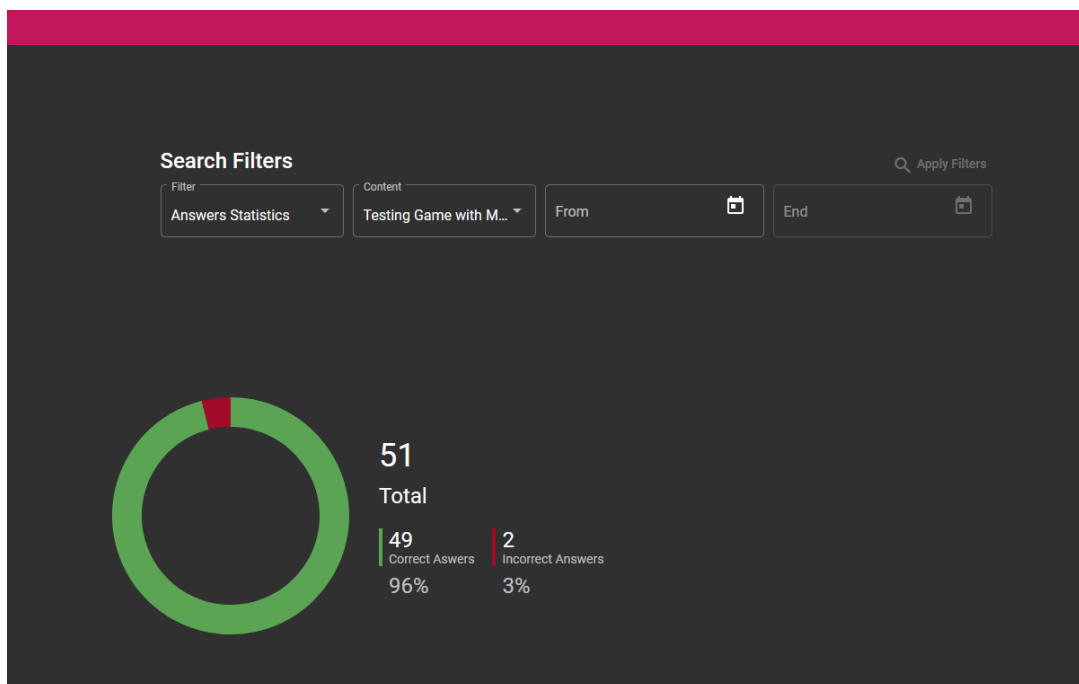


Figura 141 – Consulta estatística das percentagens de respostas corretas e erradas de um jogo, no *Back Office*

7.2 Funcionalidades Relevantes

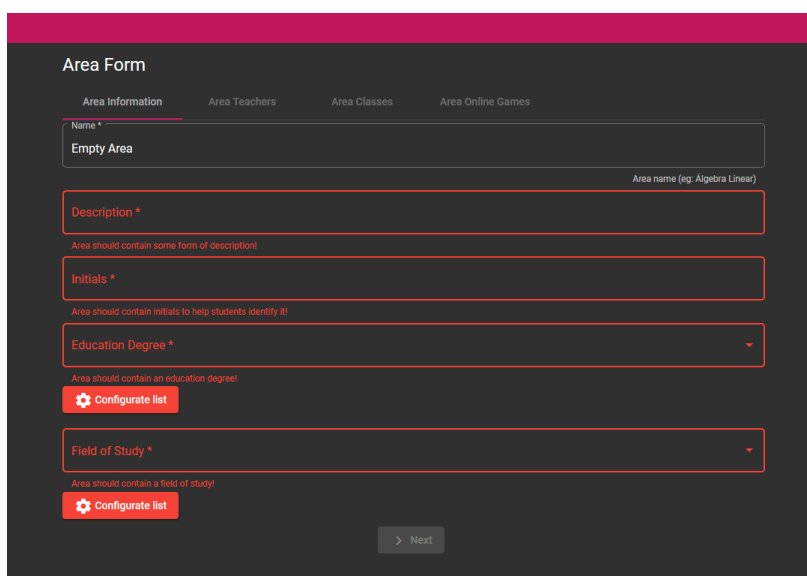
Nesta secção são abordadas implementações consideradas relevantes para o sistema. Por relevante denominam-se implementações que não se encontrem diretamente ligadas com as implementações referidas anteriormente, mas sim como implementações que elevam o valor do sistema como um todo.

7.2.1 Validação dos Formulários

Tratando-se na sua essência de um sistema dedicado ao consumo e a criação de conteúdo e atividades lúdicas, a utilização de formulários de criação tornou-se dominante, podendo-se mesmo encontrar um formulário para quase todas as ações existentes, predominantemente no *Back Office*. Atualmente a maioria das aplicações utilizam este tipo de formulários para permitir aos seus utilizadores que realizem funções como logins, atualizações de perfis, entre outras.

No caso do sistema abordado nesta dissertação, existiu desde cedo uma grande preocupação na validação destes formulários. Sendo o sistema constituído por diversas entidades, que necessitam de comunicar e de se relacionar entre si, é essencial que todas as ações efetuadas por utilizadores alheios a infraestrutura existente no sistema seja realizada de forma correta.

Para além de todas as mensagens de ajuda existentes sempre que possível e necessário, perto de todos os locais de *input* de dados, procedeu-se a validação destes mesmos quando inseridos. Com estas validações procurou-se eliminar ao máximo o número de erros possíveis durante as ações efetuadas, relacionados com a falta de dados necessários e/ou com a introdução de dados em formatos errados.



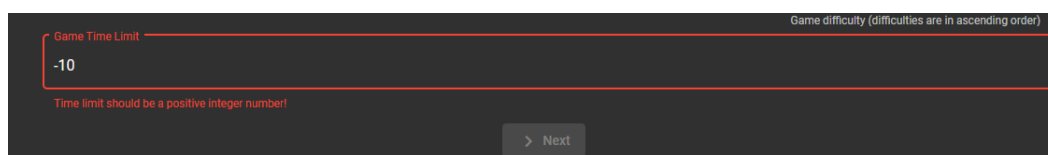
The screenshot shows a form titled "Area Form" with four tabs: "Area Information", "Area Teachers", "Area Classes", and "Area Online Games". The "Area Information" tab is active. The form contains several fields with validation errors:

- Name ***: The input field contains "Empty Area". A red border highlights the field, and a message below it says "Area name (eg. Algebra Linear)".
- Description ***: The input field is empty. A red border highlights the field, and a message below it says "Area should contain some form of description".
- Initials ***: The input field is empty. A red border highlights the field, and a message below it says "Area should contain initials to help students identify it".
- Education Degree ***: A dropdown menu is open, showing a list of options. A red border highlights the field, and a message below it says "Area should contain an education degree!". Below the field is a "Configure list" button.
- Field of Study ***: A dropdown menu is open, showing a list of options. A red border highlights the field, and a message below it says "Area should contain a field of study!". Below the field is a "Configure list" button.

At the bottom of the form, there is a "Next" button.

Figura 142 – Validações existentes em formulários, no *Back Office*

Para além da simples validação de dados obrigatórios, sempre identificados pelo carácter "*", existiu uma especial atenção para a validação de dados numéricos. Para estes dados são sempre validados tendo em conta o objetivo da ação, como por exemplo na introdução de o número de segundos para o tempo limite de um jogo. Neste caso é sempre esperado, que caso o utilizador insira um número este seja sempre positivo.



The screenshot shows a form titled "Game Time Limit" with a dropdown menu for "Game difficulty (difficulties are in ascending order)". The "Game Time Limit" field contains the value "-10". A red border highlights the field, and a message below it says "Time limit should be a positive integer number". At the bottom of the form, there is a "Next" button.

Figura 143 – Validação da introdução do tempo limite de um Jogo, no *Back Office*

Para além de dados números, dados como emails são também validados.

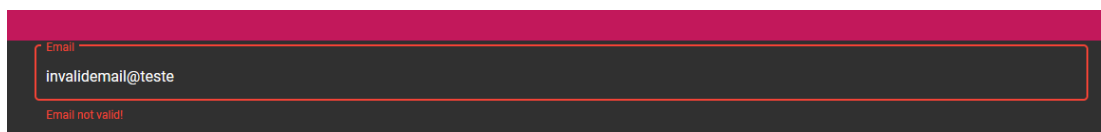


Figura 144 - Validação de um email, no *Back Office*

Dados sensíveis como datas sofrem também as mesmas validações.

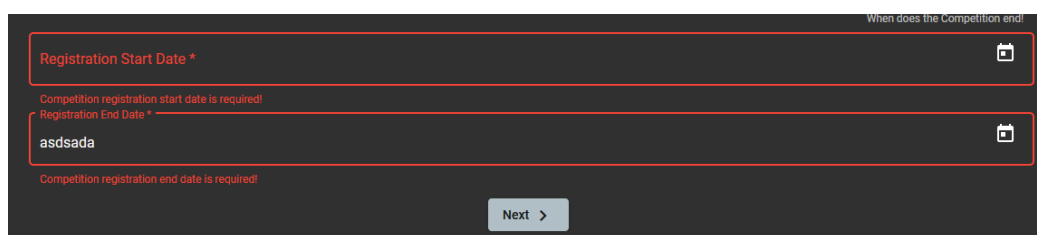


Figura 145 - Validação de datas de uma Competição, no *Back Office*

Para além de todos estes tipos de validações implementados ao longo dos diferentes formulários, o utilizador é bloqueado de efetuar qualquer ação final que necessite de comunicar com o *Back End* do sistema, enquanto que o formulário não se encontrar totalmente válido.

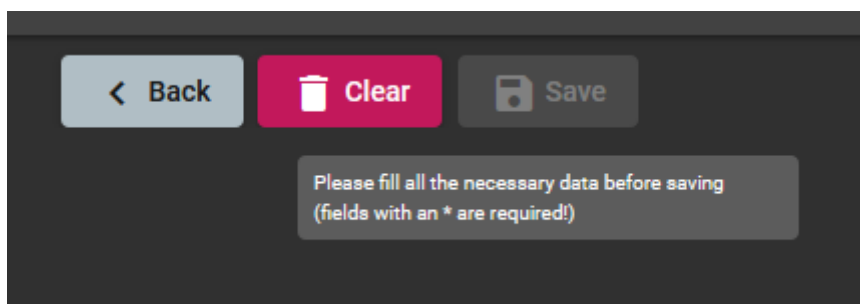


Figura 146 – Informação do estado inválido de um formulário, no *Back Office*

7.2.2 Localização

Sendo esperado que o eGYM atinja utilizadores de diferentes locais do mundo, tendo cada um deles a sua língua oficial, é necessário que o sistema se encontre preparado para oferecer conteúdos e meios de navegação em diferentes línguas.

Para tal é incluída na barra de navegação da plataforma do *Front Office* uma opção para a seleção da língua utilizada para o texto apresentado na plataforma. Esta opção encontra-se visualmente sinalizada através da bandeira do país da língua utilizada, sendo esta por defeito o Inglês.

