

**MESTRADO**  
MULTIMÉDIA - ESPECIALIZAÇÃO EM TECNOLOGIAS

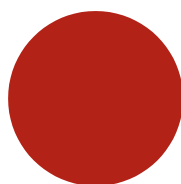
# **LUDIFICAÇÃO DO ENSINO DA PROGRAMAÇÃO: UM CASO DE ESTUDO**

Jorge Adolfo David Castro Monteiro

**M**  
2017

FACULDADES PARTICIPANTES:

**FACULDADE DE ENGENHARIA  
FACULDADE DE BELAS ARTES  
FACULDADE DE CIÊNCIAS  
FACULDADE DE ECONOMIA  
FACULDADE DE LETRAS**





# **Ludificação do Ensino da Programação: Um Caso de Estudo**

**Jorge Adolfo David Castro Monteiro**

Mestrado em Multimédia da Universidade do Porto

Orientador: António Fernando Vasconcelos Cunha Castro Coelho (Professor Auxiliar)

12 de Julho de 2017



© Jorge Monteiro, 2017

# **Ludificação do Ensino da Programação: Um Caso de Estudo**

**Jorge Adolfo David Castro Monteiro**

Mestrado em Multimédia da Universidade do Porto

Aprovado em provas públicas pelo Júri:

Presidente: Maria Teresa Magalhães da Silva Pinto de Andrade (Professor Auxiliar)

Vogal Externo: Leonel Caseiro Morgado (Professor Auxiliar com Agregação)

Orientador: António Fernando Vasconcelos Cunha Castro Coelho (Professor Auxiliar)



# Resumo

Com esta dissertação pretende-se encontrar uma solução para o problema que existe entre os alunos que entram no ensino superior, relacionado com o ensino e aprendizagem da Programação. Muitos destes alunos têm uma experiência heterogénea na prática da programação, o que faz com que a velocidade a que a matéria é lecionada nas aulas seja apropriada para apenas uma parte dos estudantes, podendo proporcionar angústia ou tédio.

O objetivo é então atender às necessidades de cada aluno, individualizando o seu percurso de aprendizagem na unidade curricular e fazendo aumentar a sua motivação e eficácia deste processo. Para isso foi realizado um caso de estudo aplicado à unidade curricular de Fundamentos da Programação do 1º ano, 1º semestre do MIEIC da Faculdade de Engenharia da Universidade do Porto, onde foram utilizados elementos e técnicas de ludificação para criar um sistema que possa ser implementado na plataforma moodle. Esse sistema é regulado por um sistema de pontuação e de níveis para fomentar a progressão do aluno na disciplina ao seu ritmo e de acordo com as suas capacidades em cada momento. À medida que o aluno vai avançando na matéria serão desbloqueados novos desafios que deverá completar e novos recursos que poderá utilizar para progredir da maneira que mais lhe convém dentro da disciplina. É introduzido o conceito de caderneta de cromos como forma de envolver e incentivar mais os alunos nas atividades de aprendizagem da programação. Os cromos bloqueados terão um sombreamento em tons de cinza e indicarão os objetivos necessários para os desbloquear. Os alunos poderão ganhar cromos e tentar completar a caderneta ao participar no sistema e ao obter bom desempenho e resultados. Após o seu desbloqueio, os cromos adotarão as suas cores originais.

A avaliação do trabalho foi realizada através de inquérito aos alunos que frequentaram esta unidade curricular anteriormente, para analisar a sua opinião em relação ao antigo e ao novo sistema ludificado. Foi também realizada uma entrevista semiestruturada a especialistas na área do Ensino da Programação para obter uma informação mais qualitativa sobre a potencialidade e impacto desde sistema, depois de implementado, na motivação e resultados dos alunos.



# Abstract

With this dissertation, it is intended to find a solution to the existing problem among the students that get into the higher education, regarding the teaching and learning of programming. Many of these students have a heterogeneous experience in programming practice, which makes the speed at which the content taught in the classes be it appropriate for only the part of the students, which may cause distress or boredom.

The goal is to attend to the needs of each student, individualizing their course in the curricular unit and increasing their motivation and effectiveness of this process. Therefore, a case study was carried out applied to Fundamentals of Programming curricular unit of the 1<sup>o</sup> year, 1<sup>o</sup> semester of MIEIC of Engineering Faculty of Porto University, where gamification elements and techniques were used to create a system that can be implemented on moodle platform. This system is regulated by a scoring and level system to foster students's progression in the discipline at their own pace and according to their skills in each moment. As the students moves forward in the content, new challenges that should be completed are unblocked so as new resources that they can utilize to progress in the way that suits them best in the unit. The concept of stickers album is introduced as a way of engaging and encourage more the students in the activities of programming learning. The locked stickers will have a shading in shades of gray and will indicate the necessary goals to unlock them. The students can collect stickers and try to complete the sticker album, participating in the system and getting good performance and results. After unlocking, the stickers will adopt their original colors.

The evaluation of the work was performed through a survey to the students that attend this curricular unit before, to analyse their opinion regarding the old and the new gamified system. It was also made a semi-Structured interview to specialists in the fields of programming teaching to get a more qualitative information on the potentiality and impact of this system, after being implemented, in the motivation and students's result.



# Agradecimentos

Quero agradecer ao professor António Coelho, meu orientador, pela sua disponibilidade, conselhos, sugestões e por me acompanhar durante todo este processo de realização desta dissertação.

Agradeço também a todos os especialistas no ensino da programação e os antigos alunos que frequentaram a unidade curricular de Fundamentos da Programação que disponibilizaram parte do seu tempo para dar a sua opinião e sugestões de melhoria ao sistema ludificado proposto nesta dissertação.

Gostaria também de agradecer a minha família e amigos que de alguma forma me apoiaram nos momentos de maior *stress* e dificuldade.

Jorge Adolfo David Castro Monteiro



# Índice

<b>1. Introdução.....</b>	<b>1</b>
1.1 Contexto/Enquadramento/Motivação .....	1
1.2 Problema e Hipóteses da Investigação.....	2
1.3 Objetivos da Investigação .....	2
1.4 Contribuições do Estudo.....	3
1.5 Metodologia de Investigação .....	3
1.6 Estrutura da Dissertação .....	4
<b>2. Estado da Arte .....</b>	<b>5</b>
2.1 <i>Gamification</i> .....	5
2.2 Ludificação do <i>Moodle</i> .....	8
2.3 Trabalhos e Estudos Relacionados .....	9
2.3.1 Codecademy .....	10
2.3.2 Outros Estudos.....	12
<b>3. Ludificação no Ensino da Programação .....</b>	<b>15</b>
3.1 Definição dos Objetivos.....	15
3.2 Delinear os Comportamentos-Alvo .....	16
3.3 Descrever os Jogadores.....	18
3.4 Ciclos de Jogo.....	19
3.5 Diversão.....	24
3.6 Ferramentas a Utilizar.....	25
3.7 Análise do sistema com a <i>framework</i> Octalsys.....	29
<b>4. Implementação do Sistema no Moodle.....</b>	<b>33</b>
4.1 Página de Fundamentos de Programação no <i>Moodle</i> .....	33
4.2 Caderneta de Cromos.....	35
4.3 Fóruns de Esclarecimento de Dúvidas .....	36
4.4 Quizzes .....	38
4.5 Partilha de <i>Links</i> .....	39
4.6 Sugestões de Exercícios.....	40
4.7 Sugestões de Projetos .....	41

4.8 Outras Atividades .....	42
<b>5. Apresentação e Discussão dos Resultados.....</b>	<b>43</b>
5.1 Questionários aos Alunos .....	43
5.2 Questionários aos Especialistas .....	47
<b>6. Conclusão e Trabalho Futuro .....</b>	<b>51</b>
6.1 Conclusão .....	51
6.2 Possíveis Aperfeiçoamentos ao Sistema Ludificado Desenvolvido .....	52
6.3 Trabalho Futuro .....	53
<b>7. Referências.....</b>	<b>55</b>
<b>8. Anexos .....</b>	<b>57</b>
ANEXO A.....	58
ANEXO B.....	62

# Lista de Figuras

Figura 1 - Tabela com lista de todas as frameworks analisadas no estudo de More et al. [8]	6
Figura 2 - <i>Badges</i> em Codecademy	11
Figura 3 - Ciclo semanal	19
Figura 4 - Ciclo Semestral	21
Figura 5 - Sistema ludificado aplicado à framework Octalsys	30
Figura 6 - Representação do plugin Level Up! à esquerda e do formato semanal da UC no centro	34
Figura 7 - Cerneta de cromos no perfil do aluno, no qual os cromos a preto e branco são aqueles que ainda não foram adquiridos e os cromos a cor são aqueles que já foram conquistados	35
Figura 8 - Lista de tópicos no fórum para esclarecimento de dúvidas	37
Figura 9 - Tópico do fórum onde é discutida a dúvida de um aluno e os outros a tentam esclarecer	38
Figura 10 - Lista de conteúdos partilhados pelos alunos numa base de dados	39
Figura 11 - Base de dados contendo uma lista de sugestões de exercícios propostos pelos alunos	40
Figura 12 - Base de dados contendo uma lista de sugestões de ideias para projetos propostos pelos alunos	41



# Lista de Tabelas

Tabela 1 - Tabela de pontos de avaliação do nível Aprendiz	22
Tabela 2 - Tabela de pontos de avaliação do nível Mágico	23
Tabela 3 - Tabela de pontos de avaliação do nível Feiticeiro	23
Tabela 4 - Tabela de Pontos de jogo	24
Tabela 5 - Resultados da primeira parte do questionário realizado aos alunos	44
Tabela 6 - Resultados da segunda parte do questionário realizado aos alunos	46
Tabela 7 - Resultados sobre a situação atual dos alunos nas unidades curriculares de programação que os especialistas lecionam	47
Tabela 8 - Resultados da opinião dos especialistas sobre o sistema ludificado	49



# Abreviaturas e Símbolos

UC	Unidade Curricular
MIEIC	Mestrado Integrado em Engenharia Informática e Computação
MM	Mestrado em Multimédia
MIEEC	Mestrado Integrado em Engenharia Electrotécnica e de Computadores
FEUP	Faculdade de Engenharia de Universidade do Porto
FPRO	Fundamentos da Programação
BD	Base de Dados
URL	<i>Uniform Resource Locator</i>



# 1. Introdução

## 1.1 Contexto/Enquadramento/Motivação

A indústria dos jogos tem vindo a aumentar de ano para ano em termos de receitas. De acordo com o último relatório da SuperData Research, no final do ano de 2016 gerou-se receitas de mais de 90 mil milhões de euros, sendo que quase metade das receitas são provenientes dos títulos para smartphones e tablets. Parte do sucesso dos jogos deve-se ao facto de estes conterem técnicas e elementos que cativam e envolvem o jogador fazendo muitas vezes com que eles se abstraia do mundo real e apenas se foque no mundo virtual proporcionado pelo jogo que está a experienciar, sentindo muitas vezes que faz parte dele e que a sua contribuição é importante naquele mundo. Elementos de *gamification* como pontos, *feedback* rápido, níveis, desafios ou *leaderboards* motivam o jogador a continuar e a atingir os objetivos propostos no jogo ou criados por ele próprio.