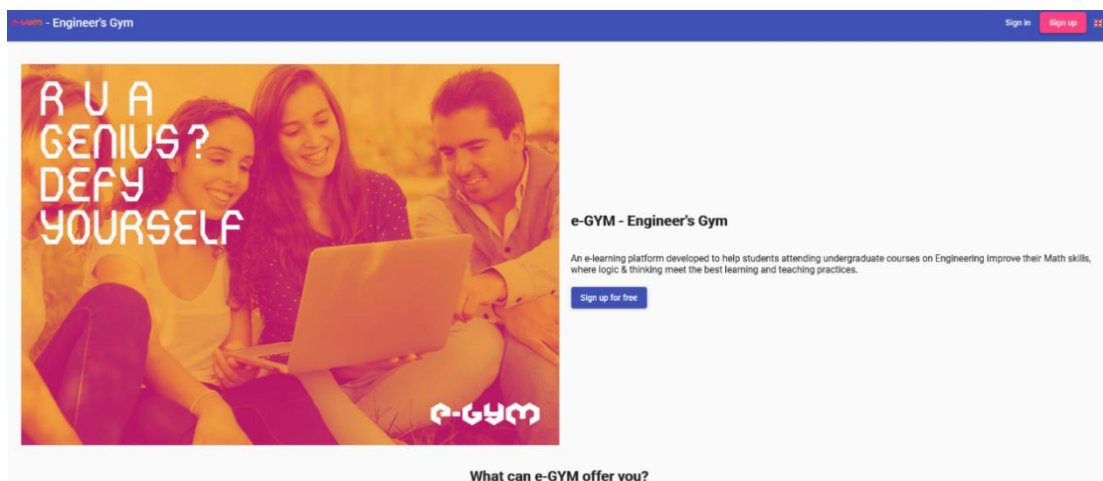


Figura 147 – Visualização da opção para a seleção de uma língua, no *Front Office*

O utilizador pode em qualquer momento proceder a troca da língua atual por uma outra língua disponível. Para esta reengenharia do sistema eGYM se encontram disponíveis duas línguas: o Português e o Inglês.

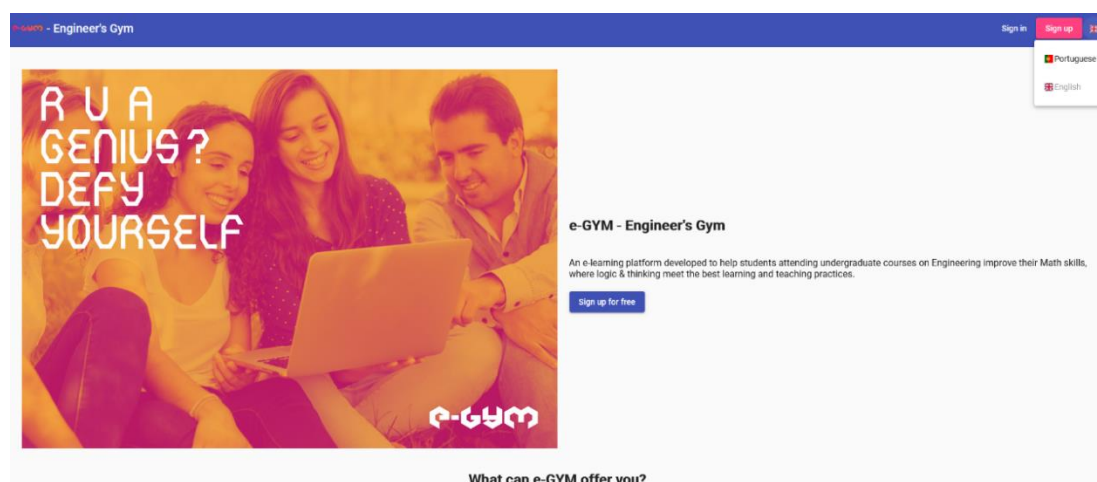


Figura 148 – Visualização das diferentes línguas oferecidas, no *Front Office*

Quando procedida à alteração da língua a utilizar pela plataforma por um dado utilizador, a página é recarregada, apresentando depois todo o texto possível de tradução na nova língua selecionada.

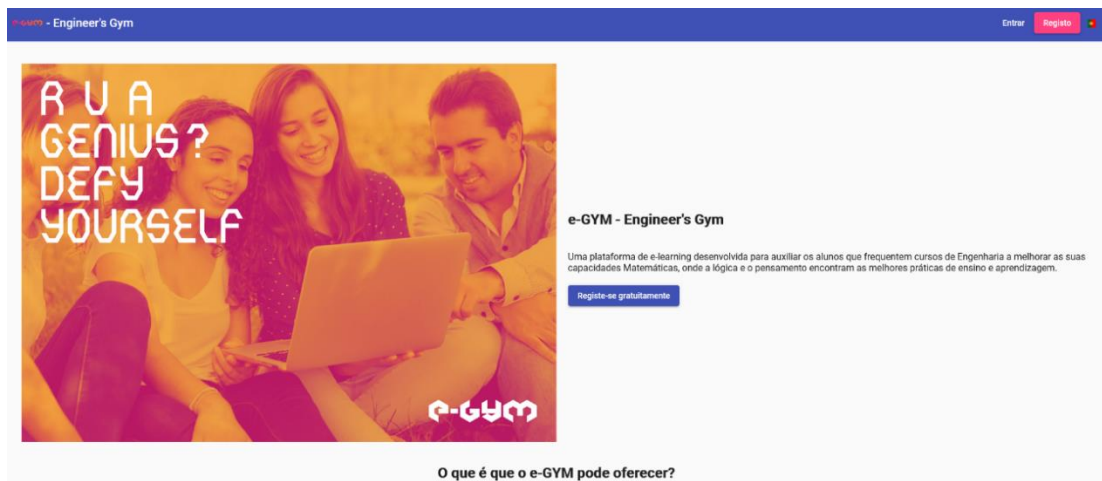


Figura 149 – Visualização da atualização do texto existente na plataforma após a seleção da língua Portuguesa, no *Front Office*

Para além dos textos as opções na barra de navegação também são traduzidas adequadamente.



Figura 150 - Barra de navegação traduzida para Português, no *Front Office*

8 Experiências e Avaliação

O presente capítulo aborda as experiências e avaliações utilizadas para avaliar a solução desenvolvida, assim como os resultados obtidos. O capítulo estará dividido em duas secções distintas.

Na primeira secção será apresentado um enquadramento teórico da avaliação utilizada analisando as grandezas ponderadas no trabalho, as hipóteses a testar e as metodologias de avaliação adotadas.

Na segunda secção serão apresentados os resultados obtidos dos inquéritos realizados, sendo depois feita uma análise estatísticas desses mesmos resultados através de testes estatísticos.

Para terminar numa terceira e última secção serão retiradas algumas conclusões tendo em conta os resultados e a análise efetuada.

8.1 Enquadramento teórico

Nesta primeira secção serão detalhados conceitos de enquadramento teóricos relativos à avaliação efetuada sobre o produto desenvolvido.

8.1.1 Grandezas

Em qualquer avaliação efetuada é necessário à priori delinear as grandezas que serão utilizadas para avaliar o trabalho efetuado.

No contexto do problema desta dissertação, o sistema desenvolvido interage com diversos utilizadores e oferece diversas funcionalidades, sendo como tal imprescindível ter em consideração a sua utilidade, usabilidade e desempenho.

8.1.1.1 Utilidade

Relativamente a utilidade esta grandeza é utilizada para garantir que o novo sistema construído e a suas funcionalidades acrescentam valor e vão de encontro as necessidades dos utilizadores.

8.1.1.2 Usabilidade

Usabilidade representa a facilidade de utilização do sistema desenvolvido. Quando se constrói um sistema desta dimensão é sempre necessário que os seus utilizadores se sintam confortáveis no seu manuseamento, através de uma interface simples e intuitiva.

8.1.1.3 Desempenho

Sendo um sistema baseado em disponibilização de conteúdos através de uma plataforma *web*, um bom desempenho torna-se fundamental para permitir uma boa experiência aos seus utilizadores. Os tempos de carregamento dos conteúdos e das atividades devem assim manter um tempo aceitável ao longo da utilização do sistema.

8.1.2 Hipóteses

Numa avaliação as hipóteses representam suposições que se pretendem comprovar tendo em conta os objetivos traçados para o projeto. Os testes de hipóteses são assim utilizados para avaliar a plausibilidade de uma hipótese quando confrontada com os dados de amostra existentes.

No âmbito da presente dissertação as hipóteses definidas vão de encontro aos objetivos delineados na secção 1.3, considerando as grandezas de avaliação enunciadas na subsecção anterior.

São assim construídas as seguintes hipóteses:

1. As novas funcionalidades introduzidas vão de encontro as necessidades existentes, acrescentando assim valor ao produto;
2. As alterações introduzidas no sistema não prejudicam a sua usabilidade, oferecendo aos seus utilizadores uma boa experiência;
3. A integração de novos conceitos não interfere no desempenho do sistema.

8.1.3 Metodologia de avaliação

Tendo em consideração as grandezas referidas e as hipóteses a provar são utilizadas metodologias de avaliação que permitam avaliar as grandezas e as hipóteses enunciadas.

Como tal para a avaliar a satisfação dos utilizadores são utilizados inquéritos de satisfação que são aplicados a um grupo de teste específico. Tendo em conta que o projeto desenvolvido ainda se encontra numa fase piloto, onde o seu conhecimento do público geral é bastante limitado ou mesmo inexistente, o grupo de controlo será apenas constituído por supervisores do sistema.

Quanto ao inquérito, este será dividido em três secções diferentes para permitir uma melhor análise das grandezas e hipóteses delineadas.

Para a primeira secção, que corresponde a grandeza da utilidade, as questões desenvolvidas procuram verificar se o sistema corresponde ao esperado, questionando o inquirido se cada uma das funcionalidades satisfaz os seus objetivos, apresentado as seguintes questões:

1. Considera-se os novos papéis de utilizador introduzidos, assim como a sua configurabilidade, uma melhoria em relação a sistemas anteriores?
2. Os novos processos de autorização na plataforma oferecem controlo sobre os utilizadores e os conteúdos existentes na plataforma?
3. O novo sistema apresenta melhorias em relação a configurabilidade de novos conteúdos e atividades?
4. O novo sistema permite de uma forma eficaz a reutilização de diversos conteúdos e atividades?
5. A implementação dos novos Jogos multijogador vai de encontro ao desejado e esperado?
6. A introdução das Competições no sistema e a sua estrutura elimina as limitações encontradas em sistemas anteriores?
7. O aprofundamento do conceito Estatístico do sistema, assim como os seus filtros de pesquisa, satisfaz os requisitos definidos?
8. A nova iteração do sistema desenvolvida acrescenta valor ao produto?

Na segunda secção, que representa a usabilidade, é baseada no método SUS (*System Usability Scale*) desenvolvido por John Brooke (Brooke, 1996) sendo este composto por 10 afirmações que permitem avaliar a qualidade de um produto relativamente a sua usabilidade:

1. Gostaria de utilizar este sistema frequentemente;
2. Acho o sistema desnecessariamente complexo;
3. Considero o sistema fácil de utilizar;
4. Sinto que necessito de ajuda para utilizar o sistema;

5. Considero que as funções do sistema estão bem integradas;
6. Penso que havia muitas inconsistências no sistema;
7. Imagino que a maioria das pessoas se adaptaria rapidamente ao sistema;
8. Acho o sistema muito complicado de se utilizar;
9. Sinto confiança durante a utilização do sistema;
10. Preciso de aprender imensas coisas antes de começar a utilizar propriamente o sistema;

Na terceira e última secção, que se relaciona com o desempenho, envolve questões que permitam comprovar que o sistema é relativamente rápido a carregar os recursos, não prejudicando assim a experiência do utilizador:

1. O carregamento das interfaces é efetuado em tempo aceitável?
2. O carregamento dos conteúdos e atividades da plataforma são efetuados em tempo aceitável?
3. A introdução ou atualização de conteúdos e atividades no sistema é efetuado em tempo aceitável?
4. O sistema na sua generalidade apresenta uma *performance* aceitável?

Quanto as respostas, é utilizada a escala de Likert onde os inquiridos especificam o seu nível de concordância ou discordância perante uma afirmação. A escala representa o somatório de diferentes itens, sendo que o formato de cada item é normalmente:

1. Discordo totalmente;
2. Discordo;
3. Indiferente;
4. Concordo;
5. Concordo totalmente.

8.2 Resultados

Na presente secção é feito um levantamento de todas as respostas obtidas ao inquérito anteriormente delineado. Para a presente dissertação, foram colocadas vinte e duas questões a três utilizadores, sendo estes os clientes do sistema.