Por outro lado a programação é uma área do conhecimento bastante útil na criação e desenvolvimento de *software* que ajuda as pessoas no dia-a-dia, tanto em termos de comunicação, gestão, de procura de informação, de entretenimento e até mesmo na saúde ou no desporto. O *software* e os dispositivos em que é aplicado fazem cada vez mais parte do nosso quotidiano sendo a programação uma parte fundamental para a criação de *software* de qualidade que proporcione as funcionalidades necessárias para tornar a vida das pessoas mais simples.

Por este motivo é importante que os aprendizes de programação se sintam motivados e envolvidos aquando da sua aprendizagem. No entanto, esta área requer bastante esforço na iniciação da sua aprendizagem por parte dos estudantes, fazendo com que alguns deles desistam da disciplina ou apenas façam o necessário para serem aprovados na unidade curricular. Por isso, será importante encontrar maneiras de motivar os alunos com o objetivo de criar um progressão constante levando o aluno a ser cada vez mais curioso e aceitar desafios cada vez mais complexos nesta área.

O tema desta dissertação aborda questões relacionadas com a ludificação aplicada à educação, mais especificamente ao uso de elementos e técnicas de jogos para aprendizagem de programação.

Pretende-se assim neste estudo conceber um sistema gamificado aplicado à plataforma do *moodle* e obter a opinião dos alunos do 1º ano do MIEIC da Faculdade de Engenharia da Universidade do Porto que frequentaram a unidade Curricular de Fundamentos de programação, de forma a fazer com que a aprendizagem de programação se torne mais motivadora e envolvente para estes bem como dar-lhe liberdade para eles progredirem na disciplina de uma forma mais individual.

## 1.2 Problema e Hipóteses da Investigação

Ensinar aos alunos as bases de programação num único semestre pode ser desafiante para os professores. Alguns alunos conseguem acompanhar a matéria que é lecionada com mais facilidade mas, por vezes, os alunos levam tempo a aprender a pensar algoritmicamente e têm uma fraca compreensão das noções básicas de programação [5] e ficam dececionados com eles próprios. Adicionalmente, estes alunos não costumam treinar em casa [5] o que torna o processo ainda mais lento. Por outro lado, há alunos que já têm bases de programação adquiridas num curso secundário e sentem-se entediados por estar a aprender o que já tinham aprendido anteriormente, sendo o desejo destes de embarcar em projetos maiores e mais desafiantes. As diferentes velocidades de aprendizagem que os alunos apresentam podem desmotivar alguns alunos e consequentemente fazer com que percam o interesse na disciplina.

Com todas estas singularidades que os alunos apresentam na aprendizagem de programação, é essencial motivar e envolver os alunos na aprendizagem e individualizar as necessidades de cada aluno e fazer com que todos sintam que estão a ter uma progressão constante.

As hipóteses colocadas para este problema são que a motivação, empenho e envolvimentos dos alunos aumenta quando expostos a um sistema ludificado e, que a gamificação do *moodle* irá possibilitar a progressão constante na aprendizagem de programação dos alunos, independentemente do seu nível de conhecimento ou velocidade de aprendizagem.

## 1.3 Objetivos da Investigação

Os objetivos propostos para validar as hipóteses da investigação são:

- Definir, atendendo às características e necessidades individuais de cada aluno e com base na revisão de literatura estudada, um *design* gamificado para a UC de Fundamentos de programação que promova um maior envolvimento e motivação por parte de todos os alunos, possibilitando-os de trilhar o seu próprio caminho na unidade curricular.

- Apresentação gráfica de como seria o sistema depois de implementado no *moodle*.
- Conhecer a opinião dos alunos acerca do sistema utilizado anteriormente e recolher dados dos elementos que eles achariam mais interessantes de implementar para os anos seguintes. Assim será possível comparar a opinião deles com o novo sistema criado.
- Entender o potencial e o impacto que o sistema poderá ter na motivação dos alunos através de entrevistas realizadas a especialistas na área do ensino da programação.

## 1.4 Contribuições do Estudo

A contribuição deste estudo é uma proposta de um *design* para o ensino de programação com recurso à *gamification*, que individualize o percurso de cada estudante consoante as suas habilidades e que os motive e direcione para obterem uma aprendizagem mais progressiva e melhores resultados. Com esta contribuição, uma possível implementação do sistema no *moodle* e respetiva investigação de campo, poderão com mais efetividade o impacto deste sistema na motivação e desempenho dos alunos.

## 1.5 Metodologia de Investigação

Ao realizar esta dissertação foi primeiramente realizado um estudo sobre os seguintes tópicos: conceito de *gamification*, *frameworks* utilizadas para implementação da ludificação e *gamification* aplicada à plataforma *moodle*. Foram também analisados trabalhos e estudos relacionados com *gamification* no ensino, que deram uma visão mais ampla das possibilidades de conceção e implementação de um novo sistema.

Com base no estudo realizado foi gerado um sistema ludificado seguindo os 6 passos de Werbach e, com a *framework* Octalsys, o sistema foi analisado consoante algumas variáveis como o significado, a perda, a influência social, realização, fortalecimento da criatividade e *feedback*, escassez, imprevisibilidade e posse ou propriedade.

Esta investigação é quantitativa e qualitativa e a amostra serão os alunos do 1º ano do MIEIC da FEUP que tenham frequentado a UC de Fundamentos de Programação e que aceitaram participar neste estudo.

O primeiro método de recolha de dados utilizado nesta investigação foi um questionário que tem como intuito conhecer a opinião dos alunos sobre o sistema utilizado anteriormente em Fundamentos de Programação e também a intenção de receber *feedback* sobre como melhorar o

novo sistema criado, de forma a motivar mais os alunos na aprendizagem de programação e potenciar a individualização dos seus percursos (os alunos poderem ter mais poder de escolha) dentro da unidade curricular. Os dados recolhidos através do questionário foram analisados e comparados com o sistema de ludificação criado para perceber se a opinião dos antigos alunos coincidia com a proposta de *gamification* concebida.

O outro instrumento utilizado para recolha de dados foi a entrevista. Esta foi aplicada a especialistas na área do ensino de programação e terá um carácter mais qualitativo onde se pretendeu obter um outro ponto de vista sobre a potencialidade do sistema proposto.

## **1.6 Estrutura da Dissertação**

Esta dissertação contém 7 capítulos. No capítulo 1, é feita uma introdução e contextualização do tema a estudar, tal como o seu problema, hipóteses, objetivos, contribuições e metodologia da investigação. No capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados. No capítulo 3 é descrito o sistema ludificado com base nos 6 passos de Werbach bem como a sua validação através da *framework* Octalysis. No capítulo 4, será explicada a forma de implementar o sistema no *moodle* e serão apresentadas representações gráficas de como seria o sistema depois de implementado. O capítulo 5 expõe os resultados e a sua discussão. No capítulo 6, reflete-se sobre as conclusões, limitações do estudo e trabalhos futuros e no capítulo 7 encontram-se as referências bibliográficas.

## 2. Estado da Arte

Neste capítulo será feita uma síntese sobre o conceito de *gamification*, algumas *frameworks* utilizadas para o *design* de gamificação, aspetos psicológicos a levar em conta como motivação intrínseca e a motivação extrínseca e também algumas opções de ludificação da plataforma *moodle*. São também apresentados estudos relacionados e um caso prático de uma plataforma *online* de aprendizagem de programação com recurso à *gamification*.

### 2.1 Gamification

Em Deterding et al. [3] é explicado que o termo *gamification* foi usado a primeira vez em 2008 mas que apenas na segunda metade do ano de 2010 é que o termo começou a ser amplamente usado por especialistas da indústria e em conferências. O termo flutua entre dois conceitos relacionados. O primeiro é a adoção e institucionalização dos jogos no nosso quotidiano e o segundo é que os jogos têm o propósito de entreter e podem motivar o utilizador a envolver-se neles, então é possível que os elementos de jogo sejam capazes de fazer com que outros serviços e produtos sejam mais agradáveis e envolventes.

Quando ao conceito de *gamification*, eles definem como o uso de elementos de *design* de jogo em contextos fora do ambiente dos jogos. Por outro lado, Houtari e Hamari passam a ideia de que *gamification* foi definida como um processo que melhora serviços com estímulos motivacionais de forma criar experiências parecidas com a dos jogos e a mudar comportamentos [4]. Ainda na revisão de literatura de Hamari et al. [4] transmitem que a *gamification* pode ser vista como tendo 3 partes: os elementos motivacionais implementados, os resultados psicológicos obtidos, e os resultados comportamentais que derivam dos resultados psicológicos.

Segundo a revisão de literatura de Mora et al. [8] muitos são os processos já existente para um *design* gamificado de sucesso. Dezoito processos foram incluídos no estudo, sendo que 10 deles são genéricos e 8 são específicos para negócios. Na Figura 1 é possível verificar a lista de todas as *frameworks* analisadas no estudo, e compará-las com as seguintes categorias: económica (viabilidade/ risco/ retorno do investimento e *stakeholders*), lógica (ciclo, fim de jogo/ vitória épica, integração e regras), analítica (métricas), psicologia (ética) e interação (UI/ UX e

tecnologia). As células da tabela informam se o item é explícito, implícito ou não definido na definição da *framework*.