Apresenta-se então de seguida todos os resultados obtidos, sendo que cada resultado se faz acompanhar de um pequeno gráfico que facilite a compreensão dos mesmos.

8.2.1 Utilidade

Considera-se os novos papéis de utilizador introduzidos, assim como a sua configurabilidade, uma melhoria em relação a sistemas anteriores?

3 respostas



Figura 151 – 1ª pergunta de utilidade do inquérito

Os novos processos de autorização na plataforma oferecem controlo sobre os utilizadores e os conteúdos existentes na plataforma?

3 respostas

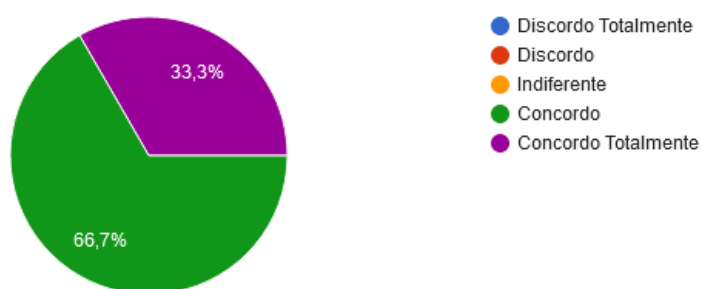


Figura 152 - 2ª pergunta de utilidade do inquérito

O novo sistema apresenta melhorias em relação a configurabilidade de novos conteúdos e atividades?

3 respostas



Figura 153 – 3ª pergunta de utilidade do inquérito

O novo sistema permite de uma forma eficaz a reutilização de diversos conteúdos e atividades?

3 respostas

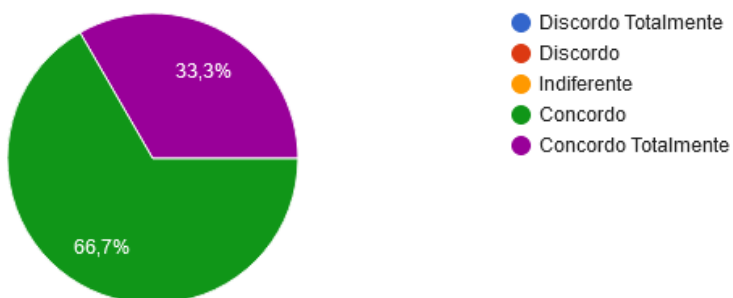


Figura 154 - 4ª pergunta de utilidade do inquérito

A implementação dos novos Jogos multijogador vai de encontro ao desejado e esperado?

3 respostas

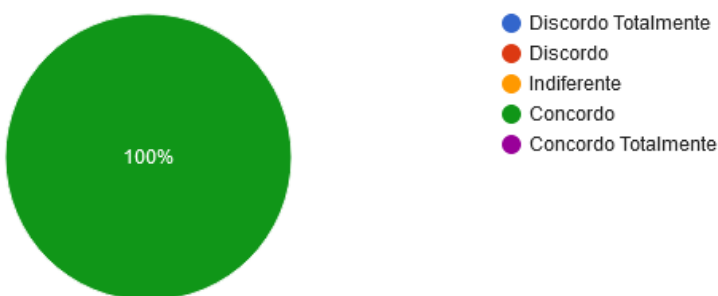


Figura 155 – 5ª pergunta de utilidade do inquérito

A introdução das Competições no sistema e a sua estrutura elimina as limitações encontradas em sistemas anteriores?

3 respostas



Figura 156 - 6ª pergunta de utilidade do inquérito

O aprofundamento do conceito Estatístico do sistema, assim como os seus filtros de pesquisa, satisfaz os requisitos definidos?

3 respostas

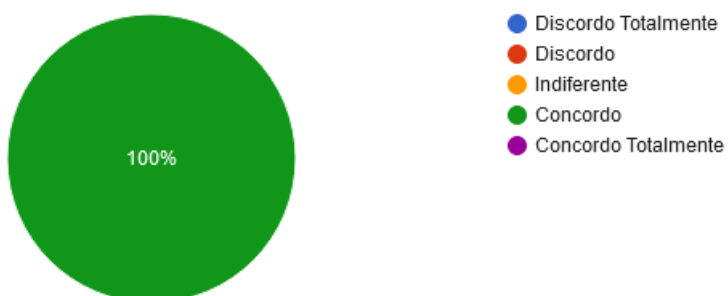


Figura 157 – 7ª pergunta de utilidade do inquérito

A nova iteração do sistema desenvolvida acrescenta valor ao produto?

3 respostas



Figura 158 – 8ª pergunta de utilidade do inquérito

8.2.2 Usabilidade

Gostaria de utilizar este sistema frequentemente?

3 respostas

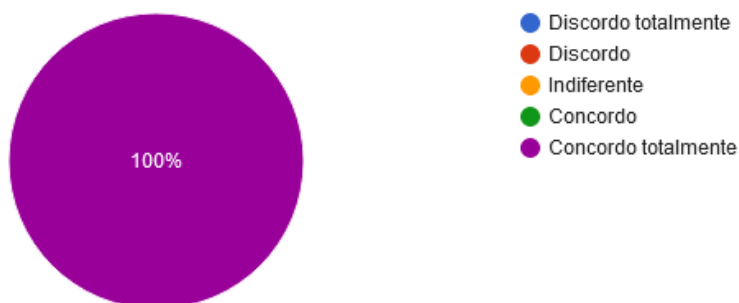


Figura 159 - 1ª pergunta de usabilidade do inquérito

Acho o sistema desnecessariamente complexo

3 respostas

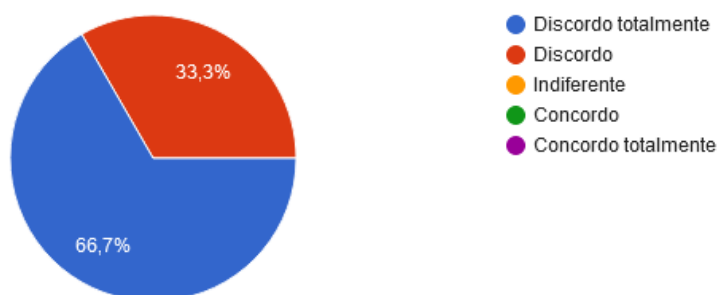


Figura 160 – 2ª pergunta de usabilidade do inquérito

Considero o sistema fácil de utilizar

3 respostas

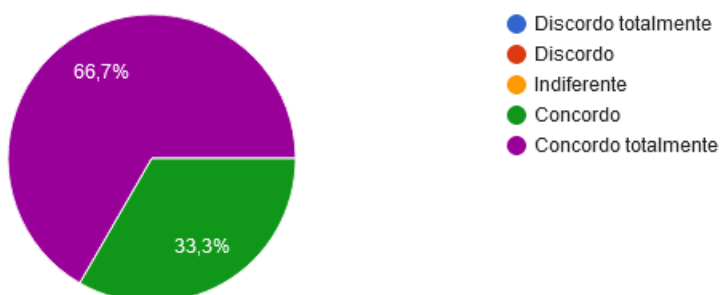


Figura 161 – 3ª pergunta de usabilidade do inquérito

Sinto que necessito de ajuda para utilizar o sistema

3 respostas

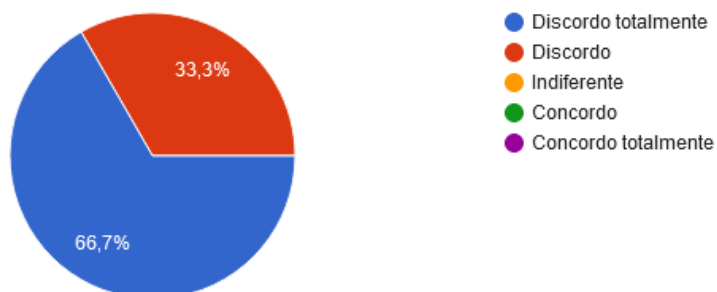


Figura 162 – 4ª pergunta de usabilidade do inquérito

Considero que as funções do sistema estão bem integradas

3 respostas



Figura 163 – 5ª pergunta de usabilidade do inquérito

Penso que havia muitas inconsistências no sistema

3 respostas

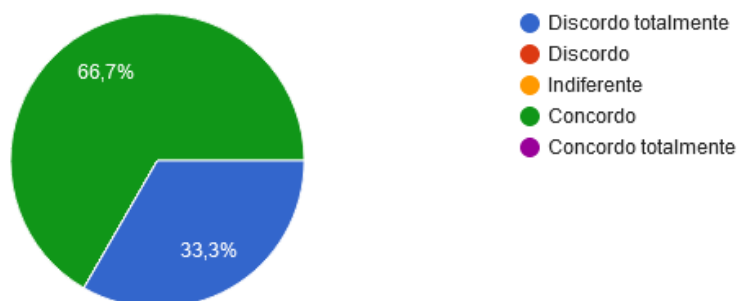


Figura 164 – 6ª pergunta de usabilidade do inquérito

Imagino que a maioria das pessoas se adaptaria rapidamente ao sistema

3 respostas



Figura 165 – 7ª pergunta de usabilidade do inquérito

Acho o sistema muito complicado de se utilizar

3 respostas

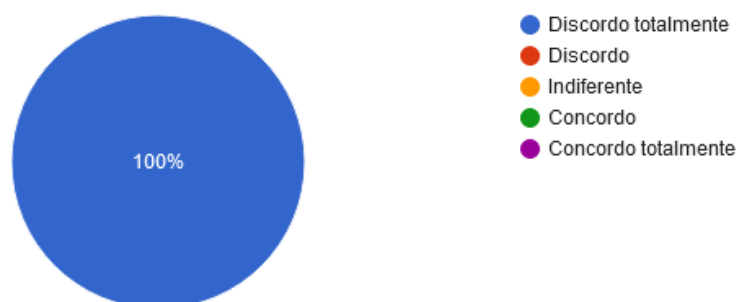


Figura 166 – 8ª pergunta de usabilidade do inquérito

Sinto confiança durante a utilização do sistema

3 respostas

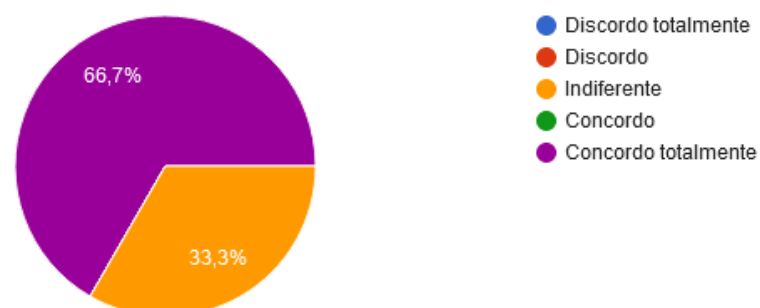


Figura 167 – 9ª pergunta de usabilidade do inquérito

Preciso de aprender imensas coisas antes de começar a utilizar propriamente o sistema

3 respostas

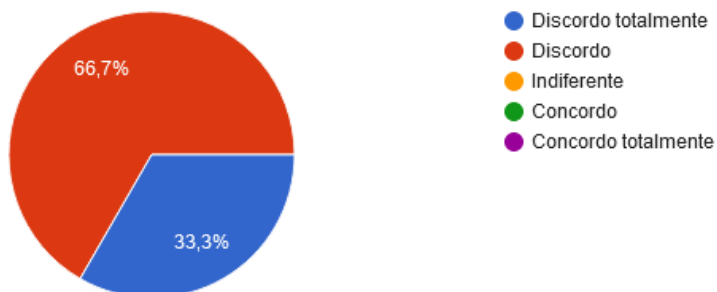


Figura 168 - 10ª pergunta de usabilidade do inquérito

8.2.3 Desempenho

O carregamento das interfaces é efetuado em tempo aceitável?