CATEGORIES FEATURES	Economic		Logic			Measurement		Psychology	Interaction	
	Viability	Stakeholders	Loop	Endgame	On-boarding	Rules	Metrics	Ethics	UI/UX	Technology
A Framework for Success. Di Tommaso (2011) [21]	U	E	I	E	U	I	E	U	U	U
Six steps to Gamification. Werbach and Hunter (2012) [23]	I	I	E	E	E	E	E	I	E	E
Gamification Framework. Marczewsky (2012) [24]	E	E	E	U	E	U	E	U	U	E
Gamification Design Process. Marache-Francisco and Brangie (2013) [26]	U	U	E	U	U	E	U	E	E	U
Steps to Gamification. De Paz (2013) [27]	I	E	E	U	E	E	E	U	I	E
Robinson and Bellotti taxonomy (2013) [28]	U	U	U	U	U	E	U	U	U	U
Francisco-Aparicio et al. framework (2013) [29]	U	U	U	U	U	E	U	E	U	E
A moral framework for taking responsibility. Versteeg (2013) [30]	I	E	I	I	U	U	U	E	I	E
Octalysis: Complete Gamification Framework. Chou (2013) [33]	U	U	U	E	E	U	E	U	U	U
A Framework for Sustainable Gamification Impact. AlMarshedi (2015) [34]	U	U	E	E	E	I	I	U	I	U
Player Centered Design Methodology. J. Kumar (2013) [37]	U	U	E	U	E	E	E	E	I	E
Role-Motivation-Interaction Framework. Gears (2013) [39]	E	E	U	U	U	E	U	U	U	U
Gamification Framework model. Jacobs (2013) [42]	I	E	E	U	U	I	E	U	I	I
A framework for gamification suited for marketing. Julius and Salo (2013) [16]	I	I	E	E	E	E	E	I	E	E
Theoretical Model for Gamification in Workplace IS context. Li (2014) [43]	U	U	U	U	U	U	U	U	E	E

**Figura 1 - Tabela com lista de todas as frameworks analisadas no estudo de More et al. [8]**

Sendo este estudo mais focado em *gamification* na educação, é mais revelante referir os processos mais genéricos e não tanto os que são focados especificamente para negócios.

Di Tommaso (2011) propõe uma *framework* para o sucesso contendo os seguintes passos:

1. Porquê ludificar? – Respondendo a perguntas relacionadas com o público-alvo como “qual a principal razão para ludificar o produto/ serviço?”, como isso pode beneficiar o utilizador e se ele vai apreciar; e também a questões como quais são os objetivos e quais são as ações que o utilizador deve desempenhar para atingir esses objetivos.
2. Identificar o perfil do jogador e quais os seus condutores de motivação.
3. Definir as metas e os objetivos, que consiste em esclarecer o objetivo final, como “aprender uma habilidade nova” e enumerar as metas (subobjetivos/ desafios) para atingir o objetivo mais abrangente.
4. Descrever as habilidades, acompanhá-las e medi-las, ou seja, listas as habilidades (físicas, mentais e sociais) necessárias para atingir determinado objetivo e pensar numa maneira de as rastrear e medir.
5. Definir os elementos que irão manter o interesse do jogador como competição, tempo limite, escassez, puzzles, novidade, níveis, provação ou pressão social, trabalho de equipa, economia interna (moedas ou outro tipo de valorização dentro do sistema) e *power-ups*.
6. Resultados desejados, através de feedback, prémios e definindo o estado de vitória épica.

7. Testar o sistema e poli-lo, sendo este um processo iterativo.

Contudo, o processo mais conhecido é o de Werbach e Hunter (2012), onde definem 6 passos para a *gamification*, sendo estes:

1. Definição dos objetivos, mencionando o porquê da utilização da *gamification*, como esta pode beneficiar os seus intervenientes e quais os resultados esperados.
2. Delinear o comportamento do público-alvo, salientando quais os comportamentos desejáveis e expectáveis dos jogadores e encontrar métricas para medi-los.
3. Descrever os jogadores, caracterizando-os demograficamente, psicograficamente, pelo seu estilo de jogo, podendo mesmo usar como referência a taxonomia de Richard Bartle (1996), que distingue 4 tipos de jogador (*killers, achievers, explorers e socializers*) ou a matriz de ação social de Kim (2010) inspirada neste último que distingue-os como competidores, criadores, exploradores e colaboradores.
4. Elaborar os ciclos do jogo, aplicando o conceito de motivação, ação e feedback ao sistema de jogo para que o jogador esteja sempre envolvido e motivado a progredir.
5. Diversão – garantir que o jogador se diverte mesmo em contextos fora dos jogos, é importante para o seu foco na atividade que está a desempenhar. Uma estratégia para analisar este passo é considerar como é que o sistema gamificado funcionaria sem os prémios extrínsecos. Outra estratégia para refinar a diversão seria utilizar as 4 chaves para a diversão propostas por Lazzaro (2004), que se divide em diversão desafiante (associado a desafios complexos), diversão simples (associado a curiosidade), diversão séria (associado à excitação e relaxamento) e a diversão social (associado à interação através de competição ou trabalho de equipa).
6. Utilizar as ferramentas apropriadas, ou seja, definir as técnicas e elementos de *design* que mais se adequam aos objetivos propostos.

Por fim, a *framework* Octalsys criada por Yu-kai Chou nomeia 8 fatores-chave da gamificação e aplica-os a um octógono, sendo cada um dos lados deste representado por um fator.

[9] Os oito fatores são:

1. Significado épico e chamamento – acontece quando o jogador sente que tem um propósito maior, ou que foi escolhido para desempenhar algo importante (ex: ajudar a criar algo que toda a comunidade possa beneficiar, como a Wikipédia).
2. Desenvolvimento e realização – relacionado com progressão, desenvolvimento de competências e superação de desafios.
3. Fortalecimento da criatividade e *feedback* – acontece quando os jogadores estão envolvidos no processo criativo e estão sempre a resolver problemas e tentar diferentes combinações. Expressam a sua criatividade, observam os resultados e respondem a este feedback.

4. Propriedade e Posse – acontece quando o jogador está motivado e sente que possui algo. Ele quer melhorar o que já possui e possuir ainda mais. Tem como principal objetivo a acumulação de riqueza, lidando com moedas e bens virtuais dentro do sistema.
5. Influência social e relacionamentos – incorpora todos os elementos sociais como mentoria, aceitação, respostas sociais, camaradagem, bem como competição e inveja.
6. Escassez e impaciência – acontece quando se quer algo que não se pode ou é complicado de obter, ou quando não se pode ter as coisas instantaneamente, tendo de esperar para o conseguir obter. Esta impossibilidade faz com que o jogador esteja motivado a pensar no assunto durante uma quantidade de tempo considerável (ex: O Facebook era limitado a Harvard e só com o tempo foi expandindo até todo o mundo ter a possibilidade de o utilizar).
7. Imprevisibilidade e curiosidade – está relacionado com o desconhecido e com aquilo que pode acontecer no futuro. Como não se sabe o que esperar, o cérebro fica entretido a tentar descobrir.
8. Perda e evitação – quando o jogador tenta evitar que algo de mal aconteça ou quando tem medo de perder uma oportunidade caso não aja rapidamente.

Para além dos fatores-chave apresentados, o octógono é também dividido entre o cérebro esquerdo (parte esquerda do octógono) e o cérebro direito (parte direita do octógono). A parte esquerda está relacionada com lógica, cálculos e posse ou propriedade e é baseada na motivação extrínseca (ex: alcançar objetivo, obter um bem). A parte direita está relacionada com a criatividade, autoexpressão e aspetos sociais, sendo baseada na motivação intrínseca (ex: uso da criatividade, sair com amigos e imprevisibilidade).

Faz-se também a distinção entre a parte de cima e a parte de baixo do octógono, estando a primeira normalmente associada a conotações positivas e a segunda a conotações negativas. No entanto, a utilização dos fatores da parte de baixo do octógono não significa obrigatoriamente algo mau (ex: evitar desligar o despertador e voltar a adormecer).

## **2.2 Ludificação do Moodle**

O Moodle é um sistema de gestão de aprendizagem, uma plataforma virtual que permite que os alunos e professores interajam para o ensinamento e aprendizagem de um ou vários cursos ao mesmo tempo. Nesta plataforma é possível criar cursos e adicionar recursos como vídeos, imagens, documentos bases de dados, fóruns, entre outros. Por vezes, no entanto, os alunos precisam de uma motivação extra para se manterem concentrados nas tarefas da disciplina e constantes no curso que estão a frequentar. A pensar nisso, a plataforma *moodle* deu a

possibilidade aos professores de implementar técnicas para manter os estudantes envolvidos na matéria através de *gamification*.

Uma dessas técnicas é a possibilidade de atribuir medalhas aos alunos [10]. Esta função está disponível a partir da versão 2.5 e permite ao professor atribuir medalhas aos alunos de 3 maneiras distintas: manualmente (as medalhas são criadas e o professor pode escolher manualmente e a qualquer altura a quem as irá oferecer), após a finalização de um curso ou após a finalização de uma certa atividade dentro de um curso (após a conclusão, a medalha é adicionada automaticamente ao perfil do aluno). Os *badges* têm um nome, uma descrição, uma imagem e refere o nome da pessoa que o atribuiu.

Outra das técnicas que ajudam na aplicação de *gamification* no *moodle* são as restrições de acesso [11]. Com esta opção, é possível impedir a progressão e desbloqueio do curso do aluno consoante algumas condições e após estas estarem satisfeitas, uma nova atividade ou recurso aparece na página do curso para o aluno. É possível restringir o acesso a novos conteúdos através da data (só aparecem a partir de uma determinada data definida), através das notas (só podendo visualizar nos conteúdos se tiver uma nota superior a um determinado valor), através dos dados do perfil do aluno ou através de um grupo de restrição.

Para além destas funcionalidades inatas do *moodle*, é também possível adicionar *plugins* que são criados especificamente para oferecer à plataforma e aos professores uma maneira de introduzir elementos de ludificação nos seus cursos. Um dos principais e mais usados *plugins* relacionados com *gamification* aplicada ao *moodle*, é o “*level up!*”. Este *plugin* permite definir o número de níveis que se quer criar e quantos pontos de experiência é necessário obter em cada um deles para passar para o próximo nível; é possível definir os pontos de experiência atribuídos a cada tarefa; deteta que um aluno realizou determinada tarefa e atribui automaticamente os pontos; o professor pode obter um relatório dos níveis e da pontuação geral de todos os alunos; os alunos recebem uma mensagem de felicitação ao subirem de nível; um bloco aparece na página do curso a indicar o nível e os pontos de experiência obtidos e também é possível visualizar o *ranking* dos alunos. Para além disto existe também uma página onde é possível visualizar todos os níveis e as suas descrições. A aparência do nível também pode ser alterada pelo professor ao fazer o *upload* de uma nova imagem que represente cada nível.