3 respostas

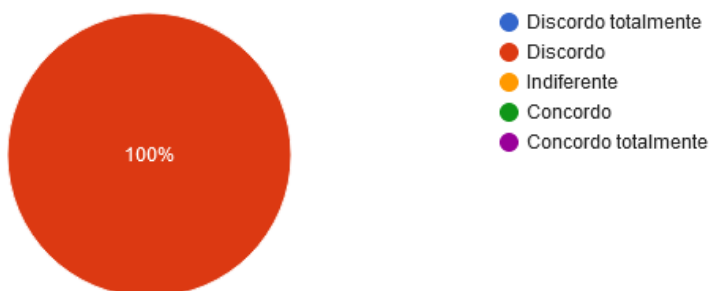


Figura 169 – 1ª pergunta de desempenho do inquérito

O carregamento dos conteúdos e atividades da plataforma são efetuados em tempo aceitável?

3 respostas

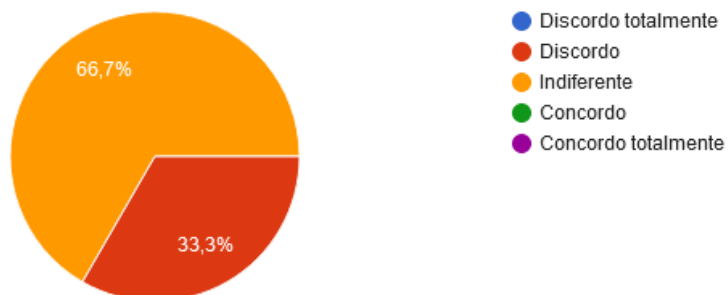


Figura 170 – 2ª pergunta de desempenho do inquérito

A introdução ou atualização de conteúdos e atividades no sistema é efetuado em tempo aceitável?

3 respostas

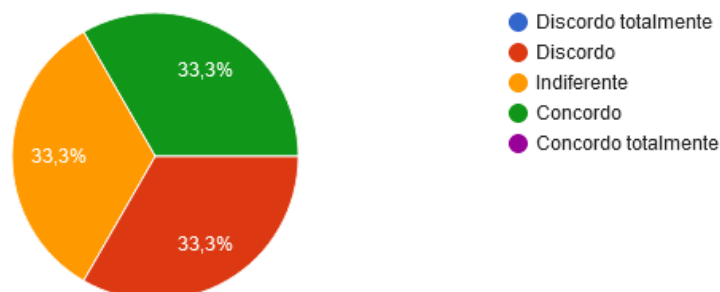


Figura 171 - 3ª pergunta de desempenho do inquérito

O sistema na sua generalidade apresenta uma performance aceitável?

3 respostas

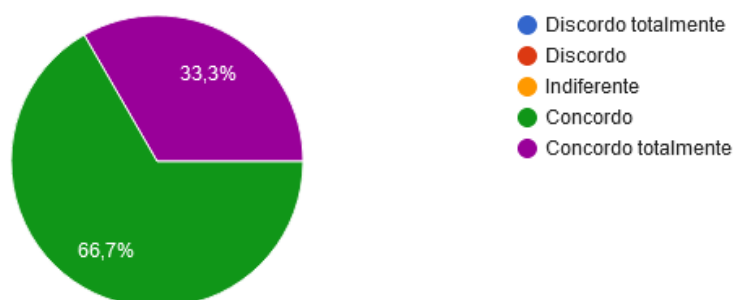


Figura 172 - 4ª pergunta de desempenho do inquérito

8.3 Análise

Antes de se proceder com a análise efetuada aos resultados obtidos é importante novamente realçar o tamanho da amostra utilizada. Como referido anterior (cf. secção 8.1.3) este projeto apesar de se encontrar na terceira iteração, para a presente dissertação, encontra-se ainda numa fase piloto, estando assim indisponível para o público geral. Como tal os únicos utilizadores com acesso ao sistema são os seus próprios proponentes, fazendo assim com que o tamanho de amostra seja de três pessoas.

De seguida para cada um dos diferentes critérios considerados na construção do inquérito de satisfação, são analisadas alguns dos resultados obtidos para certas questões que se considerem pertinentes.

8.3.1 Utilidade

Em relação as questões de utilidade do sistema, esta secção procurava perceber até que ponto os objetivos delineados para o sistema desenvolvido foram atingidos. Através de uma análise dos resultados obtidos é possível encontrar um cenário bastante positivo.

Para a nova configurabilidade introduzida no sistema, 100% dos inquiridos concordam totalmente com a sua melhoria em relação ao sistema anterior (Figura 151).

Quando questionados sobre o atingimento dos objetivos para a implementação dos novos jogos multijogador (Figura 155), 100% dos inquiridos concordam com esta afirmação, apresentando-se assim bastante satisfeitos com a mesma.

Para mais um novo conceito do sistema, como as competições, os inquiridos mostram-se mais uma vez satisfeitos com os progressos efetuados, sendo que 100% dos mesmos concordam na totalidade (Figura 156).

Por fim, 100% dos inquiridos considera que esta nova iteração desenvolvida acrescentou valor ao produto eGYM (Figura 158).

8.3.2 Usabilidade

Quanto a usabilidade do sistema esta secção do inquérito tinha como objetivo aferir de uma maneira geral a facilidade de utilização do novo sistema. É possível observar mais uma vez nos seus resultados um grau de satisfação elevado.

Quanto a sua opinião sobre a adaptação de novos utilizadores a plataforma, mais uma vez 100% dos inquiridos sentem que esta mesma seria de fácil adaptação (Figura 165).

Por fim os inquiridos demonstram ainda que consideram o sistema fácil de se usar, considerando que 100% dos mesmos discordaram da elevada dificuldade de uso do sistema (Figura 166).

8.3.3 Desempenho

Para o desempenho do sistema, a secção desenvolvida pretendia determinar a *performance* do sistema quando sob carga elevada. Ao contrário dos critérios anteriores, o cenário já não se apresenta tão satisfatório, existindo mesmo alguma variação nas respostas obtidas.

Para a classificação dos carregamentos das interfaces (Figura 169) 100% dos inquiridos releva que não considera que as interfaces carreguem em tempos considerados como aceitáveis, apresentando assim alguma insatisfação em relação a este ponto.

Quanto ao desempenho apresentado pelo sistema para operações relacionadas com os conteúdos e atividades do sistema (Figura 171), o grau de satisfação dos inquiridos melhora

ligeiramente, sendo que 33.3% dos inquiridos considera os tempos aceitáveis, 33.3% apresenta-se como indiferente e os restantes 33.3% considera que estes tempos não são aceitáveis.

No entanto quando questionados acerca do desempenho geral das plataformas, 66.7% dos inquiridos considera que o desempenho do sistema é aceitável, enquanto que 33.3% concorda na sua totalidade (Figura 172).

8.3.4 Conclusão

De uma maneira geral a satisfação revelada pelos inquiridos através do inquérito de satisfação apresenta-se positiva, especialmente para os critérios de utilidade e usabilidade. No entanto os seus graus de satisfação variam um pouco em relação ao desempenho do sistema, ficando um pouco abaixo do esperado.

Para dois primeiros, pode-se assim considerar que a reengenharia introduzida no sistema foi um sucesso. Os inquiridos consideram na sua generalidade através das suas respostas que os objetivos foram atingidos e as limitações delineadas eliminadas.

Quanto ao desempenho, o consenso é que o sistema ainda não apresenta a *performance* pretendida, existindo assim alguma margem de manobra. Num sistema desta dimensão é normal que as suas interfaces e conteúdos demorem algum tempo a serem apresentados devido ao grande consumo de recursos que cada plataforma efetua sobre os seus servidores. É por isso possível que estes resultados obtidos não sejam assim tão insatisfatórios, se for considerado que a expectativa dos clientes em relação ao desempenho destes sistemas se encontra desalinhada com a realidade. No entanto, nunca se deve menosprezar o *feedback* obtido sendo por isso necessário efetuar uma nova reflexão e análise do desempenho do sistema.

9 Sumário e trabalho futuro

Neste último capítulo são apresentadas as conclusões finais da presente dissertação.

Na primeira secção são enumerados todos os objetivos e limitações que se considerem como alcançados.

Na segunda secção são enumerados por outro lado os objetivos e limitações que não tenham sido alcançados.

Na terceira secção são abordadas as limitações do sistema desenvolvido, assim como possíveis melhorias para uma futura iteração de desenvolvimento.

Por fim e terminado a presente dissertação é reservada uma secção para uma apreciação final e pessoal do projeto desenvolvido.

9.1 Objetivos alcançados

O maior objetivo da presente dissertação consistiu na reengenharia do sistema eGYM anteriormente desenvolvido, através da introdução de novas funcionalidades que conseguissem combater e eliminar problemas existentes nas versões anteriores do sistema.

Assim e tendo em conta os objetivos enunciados na secção 1.3, é possível determinar que a grande maioria dos objetivos foram alcançados.

Para além destes objetivos delineados à *priori* da reengenharia do sistema, foram ainda alcançados outros objetivos paralelos, como a introdução de localização e internacionalização no sistema, que permitiram acrescentar valor ao sistema.

Em suma, considera-se que a reengenharia do eGYM foi bem-sucedida, mesmo sem que todos os objetivos tenham sido alcançados.

9.2 Objetivos não alcançados

Apesar dos objetivos considerados de maior importância e prioritários para esta reengenharia terem sido alcançados com sucesso, o suporte a plugins e seu respetivo controlo no sistema não foram desenvolvidos. A grande extensão do sistema e a sua baixa hierarquia de prioridade em relação aos restantes, acabaram por contribuir para tal.

No entanto e apesar da sua menor prioridade para o sistema, este nunca foi totalmente descurado. Foi efetuada uma análise extensiva de possíveis soluções existentes e de diversos estudos já efetuados, sendo possível verificar que a *framework* Angular se apresenta capaz de suportar a implementação de plugins, sendo assim por isso algo a considerar para uma futura iteração de desenvolvimento do sistema.

9.3 Limitações e trabalho futuro

Apesar de se considerar a reengenharia efetuada sobre o sistema eGYM como bem-sucedida, foi possível concluir, após uma reflexão efetuada sobre todo o trabalho efetuado, que existe ainda espaço para futuras melhorias no sistema.

É esperado que sistemas deste tipo nunca se encontrem concluídos, sendo possível acrescentar valor aos mesmos através da introdução de novas funcionalidades ou através da melhoria de funcionalidades já implementadas.

Uma primeira e óbvia melhoria passa pelo desenvolvimento de componentes externos, os denominados plugins, no sistema eGYM. Para além desta, considera-se ainda que podem ser efetuadas algumas melhorias ao sistema quanto ao desempenho de carregamento de informação existente nos servidores remotos e na *cloud*. Como descrito anteriormente (cf. secção 6.1.4), foram introduzidas algumas técnicas como o *caching* que permite aliviar alguma carga dos servidores, permitindo assim que um maior número de utilizadores tenha um acesso aos dados num tempo aceitável. No entanto, esta solução é apenas a base da pirâmide.

Podem ainda ser introduzidas algumas melhorias em relação às comunicações em tempo real com os servidores e as aplicações, no caso particular dos jogos multijogador. Ainda no tópico dos jogos multijogadores, considera-se ainda que existe alguma margem de progresso em relação aos algoritmos de emparelhamento dos jogadores. De momento, o sistema consegue cumprir com a necessidade de emparelhamento dos jogadores, mas o algoritmo é bastante simples e primitivo, existindo assim algum espaço para futuras melhorias, como a introdução de emparelhamento baseado nas classificações dos jogadores ou até emparelhamento com utilizadores específicos através de convites ou de salas de jogo personalizadas.

Como justificação para estas limitações, elas são essencialmente de limitações temporais e da grande dimensão do sistema pretendido. De facto, tendo em conta o tempo finito existente para a realização desta nova reengenharia, o estado do sistema anterior e o número de

problemas a resolver, percebeu-se desde bastante cedo que o trabalho requerido era demasiado extenso para uma única pessoa.

Para além disto, a infeliz situação pandémica existente por todo o mundo acabou por causar alguns atrasos no desenvolvimento de certas tarefas, assim como nas interações com os clientes do sistema.

É importante realçar que nunca em momento algum, apesar desta situação pandémica, que tenha existido de parte a parte um desinteresse na conclusão do projeto. No entanto e tendo em conta que esta situação atípica acabou por ser uma primeira (e esperemos que última) para ambas as partes, o processo de adaptação à situação acabou por se estender causando assim naturalmente alguns atrasos na comunicação entre as partes interessadas e na obtenção de um *feedback* mais imediato por parte dos clientes.

9.4 Apreciação final e pessoal

Num nível mais técnico, considero que os meus conhecimentos acabaram por crescer e progredir ao longo do desenvolvimento do projeto, aumentando assim as minhas competências técnicas. A utilização de diversas *frameworks* e bibliotecas distintas contribuíram de forma assertiva para esta evolução. Para além destas, a necessidade de analisar e estudar diversas soluções para os problemas encontrados nesta reengenharia do sistema, permitiram-me também adquirir novos conceitos e conhecimentos técnicos.

Num nível mais pessoal, sinto que a realização da presente dissertação e desta reengenharia permitiram-me solidificar as minhas competências pessoais. A grande dimensão do projeto e o valor esperado do mesmo, obrigaram-me desde o início que me empenhasse a 100% e que desenvolvesse ou melhorasse características pessoais que me permitissem ultrapassar todos os problemas encontrados e alcançar o objetivo pretendido.

É também um motivo de orgulho saber o que se espera deste sistema e da sua importância para os seus utilizadores e para o próprio instituto que frequentei durante todo o meu percurso académico superior. Pode-se descrever como uma sensação de realização, o facto de conseguir retribuir tudo o que a instituição do ISEP me ofereceu ao longo deste percurso académico, com o desenvolvimento deste sistema.

Para concluir, considero que o desenvolvimento desta dissertação teve um impacto bastante positivo em mim mesmo, tanto a nível pessoal como a nível técnico, saindo com um sentimento de satisfação pelo resultado obtido.

Referências

- (Allee, V. (2001) Understanding Value Networks. [ebook] Available at: [http://www.gurteen.com/gurteen/gurteen.nsf/98720698497ca9e4802568b5006aa4aa/549707c9f24ba6ae80256ef60027b138/\\$FILE/understanding-value-networks.pdf](http://www.gurteen.com/gurteen/gurteen.nsf/98720698497ca9e4802568b5006aa4aa/549707c9f24ba6ae80256ef60027b138/$FILE/understanding-value-networks.pdf) [Accessed 8 Feb. 2020].
- (Blackboard, 2020a) Teaching & Learning | Blackboard.com. [online] Available at: <https://www.blackboard.com/teaching-learning> [Accessed 10 Feb. 2020].
- (Blackboard, 2020b) Blackboard Learn | Blackboard.com. [online] Available at: <https://www.blackboard.com/teaching-learning/learning-management/blackboard-learn> [Accessed 10 Feb. 2020].
- (Brooke, 1996) SUS - A quick and dirty usability scale. [ebook] Available at: <https://pdfs.semanticscholar.org/13dd/d0ede672d91d905e56c52ed73216b17cf81.pdf> [Accessed 14 Feb. 2020].
- (Brown, 2018) *The C4 Model For Software Architecture*. [online] InfoQ. Available at: <https://www.infoq.com/articles/C4-architecture-model/>
- (Browne, 2019) *Best Free LMS Software For 2020*. [online] Learn.g2.com. Available at: <https://learn.g2.com/free-lms-software> [Accessed 9 October 2020].
- (Capterra, 2019) Best LMS Software | 2020 Reviews of the Most Popular Tools & Systems. [online] Available at: <https://www.capterra.com/infographics/most-popular/learning-management-system-software/> [Accessed 23 Feb. 2020].
- (Driessen, 2010) *A Successful Git Branching Model*. [online] nvie.com. Available at: <https://nvie.com/posts/a-successful-git-branching-model/>
- (eLearning Industry, 2017) 6 Things A Mobile Learning Management System Can Actually Do For You - eLearning Industry. [online] Available at: <https://elearningindustry.com/mobile-learning-management-system-can-do-6-things> [Accessed 23 Feb. 2020].
- (Fleming, 2019) Why Math Is So Hard for Some Students. [online] ThoughtCo. Available at: <https://www.thoughtco.com/why-math-seems-more-difficult-for-some-students-1857216> [Accessed 23 Feb. 2020].
- (João Fernandes and José Barros, 2018) e-GYM – Engineers Gym
- (Jorgenson, 2015) Why Value Creation is the Foundation of Business: How to define it, measure it, and manage it. [online] Medium. Available at: <https://medium.com/evergreen-business-weekly/why-value-creation-is-the-foundation-of-business-how-to-define-it-measure-it-and-manage-it-147c92b87aca> [Accessed 7 Feb. 2020].
- (Koen, 2002) Fuzzy Front End: Effective Methods, Tools, and Techniques. [ebook] Available at: https://web.stevens.edu/cce/NEW/PDFs/FuzzyFrontEnd_Old.pdf [Accessed 6 Feb. 2020].
- (Kotama and Saputra, 2020) *Proposed Model Of Multiplayer Matching Game Plugins Using Websocket In Moodle*. [online] Doi.org. Available at: <https://doi.org/10.3991/ijet.v14i11.10190> [Accessed 09 October 2020].
- (Kruchten, 1995) *Architectural Blueprints—The “4+1” Viewmodel Of Software Architecture*. [ebook] Available at: <https://arxiv.org/ftp/arxiv/papers/2006/2006.04975.pdf>
- (Luís Peixoto, 2019) e-Gym Engineers Gym
- (Maloney and Beilock, 2012) Math anxiety: who has it, why it develops, and how to guard against it. [ebook] Available at: https://www.cogsci.msu.edu/DSS/2014-2015/Beilock/TiCS_Final_Maloney&Beilock_2012.pdf [Accessed 23 Feb. 2020].
- (Meerts, 2013) Course Management Systems (CMS). [online] Educause.edu. Available at: <https://www.educause.edu/ir/library/pdf/DEC0302.pdf> [Accessed 4 Feb. 2020].
- (Moodle, 2020) Features - MoodleDocs. [online] Available at: <https://docs.moodle.org/38/en/Features> [Accessed 10 Feb. 2020].

- (Pappas, 2020) *The Best Learning Management Systems (2020 Update)*. [online] eLearning Industry. Available at: <<https://elearningindustry.com/the-best-learning-management-systems-top-list>> [Accessed 09 October 2020].
- (PISA, 2019) Results From PISA 2018. [ebook] OECD. Available at: https://www.oecd.org/pisa/publications/PISA2018_CN_PRT.pdf [Accessed 23 Feb. 2020].
- (Rich, 2000) Value Analysis Value Engineering. [ebook] Available at: https://www.urenio.org/tools/en/value_analysis.pdf [Accessed 5 Feb. 2020].
- (Saaty, 1990) How to make a decision: The Analytic Hierarchy Process. [ebook] Available at: https://www.researchgate.net/profile/Mohamed_Mourad_Lafifi/post/Problem_with_sub-criteria_code_for_multiple_decision_makers/attachment/59d644d679197b80779a0076/AS:450351808684035@1484383646513/download/How+to+Make+a+Decision+The+A+H+P.pdf [Accessed 20 Feb. 2020].
- (Statista, 2020) Smartphone users worldwide 2020 | Statista. [online] Available at: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> [Accessed 1 Feb. 2020].
- (TalentLMS, 2020) MS Software Features - TalentLMS' Full Feature List. [online] Available at: <https://www.talentlms.com/features> [Accessed 11 Feb. 2020].
- (theOECD, 2020) International student assessment (PISA) - Mathematics performance (PISA) - OECD Data. [online] Available at: <https://data.oecd.org/pisa/mathematics-performance-pisa.htm> [Accessed 1 Feb. 2020].
- (Watson and Watson, 2007) An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become.. [online] Cardinalscholar.bsu.edu. Available at: <http://cardinalscholar.bsu.edu/handle/123456789/194513> [Accessed 4 Feb. 2020].
- (Woodruff, 1997) Customer Value: The Next Source for Competitive Advantage. [ebook] pp.2-4. Available at: https://edisciplinas.usp.br/pluginfile.php/3725929/mod_resource/content/1/1.%20woodruff%201997%20Customer%20Value.pdf [Accessed 7 Feb. 2020].
- (Zeithaml, 1998) Consumer Perceptions of Price, Quality, and Value: A Means-End Model and Synthesis of Evidence. [ebook] Available at: <https://pdfs.semanticscholar.org/a54b/2631b97831fbea7d4155f4eff9a09220cb79.pdf> [Accessed 7 Feb. 2020].

10 Anexos

10.1 Análise de Valor

Esta secção reserva-se para o estudo da análise de valor do produto considerado na presente dissertação. Com esta análise pretende-se determinar e apresentar o valor que o produto oferece aos clientes e à organização, utilizando por base um conjunto de modelos e métodos.

10.1.1 Introdução

O presente capítulo aborda em detalhe a análise de valor do produto construído através do aprofundamento de conceitos adjacentes, que permitem descrever o valor que o sistema oferece aos seus clientes e a sua organização

Para Nick Rich (Rich, 2000) uma análise de valor representa um processo que utiliza uma abordagem organizada para aumentar o valor dos produtos gerados, utilizando várias técnicas diferentes para atingir esse objetivo. A abordagem da análise de valor é quase universal e pode ser utilizada para analisar produtos existentes ou para serviços oferecidos por empresas exteriores.

No centro do seu processo, na análise de valor existe uma preocupação em identificar e eliminar recursos alocados a produtos e serviços que não adicionem valor real ao cliente nem ao produto e que simultaneamente aumentem os custos de processo de produção ou de prestação do serviço (Rich, 2000).

Nick Rich (Rich, 2000) refere ainda que o processo da análise de valor é utilizado para oferecer ao cliente um produto ou serviço com o maior valor real possível a um custo mínimo, ao invés de substituir um produto existentes por uma solução de valor inferior.

Desde o seu primeiro desenvolvimento no final da década de 1940 depois da Segunda Guerra Mundial, que a abordagem básica da análise de valor evoluiu e foi complementada com novas técnicas (Rich, 2020). Nos dias de hoje devido as altas pressões competitivas, as empresas

sentem a necessidade de analisar todos os seus produtos, na tentativa de oferecer altos níveis de customização dos seus produtos sem grandes penalizações (Rich, 2020). As empresas de grande dimensão procuram utilizar estes processos com os seus fornecedores para estender os benefícios da análise valor em toda a sua cadeia de distribuição, sendo por isso uma abordagem universal para todo o tipo de empresas, sejam estas pequenas ou grandes (Rich, 2020).

Este capítulo é constituído pelas seguintes partes:

- desenvolvimento do modelo NCD (*New Concept Development*) de Peter Koen (Koen, 2002) procurando detalhar as fases do processo de inovação, atendendo especial detalhe ao *Fuzzy Front End*;
- desenvolvimento de conceitos de difícil compreensão, como o valor, valor para o cliente e valor percebido, procurando fazer um enquadramento dos mesmos através de uma tabela de benefícios/sacrifícios;
- enunciação da proposta de valor;
- estruturação do modelo de negócio através de um Modelo Canvas;
- desenvolvimento do método AHP para auxiliar a tomada de decisão,

10.1.2 Modelo NCD

O processo de inovação pode ser dividido em três áreas: *Fuzzy Front End* (FFE), *New Product Development* (NPD) e a Comercialização. É geralmente considerado que as maiores oportunidades da melhoria de processo geral da inovação encontram-se na primeira área, a FFE (Koen, 2002).