## 2.3 Trabalhos e Estudos Relacionados

Nesta secção será apresentado um caso prático de uma plataforma *online* de ensino de programação com recurso à gamificação bem como alguns estudos que se focam na *gamification* em educação.

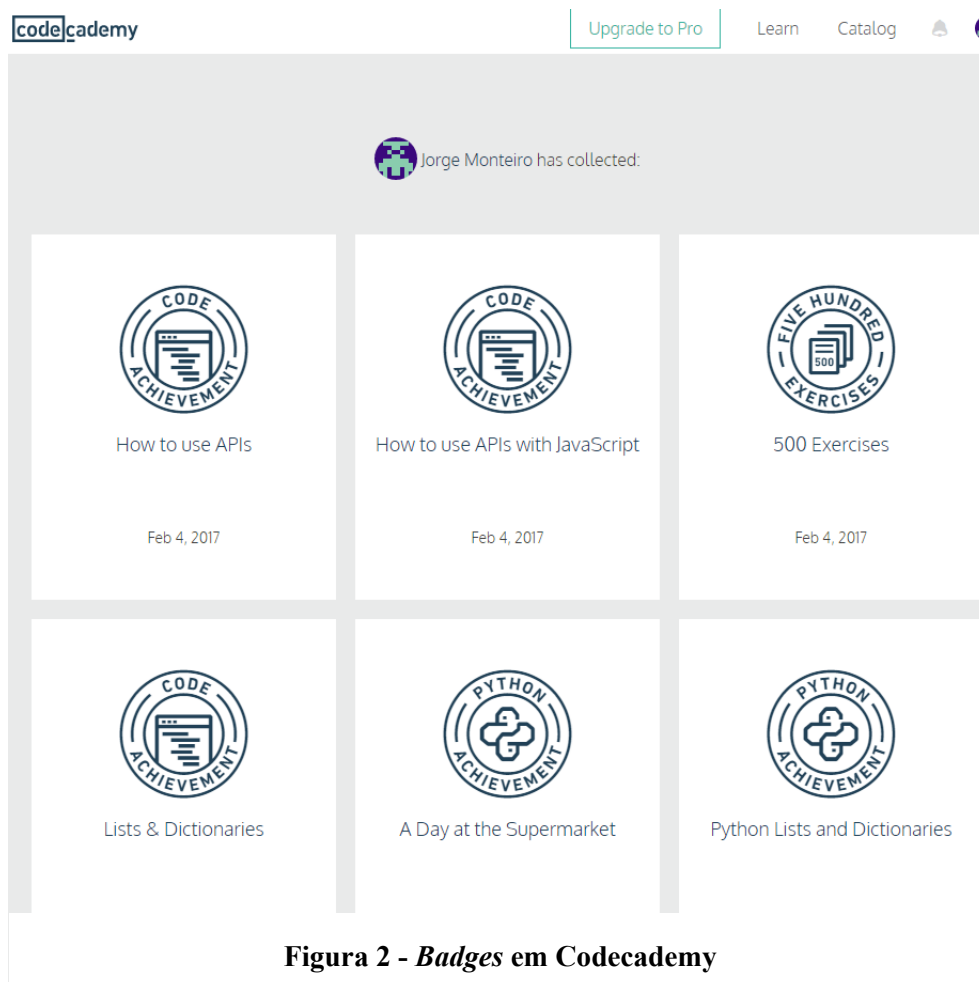
### 2.3.1 Codecademy

Codecademy [1] é uma plataforma de aprendizagem de programação que utiliza elementos de *gamification* para motivar os seus utilizadores a terem uma aprendizagem suave e progressiva da linguagem ou cursos que escolheram aprender ou realizar. As linguagens existentes no *site* que se podem aprender são: HTML, CSS, javascript, java, Sass, python, jQuery, ruby e PHP. É também possível aprender linha de comandos, SQL e como trabalhar com o Git e com APIs.

A plataforma é apropriada a utilizadores que saibam pouco ou nada sobre programação ou sobre uma linguagem em específico. É grátis, no entanto tem de se pagar mensalidade para aceder a certos conteúdos que estão bloqueados.

As características e funcionalidades deste *website* são:

- **Barras de progresso** - Em cada curso que é inicializado pelo utilizador existe uma barra de progressão que lhe permite acompanhar o seu progresso e quando resta para ele continuar o curso. A métrica utilizada é a percentagem.
- **Badges** - Os *badges* vão-se adquirindo à medida que se vai realizando os exercícios dos diversos cursos disponíveis. Alguns exemplos de *badges* seriam “500 Exercises” quando o utilizador finaliza-se com sucesso 500 exercícios ou “python achievement - functions”, quando o utilizador completa-se e entende-se como utilizar as funções em Python. Existem também diferentes *badges* que podem ser adquiridos consoante a participação no fórum do Codecademy.
- **Pontos** - É contabilizado o número total de pontos adquiridos no *website*. Os pontos são adquiridos ao realizar exercícios e possibilitam apenas ao utilizador o acompanhamento do seu progresso como aprendiz.
- **Fórum** - O fórum permite o relacionamento e entreaajuda dos utilizadores, possibilitando-os de pesquisar e fazer perguntas, responder a estas, gostar de uma publicação, partilha-la nas redes sociais ou através de *links* e guardar uma publicação para saber que já foi lida ou para poder lê-la posteriormente.
- **Feedback** - Após a realização e submissão de um exercício, o utilizador recebe um *feedback* positivo, com a cor verde, caso o exercício tenha sido solucionado de forma correta. Caso contrário, aparece um aviso de cor vermelha que transmite que algo se encontra errado e onde irá ser necessário fazer a correção.



**Figura 2 - Badges em Codecademy**

- **Conteúdo desbloqueável** - Dentro de cada secção de curso existente, existe conteúdo que pode ser desbloqueável, como *quizes* de escolha múltipla ou projetos, mas apenas se o utilizador fizer o *upgrade* da sua conta com dinheiro real.
- **Lições** - As lições são pequenas porções de texto que explicam ao aprendiz a matéria que está a ser abordada. Aparecem do lado esquerdo do ecrã para que o utilizador possa ler primeiro antes de tentar resolver o desafio proposto. As lições também costumam ter exemplos para uma melhor compreensão.
- **Desafios** - Os exercícios dados no final de cada lição são desafios que o utilizador terá de superar de forma a progredir no curso, aumentar o seu conhecimento e ganhar *badges* como recompensa pelo seu esforço.

### 2.3.2 Outros Estudos

No estudo de Regalado et al. [6], intitulado “A Systematic Mapping on the use of Gamification in Virtual Learning Environments”, partem do principio que quando são usadas plataformas de aprendizagem à distância, normalmente não há uma preocupação acerca da qualidade da educação que está a ser passada, isto porque, em parte, há poucos diplomados desta modalidade. Depois de implementada a *gamification* nessas plataformas, deve haver uma preocupação para entender se está a afetar positivamente a aprendizagem dos alunos.

É então feito um mapeamento para identificar e classificar os ambientes de aprendizagem *online* gamificados feitos em estudos anteriores, e ajudar todos os trabalhos futuros relacionados com ensinamento ou *gamification*.

Foram primeiramente pesquisados artigos em motores de busca como ACM Digital Library, IEEEExplore e Science Direct relacionados com "gamification", "lms" e "e-learning". Posteriormente, esses artigos passaram por um processo de inclusão ou exclusão, sendo que dos 149 artigos iniciais acabaram por restar 17 artigos que seriam analisados na integra para o presente estudo.

As 5 questões de investigação pretendiam obter as seguintes respostas: quais as técnicas de *gamification* foram usadas, o nível de escolaridade onde a plataforma com *gamification* é testada, a área do conhecimento a que é aplicada, em que plataforma foi testada e qual a linguagem de programação foi utilizada para desenvolver os sistemas de aprendizagem gamificado.

Os resultados mostram que a maior parte das plataformas referidas nos estudos analisados utilizam *scoreboards* e badges ou *achievements* como técnica de ludificação, o nível de escolaridade onde a *gamification* é mais utilizada é no ensino universitário, e em relação à área de conhecimento é mais utilizada no ensino da área das tecnologias da informação. As plataformas onde foram aplicados os elementos do jogo são maioritariamente plataformas de desenvolvimento próprio, vindo em segundo lugar o desenvolvimento para a plataforma do *moodle*. Dos estudos que especificaram a linguagem de programação utilizada para implementar o sistema ludificado, o PHP foi o mais utilizado, aparecendo também linguagens como python, javascript e java.

Das quatro plataformas que utilizaram o *moodle*, todas usaram *scoreboards*, metade tem como área de conhecimento as tecnologias da informação e todas elas usam PHP como linguagem de programação.

Tatyane Silva et al. [5] no seu estudo “Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando *gamification*”, enumera as dificuldades que os alunos mais encontram na aprendizagem de programação: baixa compreensão das noções básicas, como estruturas de controlo, criação de algoritmos que resolvam problemas concretos, aprender a pensar algoritmicamente, têm uma base matemática fraca, uso de linguagens de programação com sintaxes rebuscadas para estudantes em início de curso, a falta de entendimento dos enunciados, exercícios dissociados de problemas “reais” e pouca disponibilidade à parte das aulas para

estudar. Com o objetivo de motivar, envolver e influenciar positivamente a mudança de comportamento do aluno em relação à programação para obtenção de um melhor desempenho e resultados, propõem então, utilizando *gamification*, a criação de um modelo conceptual que auxilia os estudantes no desenvolvimento de competências e habilidades para aprender programação. O modelo é composto por 3 partes: perfil dos jogadores (*achievers*, *killers*, *socializers* e *explorers*), técnicas de "engajamento" (desafios e missões, rankings e *leaderboards*, Interação social, *achievements* ou badges) e elementos de jogos (pontuação, narrativa e feedback). A plataforma cod[edu], desenvolvida através do modelo, usa as interações do utilizador para classificar o perfil de jogador do aluno e lança técnicas de envolvimento baseadas no seu perfil.