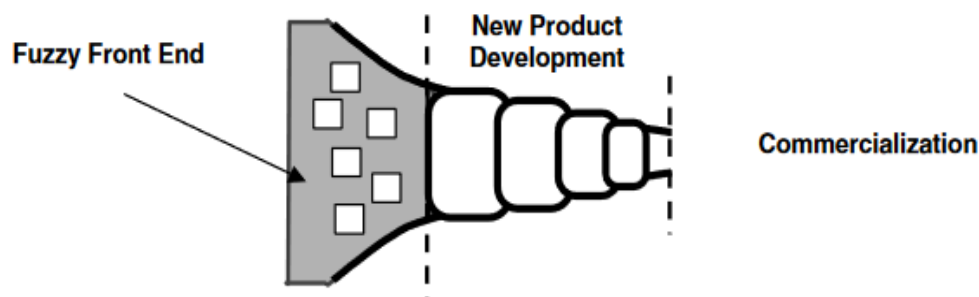


Figura 173 – O processo de inovação dividido em três partes (Koen, 2002)

Quando se trata de um processo de inovação é sempre necessário que existam boas práticas para auxiliar quem procurar implementar este tipo de processo. Desde o seu início que boas práticas são conhecidas, em particular para a área NDC. No entanto para a área da FFE, estas são inexistentes ou escassas, sendo que muitas das práticas que auxiliam a porção NPD não são aplicáveis a FFE (Koen, 2002).

Apesar disto a falta de informação sobre as melhores práticas faz da FFE uma das maneiras mais promissoras de melhorar o processo de inovação. No entanto a falta de informação e de referências torna o processo um pouco complicado devido a falta de termos comuns e definição chaves, criando assim alguma confusão na distinção entre as diferentes partes do processo de inovação (Koen, 2002).

Peter Koen (Koen, 2002) refere que modelo NCD, representado na Figura 173, nasce para resolver o problema anteriormente referido, com a intenção de fornecer informações e conceitos comuns para a FFE. O novo modelo é dividido em três componentes principais: o motor, cinco elementos interiores e por fatores que influenciam a capacidade da organização.

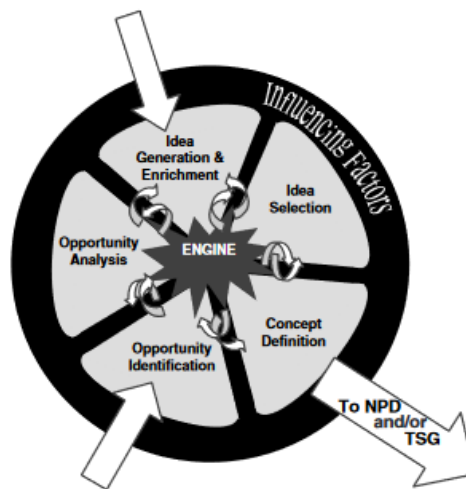


Figura 174 – Modelo NCD

É possível observar através da Figura 174, que a forma circular do modelo sugere que o ponto de partida do processo acontece na identificação ou na geração e enriquecimento de oportunidades, representado na figura através da seta incidente. Por fim o processo termina assim que a definição de conceitos acontece, entrando nos processos NPD ou de *technology state gate* (TSG).

Para além relações e das direções de entrada e saída do modelo é possível observar as três componentes principais do modelo referidas anteriormente. Para uma melhor compreensão das mesmas, essas três componentes serão analisadas em detalhe nas seguintes subsecções.

10.1.2.1 Motor

O motor, colocado na parte central do modelo, representa a gestão executiva de topo, alimentando os cinco elementos do modelo NCD.

10.1.2.2 Elementos

Na seguinte lista serão detalhados os cinco elementos do modelo NCD, tendo em consideração o projeto desenvolvido na presente dissertação.

10.1.2.2.1 Identificação de oportunidades

Nesta fase inicial a organização procura identificar oportunidades que possam ser do seu interesse de persuadir.

Os baixos índices de aproveitamento dos alunos nas áreas das Engenharias e principalmente nas áreas das Matemáticas tem aumentado a procura de novas abordagens que tentem contornar essa situação.

O ensino-aprendizagem da Matemática no ensino superior, e em particular no ISEP, sofre de problemas semelhantes dificultando a progressão académica de grande parte dos seus estudantes.

Surge assim a oportunidade de criar um sistema de ensino-aprendizagem que ofereça numa única plataforma conteúdos lúdicos para os seus utilizadores de diferentes tipos e formatos, tentando facilitar os processos de aprendizagem e que promova o pensamento crítico dos seus utilizadores.

Com a crescente integração das tecnologias no Ensino, um grupo de professores do Departamento de Matemática encararam com bons olhos esta oportunidade e procuraram tomar a iniciativa com o desenvolvimento de uma plataforma LMS, numa tentativa de suportar o processo de aprendizagem dos seus alunos.

10.1.2.2.2 Análise de oportunidades

Depois de identificada a oportunidade delineada torna-se necessário analisar a mesma para se saber se esta é viável ou não. Como tal são efetuados estudos de mercado onde se procura perceber se a oportunidade identificada é inovadora e têm mercado.

Para o presente projeto são efetuados estudos sobre o estado da arte, identificando e analisando soluções que se assemelhem de algum modo à oportunidade identificada e que se possam considerar como concorrentes.

Para cada uma delas são anotadas todas as características relevantes de modo a que no fim se possa fazer uma análise comparativa de modo a verificar se alguma das soluções no mercado já oferece aquilo que se pretende oferecer, isto é se alguma das soluções já no mercado resolve de uma maneira global os problemas que originam a oportunidade identificada.

No estudo efetuado (cf. secção 2.2.4) é possível observar que apesar de que as soluções existentes no mercado se apresentam bastante robustas e com um grande número de características essenciais, não conseguem na sua totalidade oferecer o pretendido, tornando assim a oportunidade encontrada viável de ser explorada.

10.1.2.2.3 Identificação ou geração e enriquecimento de ideias

Tendo em conta o problema a resolver e a análise efetuada sobre o negócio existente, são desenvolvidas e/ou identificadas ideias que permitam ir de encontro as necessidades dos clientes e que permitam adicionar valor ao produto pretendido.

A identificação ou enriquecimento de ideias não acontecem num único momento temporal, sendo o processo repetido ao longo de todo o ciclo de desenvolvimento do produto. A medida que novas necessidades aparecem ou que novas reuniões com o orientador e/ou clientes acontecem novas ideias são geradas. Estas novas ideias introduzidas podem ser completamente novas ou podem ser identificadas para colmatar possíveis falhas em ideias anteriores.

Para além das reuniões com as partes interessadas, durante a própria construção do produto são identificadas ideias potencialmente favoráveis para o valor do mesmo. Estas ideias podem surgir através da implementação de funcionalidades requisitadas ou através de um *brainstorming* individual.

Para o presente projeto são identificadas as seguintes ideias:

- O sistema deve oferecer configurabilidade e extensibilidade dos seus conteúdos.
- O sistema deve oferecer uma maior capacidade de configurabilidade em termos de autenticação e autorização dos seus utilizadores.
- O sistema deve adotar componentes/módulos externos, tornando o sistema o mais modular possível.
- O sistema deve implementar o conceito de competições, oferecendo aos seus utilizadores atividades competitivas.
- O sistema deve oferecer aos seus utilizadores conteúdos *premium* que podem ser adquiridos através de um pagamento.
- O sistema deve aprofundar o conceito de estatística existente, oferecendo um maior número de variáveis.
- Deve ser efetuado um *redesign* do sistema a nível de interfaces.

10.1.2.2.4 Seleção de ideias

Na seleção de ideias procede-se a escolha das ideias que se considerem mais importantes e que não prejudiquem o desenvolvimento da solução, tendo em conta o tempo e as limitações existentes. Foram selecionadas as seguintes:

- O sistema deve oferecer configurabilidade e extensibilidade dos seus conteúdos.

- O sistema deve oferecer uma maior capacidade de configurabilidade em termos de autenticação e autorização dos seus utilizadores.
- O sistema deve adotar componentes/módulos externos, tornando o sistema o mais modular possível.
- O sistema deve implementar o conceito de competições, oferecendo aos seus utilizadores atividades competitivas.
- O sistema deve aprofundar o conceito de estatística existente, oferecendo um maior número de variáveis.
- Deve ser efetuado um *redesign* do sistema a nível de interfaces.

10.1.2.2.5 Definição de conceitos

Nesta última fase do modelo NCD é normalmente efetuada uma reunião com a organização procurando atrair e argumentar o investimento no negócio, considerando as necessidades dos clientes.

Para o presente tema, é nesta fase que se define a introdução dos novos conceitos de configurabilidade e extensibilidade no sistema, assim como novas funcionalidades a implementar como as competições. Depois desta definição estar concluída, inicia-se a fase do desenvolvimento de um novo produto.

10.1.2.3 Fatores influenciadores

Os fatores influenciadores consistem nas capacidades organizacionais, nos fatores externos (políticos, sociais, económicos e tecnológicos) e nas ciências facilitadoras (internas e externas) que possam estar envolvidas. Todos estes fatores afetam a organização como um todo através da sua influência no processo de inovação, sendo estes incontroláveis pela empresa (Koen, 2002).

Normalmente, na análise dos fatores externos é utilizada a análise PEST. O acrónimo representa quatro fatores exteriores utilizados na gestão estratégia das empresas sendo estes: Política, Económica, Social e Tecnológica.

Para o presente projeto são considerados os seguintes fatores externos:

- **Tecnológicos:** mudanças nas tecnologias são sempre um fator importante para sistemas deste tipo. Com possíveis inovações nas tecnologias, o sistema pode ser afetado introduzindo assim custos e necessidades de inovações tecnológicas;
- **Sociais:** fatores sociais como estilos de vida, envelhecimento da população e comportamentos podem influenciar o número de utilizadores alvo da solução;

10.1.3 Valor, valor para o cliente e valor percebido

Numa primeira fase do processo de inovação, o FFE, existe sempre um grande número de ideias que tornam o processo bastante confuso. Como tal torna-se essencial perceber quais as ideias que podem adicionar valor ao produto. A definição e estudo aprofundado destes conceitos agiliza a transição do primeiro processo para o segundo, ajudando na seleção de ideias e oportunidades que aumentem o valor do produto.

Nas seguintes secções serão abordados em maior detalhe conceitos relacionados com a definição de valor, começando pela definição do próprio valor, seguido de uma análise do conceito de valor para o cliente, terminando com a definição do valor percebido. Será ainda reservada uma pequena área para a contextualização do valor percebido para o cliente tendo em consideração o tema da presente dissertação.

10.1.3.1 Valor

Todos os negócios começam numa simples criação de valor. Na essência do negócio surge o objetivo de criar e agregar valor de uma maneira suficiente para se conseguir obter lucro com esse mesmo valor.

O valor pode ser gerado através de trabalho físico ou através de trabalho criativo. Depois de criado, o negócio procura vender ou trocar o valor aos clientes, tentando capturar alguma parte desse valor como lucro (Jorgenson, 2015). No entanto nem todo o trabalho cria valor!

O valor pode ainda existir através de um produto ou de um serviço, podendo para o último ser tangível ou não. É importante ainda perceber que a medição do valor estará sempre associada as necessidades especificadas de cada consumidor. Para alguns um dado produto ou serviço pode criar imenso valor, tendo em conta as suas necessidades e requisitos, mas para outros o valor gerado pode ser relativamente pequeno.

10.1.3.2 Valor para o cliente

O valor para o cliente é a ponderação das qualidades de um produto, sendo estas os seus atributos ou desempenho, e as consequências do seu uso, que podem facilitar ou dificultar o alcance dos objetivos e propósitos dos clientes para cada situação específica de uso do produto (Woodruff, 1997).