Os resultados obtidos mostraram que os indicadores de envolvimento dos alunos apresentaram mudanças positivas após os testes com o modelo criado, podendo isto dever-se ao uso da *gamification*. Os alunos com maior envolvimento foram influenciados de forma positiva quando tiveram o perfil de jogador classificados pelo cod[edu].

Na entrevista feita aos professores após os testes, conclui-se que eles avaliam a plataforma como um modelo que auxilia na envolvimento dos estudantes, ajudando também no processo de aprendizagem dos alunos na disciplina de programação.

Já o estudo de Swacha e Baszuro [7] propõe um *design* inovador de uma plataforma *e-learning* que faz uso da *gamification* com o objetivo de aumentar o envolvimento dos alunos na aprendizagem, premiando o trabalho individual mas também estimulando o estudante tanto em competição como em trabalho de equipa. É uma plataforma independente, não sendo uma extensão de uma plataforma *e-learning* já existente.

A plataforma apoia o ensino da programação e requer a participação ativa do professor. Possibilita este de distribuir material de aprendizagem aos alunos, acompanhar a sua progressão e comunicar com estudantes individualmente e em grupo.

Emprega mecânicas como progressão através de níveis e pontos, *achivements* como reconhecimento pelo seu trabalho, colaboração na resolução de exercícios, matéria e informação teórica desbloqueada em cascata, tempo limite para realização de desafios, concursos ou *quests*, descoberta de elementos do curso pelos próprios estudante, aversão à perda premiando a prática regular, surgimento de novos exercícios à medida que o estudante vai progredindo e encoraja também a participação no fórum e na entreajuda dos alunos.

O estudo, no entanto, não valida a plataforma através de testes, não sendo possível avaliar os resultados que esta plataforma poderia trazer em termos de motivação e envolvimento dos alunos.



# 3. Ludificação no Ensino da Programação

Neste capítulo irá ser desenvolvido o sistema ludificado aplicado à unidade curricular de FPRO. Para a criação do sistema irão ser utilizadas duas *frameworks* descritas no capítulo anterior: os 6 passos de Werbach (pontos 3.1 a 3.6) e o Octalsys (ponto 3.7).

## 3.1 Definição dos Objetivos

O primeiro passo é definir o que se pretende alcançar com a criação deste sistema de ludificação, então deve-se enumerar os objetivos que se pretendem atingir.

### **Incentivar os alunos a praticarem programação semanalmente**

Para interiorizar e aprender programação deve-se praticar de uma forma contínua, mantendo objetivos de curto prazo (1 ou 2 semanas). A escrita frequente de código para solucionar exercícios ou para desenvolvimento de um projeto, irá ajudar os alunos a adquirirem as competências básicas para o desenvolvimento de programas de baixa/média complexidade. No entanto, muitos alunos apenas praticam quando são motivados por fatores extrínsecos (ex: fazer exercícios ou testes para passar à unidade curricular).

Pretende-se criar um conjunto de atividades que promovam um estudo mais regular.

### **Individualizar os percursos de cada aluno consoante o nível das suas competências**

Nas unidades curriculares introdutórias de uma licenciatura / mestrado integrado, que envolvam aprendizagem de programação, verifica-se que os estudantes têm, de base, competências diferentes: uns já têm experiência anterior com programação, com maior ou menor

proficiência, e outros não têm qualquer prática anterior de programação. Verifica-se assim que cada estudante necessita de progredir ao seu próprio ritmo.

Tendo esse elemento em consideração, pretende-se possibilitar que cada estudante possa traçar o seu próprio caminho na unidade curricular consoante o seu nível de competências.

### **Motivar e tornar a aprendizagem de programação mais divertida para os estudantes**

Um outro elemento importante para ter sucesso ao programar é a existência de diversão. A maior parte dos profissionais com sucesso nas suas respetivas áreas são apaixonados por aquilo que fazem, e parte responsável dessa paixão deve-se ao facto de a experiência ser divertida.

O objetivo de criar elementos de diversão é fazer com que no final os alunos aprendam programação por motivadores intrínsecos e não extrínsecos. O aluno deve aprender porque gosta, porque a área o fascina, porque pode ajudar pessoas com essa competência que está a aprender. Deve priorizar a recompensa e realização interna em vez da recompensa externa, como por exemplo o dinheiro que pode vir a ganhar.

Alguns alunos realizam exercícios porque se sentem obrigados a fazê-lo devido às regras de avaliação e não por opção própria.

Pretendem-se criar componentes (caderneta de cromos, exercícios mais estimulantes, ajuda entre alunos) que forneçam mais algum divertimento para motivar o estudante a não desistir à primeira dificuldade e ter vontade de progredir.

## **3.2 Delinear os Comportamentos-Alvo**

O segundo passo para a criação do sistema, é definir os comportamentos que os alunos devem adotar de forma a atingir os objetivos definidos no primeiro passo.

- **Resolver e submeter os exercícios de autoavaliação regularmente** – saber as bases teóricas de programação é importante, mas a prática é aquilo que leva os alunos a realmente aprender e melhorar as suas capacidades de programação. Por esta razão é importante resolver estes exercícios de autoavaliação com regularidade para ter feedback sobre se o esforço despendido na aprendizagem é suficiente.

- **Participar ativamente no fórum** – Por vezes, os alunos têm dúvidas específicas sobre algum exercícios ou sobre alguma parte da matéria e nem sempre o professor está disponível para responder de imediato. Com este esquema de perguntas e respostas pretende-se que os alunos interajam entre si e se consigam ajudar uns aos outros, esclarecendo dúvidas que possam surgir.

- **Ler os módulos do livro de estudo antes da aula respetiva** – Ter noções e conhecimentos prévios antes das respetivas aulas faz com que o aluno possa trazer dúvidas para as aulas e discutir os tópicos com mais profundidade com os outros alunos e professores, sendo assim, ler os módulos do *e-book* antes das aulas faz com que estas sejam mais produtivas do que se simplesmente o professor tivesse de lecionar a matéria na aula.

- **Responder a quizzes que promovam a validação dos conhecimentos mais teóricos** – Enquanto ler a matéria pode ser útil, responder a *quizzes* pode ser uma maneira mais rápida de aprendizagem pois requer que o aluno pense sobre qual será a resposta certa às perguntas apresentadas e recebe feedback imediato, sabendo qual a resposta certa e quais as erradas. O aluno retém a matéria mais rápido no cérebro pois esforça-se mais para responder a uma pergunta do que simplesmente a ler os apontamentos. Serve também como forma de avaliação dos conhecimentos teóricos adquiridos antes e durante as aulas.

- **Sugerir propostas de projetos para os estudantes mais motivados** – A maior parte dos alunos tem uma ou várias ideias sobre projetos que gostariam de ver implementados. Este espaço serve para que todos os alunos tenham a possibilidade de partilhar ideias de jogos ou programas que possam ser desenvolvidos pelos alunos mais motivados e envolvidos na prática de programação e, caso haja uma grande quantidade de ideias, elas poderão ser acumuladas e propostas para os alunos que irão frequentar os anos seguintes da unidade curricular. A ideia é, não só criar uma lista de possíveis projetos académicos, de onde os alunos com melhor desempenho podem escolher um deles e desenvolver, mas também promover a criatividade e iniciativa dos alunos na geração de ideias.

- **Sugerir exercícios** – Seguindo a mesma linha do tópico anterior, neste espaço os alunos poderão sugerir exercícios para possível resolução em casa ou na sala de aula. Para além de eles arranjam maneiras de resolver problemas, é também importante eles pensarem em problemas que possam ou gostariam de resolver. De certa forma isto também ajuda o professor a adquirir novos exercícios para apresentar aos novos alunos dos anos seguintes.

- **Partilhar leituras / narrativas (links)** – Por vezes encontra-se na internet alguma informação interessante e adicional que não é transmitida em sala de aulas. Sejam vídeos, infografias ou artigos, tudo isto contribui para a curiosidade e o enriquecimento da cultura dos alunos acerca de programação.

- **Realizar os trabalhos práticos e teste, tendo a possibilidade de recuperar de falhas** – Estes são os momentos de avaliação em que os alunos testam os conhecimentos que adquiriram, através de trabalhos práticos (com correção automática) e de um teste teórico.

- **Ser assíduo e pontual** – frequentar todas as aulas e chegar sempre a tempo demonstra interesse por parte do aluno em aprender a matéria lecionada.

- **Desafiar alguns estudantes a resolver problemas mais complexos (torneios)** - com o objetivo de trazer alguma competitividade, mas sem por em jogo uma tabela de classificações que possa desmotivar os alunos, estes poderão desafiar-se entre si em torneios após a finalização de cada módulo lecionado. Estes torneios consistem na solução de desafios mais complexos

propostos pelo professor (ou pelos alunos). Todos começam o desafio ao mesmo tempo e aquele que conseguir solucionar o problema no menor tempo possível ganha uma medalha de vencedor e alguns pontos. Aquele que solucionar o problema mas não tenha sido rápido o suficiente para vencer, também ganhará uma medalha por ter completado o desafio proposto no torneio.

### 3.3 Descrever os Jogadores

Saber sobre os diferentes tipos de alunos que chegam todos os anos ao ensino superior é importante pois a partir dessa análise é possível entender melhor as suas personalidades e maneira como eles se irão comportar durante o decorrer da disciplina e quais as ferramentas que irão ser mais uteis no sistema para cada um. Com isso em mente foram criadas *personas* com base nos arquétipos de Richard Bartle, que define quatro tipos de jogador: *killers* (normalmente focados em competir e ganhar a outros jogadores), *achievers* (orientados para atingir objetivos), *explorers* (que gostam de explorar e descobrir o desconhecido) e os *socializers* (que gostam de interagir com os outros e fazer amigos).

**Daniel** – É um rapaz determinado e esforçado. Em pequeno praticou natação e atletismo, vindo a tornar-se atleta de alta competição neste último. Está por isso habituado à competitividade. Daniel inscreveu-se no curso de informática pois no futuro quer ter autonomia e competências para desenvolver as suas aplicações desportivas que lhe irão ajudar a melhorar o seu rendimento.

**Ricardo** – O ricardo é uma pessoa disciplinada e orientada para realização de objetivos. No futuro ele quer ser um empresário e criar usa empresa de desenvolvimento de *software* de sucesso que ajude milhões de pessoas a obter mais comodidade e bem-estar.

**Patrícia** – É uma pessoa curiosa e que fica facilmente intrigada com diversos assuntos. Gosta de explorar sobre diferentes assuntos na internet, através de artigos, vídeos ou outros meios onde possa obter as informações que pretende. Sempre gostou de jogos, então o seu objetivo é um dia ser game designer numa grande produtora de jogos. Entrou num curso de informática para que possa tirar as suas ideias do papel e coloca-las em prática. Patrícia é o tipo de pessoa que sabe um pouco de tudo e tem uma visão mais generalista.

**Rita** – É uma pessoa descontraída e bem-disposta. Gosta de socializar e ir sair à noite com os amigos, mas também consegue ser empenhada e assumir as suas responsabilidades. Faz voluntariado com animais, tentando angariar ração para os alimentar, promovendo a adoção de animais abandonados e levando-os ao veterinário quando necessário. No futuro ela que

desenvolver máquinas ou dispositivos que ajudem na prevenção e tratamento de doenças dos animais.

### 3.4 Ciclos de Jogo

Neste sistema gamificado existirá um ciclo em que todas as semanas, desde o início até ao final do semestre, irá ocorrer o mesmo tipo de procedimentos e tarefas. Algumas delas, os alunos terão obrigatoriamente de completar, enquanto outras serão tarefas opcionais. Na Figura 3 podemos ver a ilustração do ciclo semanal do sistema gamificado.

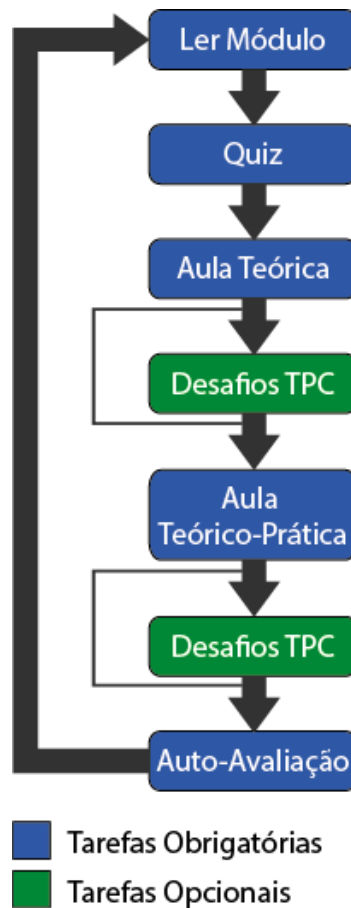
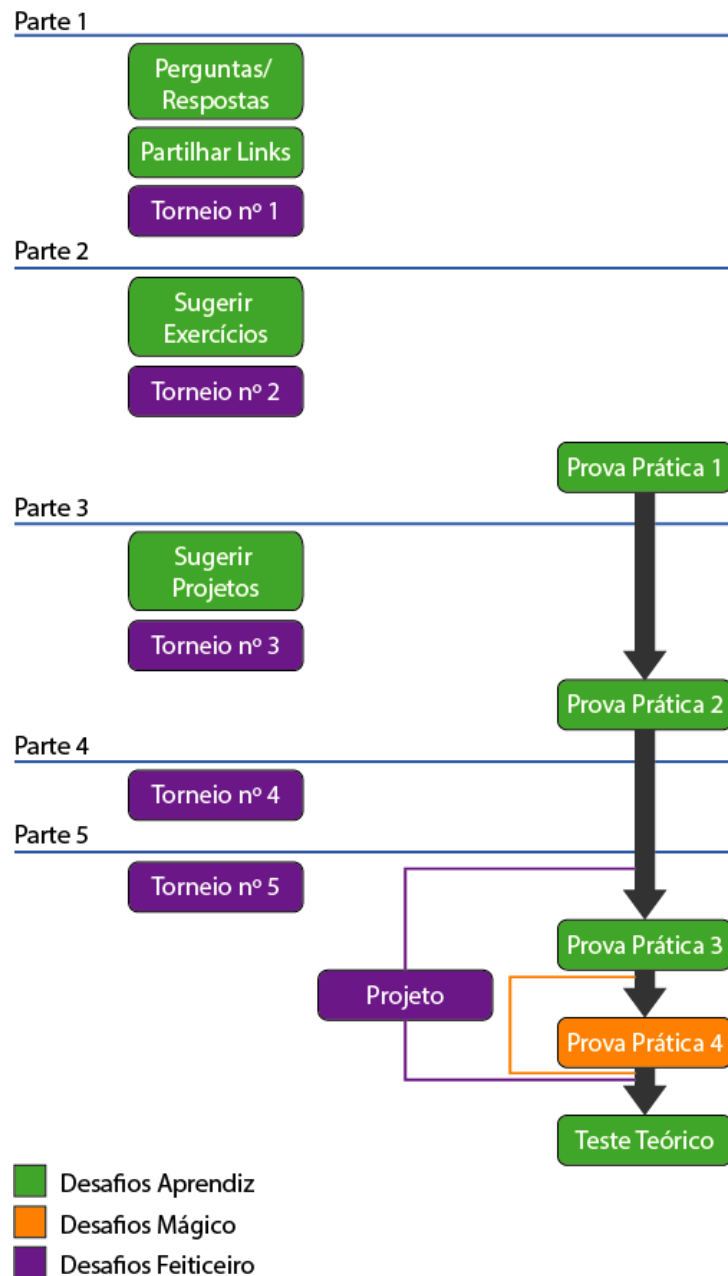


Figura 3 - Ciclo semanal

O aluno terá inicialmente de ler o módulo ou os diapositivos necessários e estudar a matéria de antemão, antes a dita aula sobre essa mesma matéria. Irá então validar os conhecimentos teóricos adquiridos, através de um *quiz* com perguntas de escolha múltipla. O aluno poderá fazer esse *quiz* o número de vezes que achar adequado. Após o *quiz*, o aluno terá a aula teórica onde poderá discutir os temas estudados, com o professor e os outros alunos, expor dúvidas que tenham surgido. O professor irá propor alguns desafios (TPC) adicionais para resolver em casa, sendo estes opcionais e não obrigatórios. Segue-se então a aula teórico-prática onde se pode trabalhar mais a parte prática da programação e tirar mais algumas dúvidas que tenham surgido. Após a aula serão outra vez enviados TPC opcionais e por fim será realizada uma autoavaliação obrigatória para o aluno verificar se adquiriu corretamente os conteúdos lecionados da semana.

Para além deste ciclo semanal existe também o ciclo semestral (Figura 4) que ilustra o percurso que o aluno poderá percorrer na disciplina durante o semestre, mostrando principalmente quando surgem as provas práticas/ projeto e o teste teórico, mas também em que momentos surgem os torneios e os outros elementos que o sistema gamificado proporciona.

A primeira prova prática será realizada apenas após a matéria da parte 1 e 2 ser lecionada, a prova 2 será lecionada após a conclusão da parte 3 e as restantes prova serão realizadas após a finalização de todas as partes. Consoante o seu desempenho e vontade e tendo em conta o nível em que o aluno se encontra, este poderá eliminar uma prova prática, se for mágico, contando apenas para a nota final 3 provas e o teste, ou poderá eliminar 2 provas, e trocar por um projeto mais desafiante e aliciante, se for feiticeiro, contabilizando assim 2 provas, um projeto e um teste. Do lado esquerdo da Figura 4 pode-se ver os elementos do sistema gamificado e em que momento do semestre eles irão estar acessíveis. Os torneios realizam-se no final de cada parte, apenas para os alunos mais motivados e com melhor desempenho.



**Figura 4 - Ciclo Semestral**

## Pontuação

As ações dos alunos serão traduzidas por pontos. Cada tarefa tem determinado número de pontos associados a ela. Isto permite medir o esforço e desempenho do aluno durante todo o semestre.

Neste sistema gamificado irão existir dois tipos de pontos:

- **Pontos de avaliação** – Ganham-se realizando as tarefas associadas à avaliação que se irão traduzir na nota final do aluno. Pode adquirir-se um máximo de 1600 pontos: o correspondente a 20 valores.
- **Pontos de jogo** – Ganham-se realizando qualquer tipo de tarefa que seja possível realizar no sistema gamificado, independentemente de serem tarefas de avaliação ou não. Pode-se adquirir um número ilimitado de pontos de jogo.

Os **pontos de avaliação** irão também determinar o nível do aluno, sendo que todos começam como aprendiz, mas podem atingir o nível de feiticeiro, passando pelo de mágico. Abaixo descreve-se os níveis e os intervalos de pontos de cada um deles:

- Aprendiz – De 0 a 480 pontos (30% da nota máxima)
- Mágico – De 480 a 1120 pontos (70 % da nota máxima)
- Feiticeiro – De 1120 a 1600 pontos

Segue-se então a Tabela 1 com as tarefas de avaliação possíveis e os pontos associados por estatuto:

**Tabela 1 - Tabela de pontos de avaliação do nível Aprendiz**

Aprendiz			
Tarefa	Pontos	Percentagem	Observações
1 ° Trabalho Prático	200	50%	-----
2 ° Trabalho Prático	200		
3° Trabalho Prático	200		
4 ° Trabalho Prático	200		
Teste Teórico	640	40%	-----
Responder <i>Quizzes</i>	40	2.5%	3 Pontos cada <i>quiz</i> . Necessário realizar 13 <i>quizzes</i> para obter os 40 pontos (cerca de 1 quiz semanalmente).
Assiduidade/ Pontualidade	40	2.5%	Depende da quantidade de aulas (40/ n° aulas = pontos por aula).
Fazer questões no fórum	40	2.5%	10 Pontos por cada questão exposta. Necessário fazer 4 questões para obter os 40 pontos.