10.1.3.3 Valor percebido

Valor percebido pode ser definido para Zeithaml (Zeithaml, 1998) como “uma avaliação geral do consumidor sobre a utilidade de um produto (ou serviço) com base nas perceções do que é recebido e do que é dado”¹⁹.

Valarie Zeithaml (Zeithaml, 1998) refere ainda este valor percebido pode variar entre os consumidores tendo em conta o que é recebido (uns podem desejar volume, outros alta

¹⁹ Tradução livre do autor. No original “the consumer’s overall assessment of the utility of a product (or service) based on perceptions of what is received and what is given”.

qualidade, outros ainda conveniência) e pelo que é oferecido (alguns apenas se preocupam com valores monetários, outros com o tempo e esforço), sendo que o valor representa uma troca dos componentes recebidos e dados.

No âmbito do eGYM o valor percebido representa-se através de um enquadramento dos vários benefícios/sacrifícios existentes no valor gerado, representados na Tabela 2

Tabela 3 – Sacrifícios/Benefícios

| | Produto | Serviço | Relação |
|--------------------|--|--|--|
| Benefícios | <ul style="list-style-type: none"> • Melhoria da experiência do utilizador • Qualidade do produto • Customização do produto • Desempenho | <ul style="list-style-type: none"> • Desempenho • Confiabilidade | <ul style="list-style-type: none"> • Apreciação dos utilizadores • Fortalecimento da imagem • Confiança |
| Sacrifícios | <ul style="list-style-type: none"> • Custos de desenvolvimento | <ul style="list-style-type: none"> • Preço | <ul style="list-style-type: none"> • Tempo • Esforço |

10.1.4 Proposta de valor

Uma proposta de valor reflete uma afirmação através da qual se pretende demonstrar a correspondência entre as necessidades dos consumidores e os benefícios que resultam de utilizar o produto desenvolvido. Devem ser curtas, simples e eficazes. Para o eGYM é definida a seguinte proposta de valor:

“O eGYM cria um sistema extremamente simples e intuitivo de se utilizar que oferece aos seus utilizadores uma plataforma de suporte no percurso curricular de aprendizagem, através da disponibilização de conteúdos lúdicos. O sistema é construído com o objetivo de suavizar a progressão dos alunos no ensino, disponibilizando conteúdos e atividades de diversos formatos, como vídeo-aulas, jogos e exercícios. Para melhor a experiência dos seus utilizadores o eGYM oferece aos demais um sistema de progressão, garantindo assim uma escala gradual e continua da dificuldade dos conteúdos e atividades, evitando assim que os seus utilizadores avancem para níveis posteriores sem estarem preparados para tal. Para além dos alunos dos diversos níveis de ensino, o sistema oferece ainda uma plataforma de gestão tornando-se assim numa ferramenta ideal para os próprios professores de ensino.”

10.1.5 Modelo Canvas

Considerando que um modelo de negócio é a forma como uma organização se estrutura para gerar e capturar valor, torna-se necessário utilizar estratégias que auxiliem todo este processo para que se permita combinar os meios da empresa para entregar valor às partes interessadas e conseqüentemente capturar valor para a própria organização.

Uma forma bastante prática de estruturar todo o negócio para a geração de valor é a utilização do modelo Canvas. O modelo Canvas é uma ferramenta de gestão estratégica, que permite desenvolver modelos de negócios novos ou existentes, contendo nove diferentes blocos únicos que representam diferentes áreas de negócio.

Para a presente dissertação, o modelo de negócio desenvolvido encontra-se representado na Figura 175:

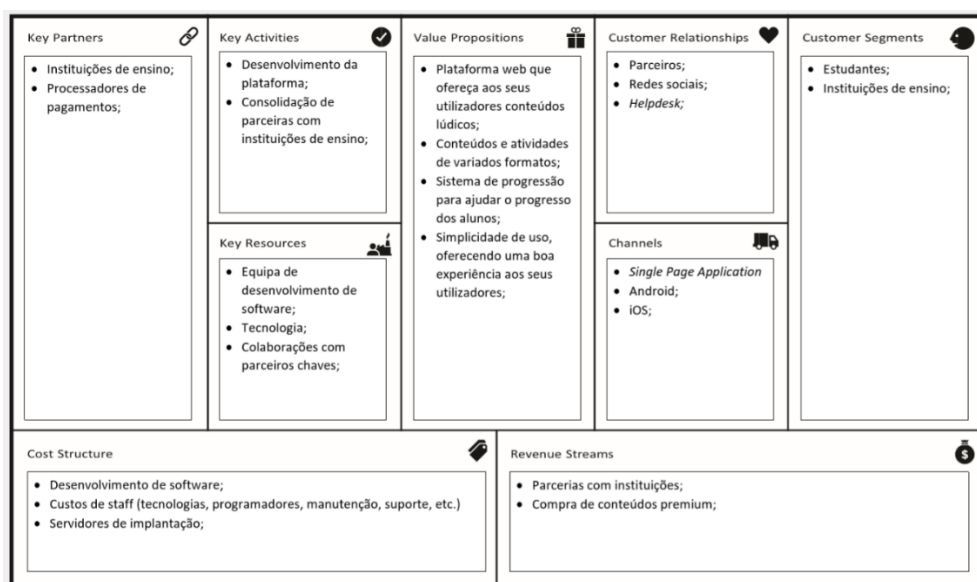


Figura 175 – Modelo Canvas

São considerados parceiros chaves instituições de ensino e processadores de pagamento. Quanto aos primeiros, estas parcerias com diferentes instituições de ensino nacionais ou internacionais permitem aumentar o número de utilizadores na plataforma e conseqüentemente expandir o reconhecimento da mesma, fazendo com que o sistema se torne uma ferramenta oficial que os seus alunos possam utilizar como suporte de aprendizagem. Os processadores de pagamento são também considerados parceiros chaves tentando assim oferecer um maior número de diferentes opções de pagamento.

As atividades chaves do negócio passam pelo desenvolvimento da plataforma e pela consolidação de parcerias, tentando fortalecer as parcerias com as diferentes instituições de ensino.

Recursos chave são os recursos essenciais para o desenvolvimento e melhorias do sistema. Como tal, considera-se a equipa de desenvolvimento do software, tecnologias utilizadas para a produção do sistema e a colaboração com parceiros chave permitindo assim recolher *feedback* constante.

No centro do modelo encontra-se a proposta de valor, que obviamente coincide com a proposta enunciada na subsecção anterior.

Quanto a relação com os clientes considera-se relações indiretas através de parceiros chave e relações diretas através de redes sociais.

Para os canais de distribuição considera-se como objetivo principal a construção de uma SPA (*Single Page Application*) através de uma aplicação *web*, sendo objetivos secundários e para uma fase posterior do ciclo de vida do sistema, o desenvolvimento de aplicações móveis para os sistemas Android e iOS.

Os segmentos de mercados definidos para o negócio são os estudantes e instituições de ensino que pretendam estabelecer parcerias com a organização.

Relativamente aos custos, a organização apresenta custos relacionados com o desenvolvimento do software e de staff, assim como os servidores de alojamento do sistema. Por outro lado, são consideradas fontes de rendimento as parcerias com instituições e as compras de conteúdo premium.

10.1.6 Método AHP

Os métodos de apoio a decisão multicritério permitem a priorização de alternativas em situações conflituosas, procurando auxiliar o processo de escolha da melhor opção tendo em conta as restrições e objetivos do problema.

Um dos principais métodos no ambiente das decisões multicritério é o Método de Análise Hierárquica, do inglês *Analytic Hierarchy Process* (AHP), desenvolvido pelo professor Thomas L. Satty em 1980.

Este método utiliza critérios qualitativos bem como quantitativos no processo de avaliação procurando assim determinar a melhor alternativa para o problema a enfrentar. Para além dos critérios, o método divide o problema de decisões em níveis hierárquicos, facilitando assim a sua compreensão e avaliação.

O AHP encontra-se dividido em sete fases, sendo estas:

1. Construção da árvore hierárquica de decisão;
2. Comparação de alternativas e critérios;
3. Prioridade relativa de cada critério;

4. Avaliar a consistência das prioridades relativas;
5. Construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas;
6. Obter a propriedade composta para as alternativas;
7. Escolha da alternativa;

Para melhor compreensão cada uma destas fases são discutidas e analisadas em detalhe nas subsecções seguintes.

10.1.6.1 Fase 1: Construção da árvore hierárquica de decisão

Nesta primeira fase é definido o problema, estruturando o mesmo num diagrama hierárquico, decompondo o problema/decisão numa hierarquia, composta, no mínimo, de um objetivo, critérios e alternativas.

No âmbito do projeto, a introdução de competições no sistema representa um problema a resolver, procurando determinar qual a melhor abordagem possível para tal.

Como tal são considerados os seguintes critérios:

- Desempenho, que representa o impacto da nova funcionalidade no sistema.
- Custo, que representa a relação entre tempo/recursos necessários para implementar a funcionalidade desejada.
- Dificuldade de implementação, que representa a dificuldade do processo de implementação da funcionalidade.

Quanto as alternativas, para o problema inicial são consideradas duas alternativas: (i) a nova funcionalidade é implementada sobre uma API existente, denominada de “API Principal” (cf. secção 5.2); (ii) a nova funcionalidade é implementada numa nova API dedicada, denominada de API “*Standalone*”²⁰ para a restante análise.

²⁰ Por API *Standalone* compreende-se a criação de uma nova API completamente dedicada para o conceito das Competições.

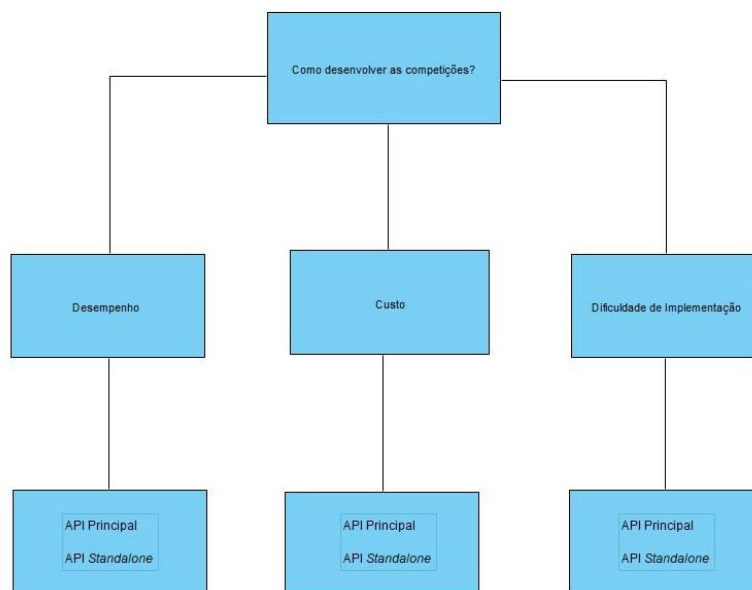


Figura 176 – Árvore hierárquica de decisão

10.1.6.2 Fase 2: Comparação entre os elementos da hierarquia

Esta segunda fase consiste no estabelecimento das prioridades entre os elementos para cada nível de hierarquia, por meio de uma matriz de comparação. Como tal Saaty (Saaty, 1990) definiu uma Escala Fundamental, representada na Tabela 3 que não excede um total de nove fatores, a fim de manter uma matriz consistente.