Responder a questões no fórum	40	2.5%	8 Pontos por cada questão esclarecida. Necessário responder a 6 questões para obter 40 pontos.
-------------------------------	----	------	--

**Tabela 2 - Tabela de pontos de avaliação do nível Mágico**

Mágico			
Tarefa	Pontos	Porcentagem	Observações
1 ° Trabalho Prático	260	50%	Pontuação adotada em caso do aluno eliminar um trabalho prático. Caso contrário será a mesma pontuação do aprendiz.
2 ° Trabalho Prático	270		
3 ° Trabalho Prático	270		
Teste Teórico	640	40%	-----
Responder <i>Quizzes</i>	40	2.5%	3 Pontos cada <i>quiz</i> . Necessário realizar 13 <i>quizzes</i> para obter os 40 pontos (cerca de 1 quiz semanalmente).
Assiduidade/ Pontualidade	40	2.5%	Depende da quantidade de aulas (40/ n° aulas = pontos por aula).
Fazer questões no fórum	40	2.5%	10 Pontos por cada questão exposta. Necessário fazer 4 questões para obter os 40 pontos.
Responder a questões no fórum	40	2.5%	8 Pontos por cada questão esclarecida. Necessário responder a 6 questões para obter 40 pontos.

**Tabela 3 - Tabela de pontos de avaliação do nível Feiticeiro**

Feiticeiro			
Tarefa	Pontos	Porcentagem	Observações
1 ° Trabalho Prático	200	50%	Pontuação adotada em caso do aluno eliminar dois trabalhos práticos. Caso elimine só um trabalho prático será adotada a mesma pontuação do Mágico. Caso contrário será a mesma pontuação do aprendiz.
2 ° Trabalho Prático	200		
Projeto	400		
Teste Teórico	640	40%	-----
Responder <i>Quizzes</i>	40	2.5%	3 Pontos cada <i>quiz</i> . Necessário realizar 13 <i>quizzes</i> para obter os 40 pontos (cerca de 1 quiz semanalmente).
Assiduidade/ Pontualidade	40	2.5%	Depende da quantidade de aulas (40/ n° aulas = pontos por aula).
Fazer questões no fórum	40	2.5%	10 Pontos por cada questão exposta. Necessário fazer 4 questões para obter os 40 pontos.

Responder a questões no fórum	40	2.5%	8 Pontos por cada questão esclarecida. Necessário responder a 6 questões para obter 40 pontos.
-------------------------------	----	------	--

Em seguida, irão ser apresentadas as tarefas em que se pode receber pontos de jogo, e os seus respetivos pontos:

**Tabela 4 - Tabela de Pontos de jogo**

Tarefa	Pontos	Observações
Fazer questão no fórum	10	Caso faça mais de 4 questões, os pontos não serão acumulados na pontuação correspondente à avaliação.
Responder a questão no fórum	8	Caso esclareça mais de 6 questões, os pontos não serão acumulados na pontuação correspondente à avaliação.
Partilhar <i>link</i> sobre a matéria/ programação	10	
Sugerir ideia para projetos de feiticeiro	50	
Sugerir exercício	20	

### 3.5 Diversão

Uma das partes essenciais para uma melhor aprendizagem é a diversão. É possível a aprendizagem de conteúdos entediantes, mas isso requer um maior esforço mental e por vezes faz com que a pessoa veja essa aprendizagem como algo que tem de ser feito e não como algo que a pessoa faça voluntariamente e por gosto.

Por outro lado sabe-se que para os alunos que estão a iniciar na área da programação, o modo de pensamento para compreensão desta área pode-se tornar lento e entediante se não houver alguma motivação extra que os faça continuar a querer aprender. Torna-se então importante definir quais são os aspetos que irão tornar divertida a aprendizagem dos alunos aquando da utilização deste sistema.

Para a criação desta diversão torna-se importante usar elementos que motivem intrinsecamente o aluno a aprender programação provocando emoções como a curiosidade, o desejo de aprender, de ajudar, e de resolver desafios complexos; mas também elementos de motivação extrínseca que podem ser uteis numa fase inicial de aprendizagem, quando o aluno

ainda não está muito conectado e envolvido. A diversão, tal como referido no capítulo 2, pode ser distinguida entre diversão desafiante, diversão simples, diversão séria e diversão social.

Segue-se então as atividades que mais poderão provocar um sentimento de diversão nos alunos enquanto aprender programação:

- **Completar a caderneta de cromos** – a primeira reação do aluno ao perceber da existência de uma caderneta de cromos deverá ser a intenção de completá-la. Ver a caderneta a ser preenchida aos poucos irá entusiasmar o aluno e fazer ao mesmo tempo com que ele participe nos desafios e atividades do sistema pois apenas é possível adquirir cromos atingindo determinados objetivos. Este processo de atingir objetivos e receber cromos por isso torna-se divertido ainda que alguns desses objetivos sejam difíceis de alcançar.
- **Responder a quizzes** – A participação nos *quizzes* pode ser uma maneira diferente da tradicional de aprender e testar os conhecimentos teóricos adquiridos. Torna-se ainda mais divertido pois o aluno não terá a pressão de acertar todas as respostas à primeira tentativa, podendo repetir o *quiz* as vezes que forem necessárias.
- **Esclarecer dúvidas de outros alunos nos fóruns** – Ajudar os outros alunos com as suas dúvidas criará conexões entre eles e fará com que as duas partes envolvidas se sintam bem: aquele que ajuda sente-se útil e fica feliz por poder contribuir e aquele que é ajudado sente-se agradecido pelas respostas recebidas de quem sabe.
- **Partilhar recursos adicionais com outros alunos sobre programação** – A partilha de vídeos, tutoriais ou artigos relacionados com programação pode ser um complemento para o melhor entendimento da matéria que está a ser lecionada e faz aumentar a curiosidade dos outros alunos sobre os conteúdos que foram partilhados provocando uma sensação de “prazer” após essa curiosidade ter sido satisfeita, e ao mesmo tempo faz com que o aluno aprenda sempre algo novo e diferente.
- **Competir em torneios** – tal como ao jogar um jogo onde se tenha de defrontar oponentes se torna divertido, o processo que competir com outros alunos em torneios de programação, e de ter em conta que o tempo e a eficácia da resolução dos desafios, pode tornar-se divertido mesmo que no final não se seja o vencedor.

### 3.6 Ferramentas a Utilizar

Para completar os 6 passos desta *framework* é necessário definir as ferramentas que irão ser utilizadas para atingir os objetivos traçados no primeiro passo, tendo em conta os comportamentos-alvo desejados, os ciclos, o tipo de jogador ou aluno e a diversão.

Então, em seguida, serão apresentados os componentes, mecânicas e dinâmicas criadas neste sistema ludificado:

### Componentes

- **Conquistas** – A aquisição de um cromo para adicionar à caderneta de cromos pressupõe a conquista dos objetivos nele mencionados.
- **Cromos** – A representação visual dos objetivos e feitos alcançados é dada pelos cromos.
- **Caderneta de cromos** - Todos nós quando eramos pequenos tínhamos uma caderneta de cromos e pedíamos aos nossos pais para comprar mais produtos que continham os cromos com o objetivo de ficar com a caderneta completa. Por vezes obtínhamos cromos repetidos e trocávamos com colegas para preencher mais rápido a caderneta. O objetivo de inserir uma caderneta neste sistema gamificado é fazer lembrar a nostalgia de criança e incentivar e motivar os alunos a preenchê-la por completo, sendo que para isso é necessário eles se esforçarem para obter a totalidade dos cromos. Segue-se a lista de cromos possíveis de adquirir

Lista de cromos:

- Realizou trabalho prático nº 1
- Realizou trabalho prático nº 2
- Realizou trabalho prático nº 3
- Realizou trabalho prático nº 4
- Realizou projeto
- Realizou teste teórico
- Realizou teste teórico em recurso
- Eliminou um trabalho prático
- Eliminou dois trabalhos práticos
- Obteve nota final máxima
- Obteve nota final superior a 15 valores
- Obteve mais de 70% no trabalho prático nº 1
- Obteve mais de 70% no trabalho prático nº 2
- Obteve mais de 70% no trabalho prático nº 3
- Obteve mais de 70% no trabalho prático nº 4
- Obteve mais de 70% no projeto
- Ganhou o título de Aprendiz
- Ganhou o título de Mágico
- Ganhou o título de Feiticeiro
- Sugeriu mais de 5 exercícios
- Sugeriu mais de 3 ideias para projetos de Feiticeiro
- Esclareceu mais de 10 dúvidas
- Formulou mais de 5 questões
- Partilhou 3 *links* sobre a matéria ou sobre programação
- Realizou mais de 10 *quizzes*
- Vencedor do torneio nº X (ou vencedor do torneio **nomeDoTorneio**)
- Completou o desafio do torneio nº X

- **Desbloqueio de Conteúdos** – Para um aluno progredir na UC, ele terá de participar na maior parte das atividades que irão ser inseridas no *moodle*, para assim poder desbloquear outras mais complexas. Por exemplo um aluno terá de ler primeiro o módulo do *e-book* para poder aparecer o *quiz* relacionado com a matéria desse módulo, para desbloquear os exercícios o aluno precisa de responder corretamente a todas as perguntas do *quiz* (tendo a possibilidade de responder as vezes que forem necessárias), e por aí adiante até todos os conteúdos estarem desbloqueados.
- **Pontos** – Cada desafio ou atividade tem um número de pontos associados a eles. Os pontos podem ser ganhos a partir da realização e conclusão das tarefas e desafios com sucesso.
- **Níveis** – Existem três níveis neste sistema (aprendiz, mágico e feiticeiro) que podem também ser considerados como estatutos. Para subir de nível os alunos precisam de ganhar os pontos ao realizar as atividades até preencherem completamente a barra de progressão inserida na plataforma *moodle*.
- **Aquisição de recursos ao subir de nível** – Ao alcançar níveis superiores no sistema, o aluno ganha recursos que o possibilitam escolher o caminho que mais preferem percorrer dentro da UC. Por exemplo, ao alcançar o nível de mágico, é possível eliminar a pior prova prática, enquanto que se o aluno chegar ao nível de feiticeiro poderá substituir duas provas práticas por um projeto.