Tabela 4 – Escala Fundamental adaptada de Saaty (Saaty, 1990)

| Nível de importância | Definição | Explicação |
|----------------------|-------------------------|---|
| 1 | Igual importância | As duas atividades contribuem igualmente para o objetivo. |
| 3 | Fraca importância | A experiência e o julgamento favorecem levemente uma atividade em relação à outra. |
| 5 | Forte importância | A experiência e o julgamento favorecem fortemente uma atividade em relação à outra. |
| 7 | Muito forte importância | Uma atividade é muito fortemente favorecida em relação à outra. |

| | | |
|---------|----------------------|--|
| 9 | Importância absoluta | A evidência favorece uma atividade em relação à outra com o mais alto grau de certeza. |
| 2,4,6,8 | Valor intermédios | Quando se procura uma condição de compromisso entre duas definições. |

Tendo em conta os critérios definidos anteriormente e a Escala Fundamental de Saaty, é construída a seguinte matriz de comparação, representada na Tabela 4:

Tabela 5 – Matriz de comparação

| | Desempenho | Custo | Dificuldade de implementação |
|-------------------------------------|-------------------|--------------|-------------------------------------|
| Desempenho | 1 | 2 | 4 |
| Custo | 1/2 | 1 | 3 |
| Dificuldade de implementação | 1/4 | 1/3 | 1 |

Analisando a Tabela 4 é possível observar que o critério mais importante é o Desempenho, seguido do critério de Custo, terminando com o critério de Dificuldade de Implementação.

10.1.6.3 Fase 3: Prioridade relativa de cada critério

Depois de construída a matriz de comparação, nesta terceira fase procede-se a normalização dos valores da matriz de comparação, com o objetivo de igualar todos os critérios a uma mesma unidade, sendo para isso cada valor da matriz dividido pelo total da sua coluna respetiva.

Tabela 6 – Matriz de comparação dos critérios de segundo nível

| | Desempenho | Custo | Dificuldade de implementação |
|-------------------|-------------------|--------------|-------------------------------------|
| Desempenho | 1 | 2 | 4 |

| | | | |
|-------------------------------------|-----|------|---|
| Custo | 1/2 | 1 | 3 |
| Dificuldade de implementação | 1/4 | 1/3 | 1 |
| Soma | 7/4 | 10/3 | 8 |

Tabela 7 – Matriz normalizada dos critérios de segundo nível

| | | | |
|-------------------------------------|-------------------|--------------|-------------------------------------|
| | Desempenho | Custo | Dificuldade de implementação |
| Desempenho | 4/7 | 3/5 | 1/2 |
| Custo | 2/7 | 3/10 | 3/8 |
| Dificuldade de implementação | 1/7 | 1/10 | 1/8 |

Assim que obtida a matriz normalizada procede-se à obtenção do vetor de propriedades que tem por objetivo identificar a ordem de importância de cada critério, sendo para isso calculado a média aritmética dos vetores da matriz normalizada.

Tabela 8 – Matriz normalizada dos critérios de segundo nível com prioridades relativas

| | | | | |
|-------------------------------------|-------------------|--------------|-------------------------------------|----------------------------|
| | Desempenho | Custo | Dificuldade de implementação | Prioridade relativa |
| Desempenho | 4/7 | 3/5 | 1/2 | 0.5571 |
| Custo | 2/7 | 3/10 | 3/8 | 0.3202 |
| Dificuldade de implementação | 1/7 | 1/10 | 1/8 | 0.1226 |

Através da análise da Tabela 7 é possível verificar que o Desempenho corresponde a 56% da prioridade, seguido do Custo com 32% e por último da Dificuldade de implementação com 12%.

10.1.6.4 Fase 4: Avaliar a consistência das prioridades relativas

O primeiro passo desta quarta etapa é calcular a Razão de Consistência (RC) para medir o quanto os julgamentos foram consistentes em relação a grande amostrar de juízos aleatórios. A fórmula da Razão de Consistência é a seguinte:

$$RC = \frac{IC}{IR} \quad (1)$$

Para a fórmula da Razão de Consistência é primeiro necessário calcular o Índice de Consistência (IC) e o Índice Aleatório (IR).

Para o IC é necessário primeiro obter os valores de λ_{max} através da fórmula:

$$Ax = \lambda_{max}x \quad (2)$$

, onde x é o vetor próprio.

Para calcular o valor do λ_{max} utiliza-se a da matriz de comparação (A) e o vetor próprio (x , que corresponde a prioridade relativa).

$$\begin{matrix} A & & x & \approx & \lambda_{max} \\ \begin{bmatrix} 1 & 2 & 4 \\ 1/2 & 1 & 2 \\ 1/4 & 1/2 & 1 \end{bmatrix} & & \begin{bmatrix} 0.56 \\ 0.32 \\ 0.12 \\ 1.67 \\ 0.96 \\ 0.37 \end{bmatrix} & \approx & \begin{bmatrix} 0.56 \\ 0.32 \\ 0.12 \\ 0.56 \\ 0.32 \\ 0.12 \end{bmatrix} \end{matrix}$$

$$\lambda_{max} = average\{1.67/0.56, 0.96/0.32, 0.37/0.12\} \quad (3)$$

$$\lambda_{max} = 3.02 \quad (4)$$

Com o λ_{max} calculado procede-se ao cálculo do IC utilizando a fórmula:

$$IC = \frac{\lambda_{max} - n}{n - 1} \quad (5)$$

, onde n representa o número de critérios definidos.

$$IC = \frac{3.02 - 3}{3 - 1} \quad (6)$$

$$IC = 0.01 \quad (7)$$

O IR representa um índice aleatório calculado pelas matrizes quadradas de ordem n pelo Laboratório Nacional de *Oak Ridge*. A Tabela 8 define os valores de IR em função do número de critérios:

Tabela 9 – Valores de IR para matrizes quadradas de ordem n

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0.00 | 0.00 | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 | 1.51 | 1.48 | 1.56 | 1.67 | 1.59 |

Observando a Tabela 8 e sabendo que o número de critérios presentes são 3, o valor do IR será de 0.58.

Com o IC e o IR obtidos é possível agora calcular o valor do RC:

$$RC = \frac{0.01}{0.58} \quad (8)$$

$$RC = 0.018 \quad (9)$$

Como o resultado obtido do RC é inferior a 0.1, pode-se concluir que os valores das prioridades relativas são consistentes.

10.1.6.5 Fase 5: Construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas selecionadas Fase 5: Construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas selecionadas

Nesta fase todos os procedimentos para a construção da matriz de comparação e para a determinação de cada critério devem ser novamente feitos, observando agora a importância relativa de cada uma das alternativas.

- Desempenho

Tabela 10 – Matriz de comparação do critério Desempenho

| Desempenho | API Principal | API Standalone |
|-----------------------|----------------------|-----------------------|
| API Principal | 1 | 1/3 |
| API Standalone | 3 | 1 |

Tabela 11 – Matriz de comparação do critério Desempenho de segundo nível

| Desempenho | API Principal | API Standalone |
|-----------------------|----------------------|-----------------------|
| API Principal | 1 | 1/3 |
| API Standalone | 3 | 1 |
| Soma | 4 | 4/3 |

Tabela 12 – Matriz normalizada para o critério de Desempenho de segundo nível

| Desempenho | API Principal | API Standalone |
|----------------------|----------------------|-----------------------|
| API Principal | 1/4 | 1/4 |

| | | |
|-----------------------|-----|-----|
| API Standalone | 3/4 | 3/4 |
|-----------------------|-----|-----|

Tabela 13 – Matriz normalizada para o critério de Desempenho de segundo nível com prioridades relativas

| Desempenho | API Principal | API Standalone | Prioridade relativa |
|-----------------------|----------------------|-----------------------|----------------------------|
| API Principal | 1/4 | 1/4 | 0.25 |
| API Standalone | 3/4 | 3/4 | 0.75 |

Para o critério de Desempenho obtém-se o seguinte vetor de propriedades:

$$\text{Vetor propriedade} = \begin{bmatrix} 0.25 \\ 0.75 \end{bmatrix} \quad (10)$$

- Custo

Tabela 14 – Matriz de comparação para o critério Custo

| Custo | API Principal | API Standalone |
|-----------------------|----------------------|-----------------------|
| API Principal | 1 | 2 |
| API Standalone | 1/2 | 1 |

Tabela 15 – Matriz de comparação para o critério Custo de segundo nível

| Custo | API Principal | API Standalone |
|-----------------------|----------------------|-----------------------|
| API Principal | 1 | 2 |
| API Standalone | 1/2 | 1 |
| Soma | 3/2 | 3 |

Tabela 16 Matriz normalizada para o critério Custo de segundo nível

| Custo | API Principal | API Standalone |
|-----------------------|----------------------|-----------------------|
| API Principal | 2/3 | 2/3 |
| API Standalone | 1/3 | 1/3 |

Tabela 17 – Matriz normalizada para o critério Custo de segundo nível com prioridades relativas

| Custo | API Principal | API Standalone | Prioridade relativa |
|-----------------------|----------------------|-----------------------|----------------------------|
| API Principal | 2/3 | 2/3 | 0.6667 |
| API Standalone | 1/3 | 1/3 | 0.3333 |

Para o critério de Custo obtém-se o seguinte vetor de propriedades:

$$\text{Vetor propriedade} = \begin{bmatrix} 0.6667 \\ 0.3333 \end{bmatrix} \quad (11)$$

- Dificuldade de implementação

Tabela 18 – Matriz de comparação para o critério Dificuldade de implementação

| Dificuldade de implementação | API Principal | API Standalone |
|-------------------------------------|----------------------|-----------------------|
| API Principal | 1 | 2 |
| API Standalone | 1/2 | 1 |

Tabela 19 – Matriz de comparação para o critério Dificuldade de implementação de segundo nível

| Dificuldade de implementação | API Principal | API Standalone |
|-------------------------------------|----------------------|-----------------------|
| API Principal | 1 | 2 |
| API Standalone | 1/2 | 1 |
| Soma | 3/2 | 3 |

Tabela 20 – Matriz normalizada para o critério Dificuldade de implementação de segundo nível

| Dificuldade de implementação | API Principal | API Standalone |
|-------------------------------------|----------------------|-----------------------|
| API Principal | 2/3 | 2/3 |
| API Standalone | 1/3 | 1/3 |

Tabela 21 – Matriz normalizada para o critério Dificuldade de implementação de segundo nível com prioridades relativas

| Dificuldade de implementação | API Principal | API <i>Standalone</i> | Prioridade relativa |
|------------------------------|---------------|-----------------------|---------------------|
| API Principal | 2/3 | 2/3 | 0.6667 |
| API <i>Standalone</i> | 1/3 | 1/3 | 0.3333 |

Para o critério de Dificuldade de implementação obtém-se o seguinte vetor de propriedades:

$$\text{Vetor propriedade} = \begin{bmatrix} 0.6667 \\ 0.3333 \end{bmatrix} \quad (12)$$

10.1.6.6 Fase 6: Obter a propriedade composta para as alternativas

Nesta penúltima etapa, são obtidas as propriedades compostas das alternativas, multiplicando os valores obtidos para cada critério com os valores das prioridades relativas da matriz normalizada da comparação inicial (cf. Tabela 7):

$$\begin{bmatrix} 0.25 & 0.6667 & 0.6667 \\ 0.75 & 0.3333 & 0.3333 \end{bmatrix} \times \begin{bmatrix} 0.56 \\ 0.32 \\ 0.12 \end{bmatrix} = \begin{bmatrix} 0.4333 \\ 0.5667 \end{bmatrix} \quad (13)$$

10.1.6.7 Fase 7: Escolha da alternativa

Finalmente na última fase do método AHP são avaliados os valores obtidos na Fase 6, de modo a escolher a melhor alternativa possível tendo em conta os cálculos realizados. Para o presente trabalho é possível verificar que a primeira alternativa, desenvolvimento de competições na API principal, apresenta 43% enquanto que a segunda, desenvolvimento sobre uma API *standalone*, apresenta 57% representando assim segundo o método AHP a alternativa mais indicada em função dos critérios definidos.

$$\begin{bmatrix} 0.25 & 0.6667 & 0.6667 \\ 0.75 & 0.3333 & 0.3333 \end{bmatrix} \times \begin{bmatrix} 0.56 \\ 0.32 \\ 0.12 \end{bmatrix} = \begin{bmatrix} 0.4333 \\ \mathbf{0.5667} \end{bmatrix} \quad (14)$$