### Mecânicas

- **Desafios** – Existem diversos tipos de desafios neste sistema, sendo uns mais complexos que os outros. Os de maior complexidade são os torneios, as provas práticas e o teste teórico. Apesar de não tão complexos, os exercícios que são enviados para casa para fazer e as autoavaliações requerem ainda algum grau significativo de esforço por parte do aluno. Já as tarefas como partilha de *links*, esclarecimento de dúvidas, *quizzes* e sugestões de exercícios e projetos, são tarefas que requerem um menor grau de esforço.
- **Competição** – A competição acontece nos torneios, onde o aluno mais rápido e com a melhor solução para o desafio proposto é aquele que ganha o torneio e a maior quantidade de pontos.
- **Cooperação** – Os alunos ajudam-se entre si aos responder às dúvidas que outros colegas colocam nos fóruns semanais para encontrar as melhores soluções para os problemas apresentados.
- **Feedback** - Para oferecer um maior *feedback* ao utilizador sobre a sua prestação durante a sua jornada, será possível visualizar algumas estatísticas como: a pontuação restante necessária para alcançar estatuto superior (mágico ou feiticeiro) e verificar se a sua nota está acima ou abaixo da média da turma. Segue em baixo a lista de informações que poderão ser consultadas nesta secção:
  - Pontos de avaliação necessários para atingir o próximo estatuto

- Notas de cada um dos trabalhos/ projeto e do teste
- Posicionamento de cada uma destas notas em relação à média da turma
- Nota corrente/ final do aluno
- Média da turma da nota corrente final
- N° de *quizzes* realizados
- Média de respostas certas nos *quizzes* realizados
- N° de *quizzes* onde acertei todas as perguntas
- Tempo mínimo num *quiz*
- Tempo máximo num *quiz*
- Tempo médio dos *quizzes* realizados
- Torneios realizados
- Torneios ganhos
- Torneios completados
- Torneios não completados
- Tempo mínimo para completar um torneio
- Tempo máximo para completar um torneio
- Pontos ganhos em torneios
- N° de perguntas feitas
- N° de perguntas esclarecidas (respostas)
- N° de sugestões de exercícios
- N° de sugestões de projeto para o estatuto de feiticeiro
- N° de *links* partilhados
- N° de cromos atual/ N° de cromos total

Outros elementos que também ajudam a dar feedback ao aluno são os pontos adquiridos quando um aluno conclui uma tarefa com sucesso, e os cromos que se vão conquistando ao atingir determinados objetivos.

### Dinâmicas

- **Restrições** – As restrições colocadas ao aluno são: o desbloqueio de atividades e desafios apenas após a conclusão de desafios já presentes no *moodle*, a eliminação da pior prova prática se o aluno atingir o nível de mágico, a substituição de duas provas práticas por um projeto caso o aluno atinja o nível de feiticeiro e a entrada em torneios só será possível caso o aluno seja feiticeiro.
- **Emoções** – As emoções provocadas aos alunos são a curiosidade sobre os conteúdos e sugestões de exercícios e projetos partilhados por outros alunos, a determinação em desbloquear todos os desafios da UC no *moodle* e para completar a caderneta de cromos, o desejo de vitória ao participar nos torneios, o gosto de ajudar os outros alunos a responder às suas dúvidas, a sensação de controlo na progressão na UC através das estatísticas e o uso da criatividade na criação de exercícios e projetos e na resolução de problemas.
- **Progressão** – A progressão do aluno na disciplina pode ser representada através do desbloqueio de desafios, da acumulação de pontos, da subida de níveis e da conquista de cromos para preencher a caderneta.

- **Relações** – As relações entre os alunos pode acontecer através da cooperação nos fóruns de esclarecimento de dúvidas e através da competição nos torneios que acontecem no final de cada módulo.

### 3.7 Análise do sistema com a *framework Octalsys*

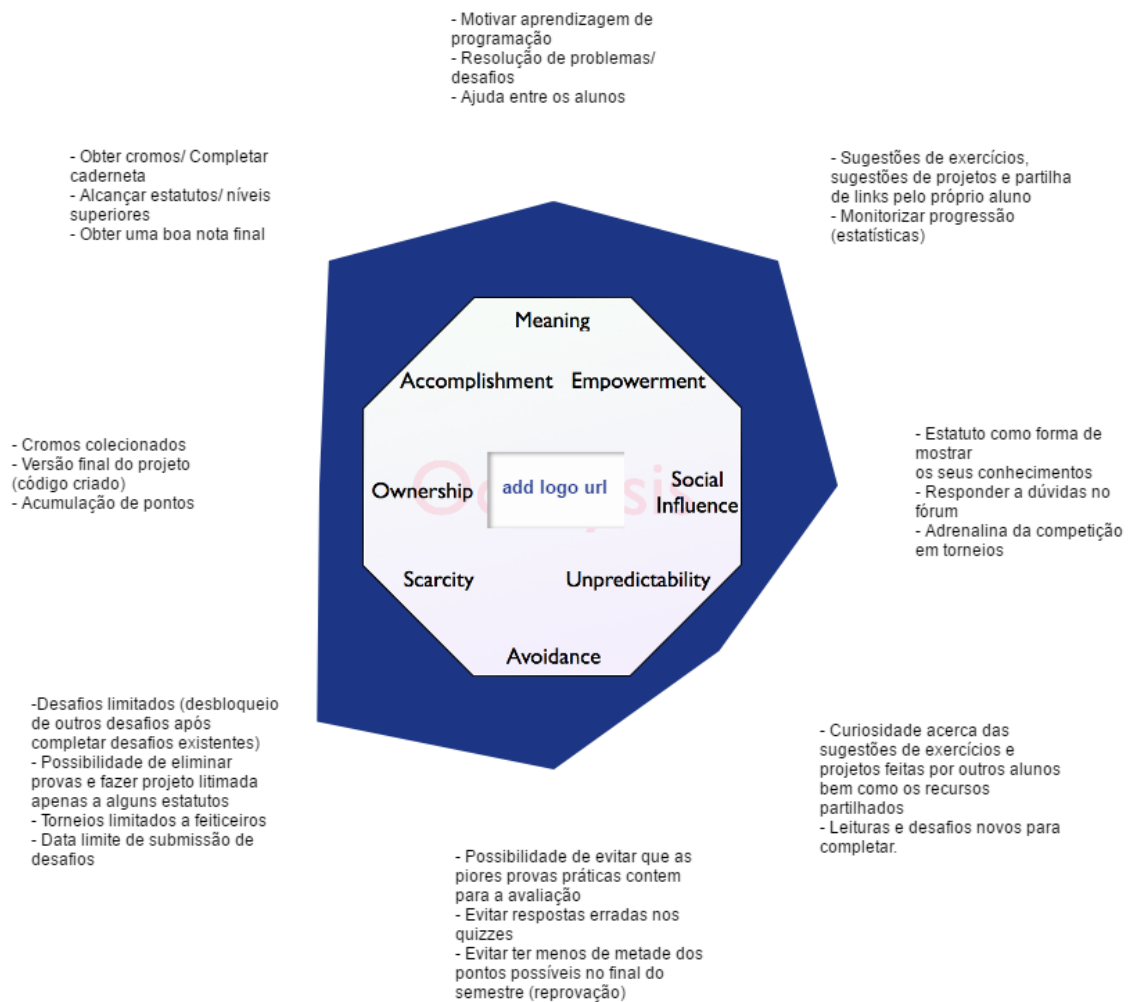
Para validar o sistema descrito anteriormente, far-se-á uso da *framework Octalsys* para entender os seus pontos fortes e os pontos fracos, se faz mais uso da criatividade ou da lógica e se o sistema se concentra mais da parte inferior do octógono, que geralmente tem uma conotação mais negativa e o jogador sente-se motivado a realizar as tarefas por medo de perda ou por imprevisibilidade do que poderá vir a seguir; ou se o sistema é mais forte na parte superior do octógono, onde o jogador pode expressar a sua criatividade, sentir-se bem-sucedido ao dominar uma habilidade nova ou possuir um significado maior nas tarefas que realiza.

Na Figura 5 é apresentada a análise do sistema ludificado criado segundo a *framework octalsys*.

Como se pode ver, o sistema está relativamente equilibrado em termos dos fatores presentes no octógono, no entanto o desenvolvimento e realização, fortalecimento da criatividade e *feedback* e também a escassez e impaciência são os que mais definem e mais importância têm no sistema.

No campo da imprevisibilidade e curiosidade, as sugestões de ideias e *links* partilhados por outros alunos servem para estimular a curiosidade em aprender mais sobre programação, no entanto os alunos que exploram o que os outros partilham não terão grande vantagem dentro do próprio sistema. Já as leituras e desafios que são lançados são importantes pois é necessária a sua realização para progredir dentro da UC.

No campo da perda e evitação, o aspeto mais importante é evitar ter menos de metade dos pontos que são possíveis de adquirir no sistema, pois isso significaria a reprovação do aluno na disciplina. Existe também a possibilidade de evitar que as piores provas práticas contem para a classificação final e evitar respostas erradas nos *quizzes* pois só assim o aluno poderá progredir na matéria.



**Figura 5 - Sistema ludificado aplicado à framework Octalsys**

Quanto à escassez, o limite de leituras e desafios que é possível fazer a cada altura são poucos. Este aspeto pode induzir a ideia que o aluno tem pouca possibilidade de escolha sobre os desafios que pretende completar, no entanto, é definido este método como uma forma de guiar o aluno durante todo o semestre, desde o nível mais baixo de dificuldade até alcançar os desafios mais complexos. O recurso de eliminação de provas e substituição por um projeto também está apenas disponível para aqueles alunos que consigam chegar a níveis superiores (mágico ou feiticeiro) bem como a participação nos torneios é apenas possível sendo um feiticeiro. Outro ponto importante é a escassez ou limite de tempo para submissão de alguns desafios.

A ter o estatuto de feiticeiro, o aluno torna-se mais influente e os outros alunos podem levar mais em consideração os seus conhecimentos, sugestões e respostas a

dúvidas no fórum. Por fim a adrenalina da competição em torneios é também um aspeto a considerar no campo da influência social.

No fator de posse encontram-se os cromos adquiridos ao alcançar objetivos e participar no sistema e obter bom desempenho nas tarefas realizadas. Em caso de realização de um projeto, o código criado será propriedade do autor do projeto e a acumulação de pontos servirá como forma de passar de nível e irá traduzir-se na nota final do aluno.

Quanto ao fortalecimento da criatividade, são atribuídos pontos aos alunos que sugeriram exercícios, projetos e partilhem *links* como forma de recompensa pela sua criatividade e ajuda no crescimento do conhecimento e aprendizagem de programação. Já as estatísticas dão o feedback necessário aos alunos sobre o seu desempenho e progressão na UC.

No campo do desenvolvimento e realização, a aprendizagem da matéria lecionada na disciplina e a sua aplicação faz com que o aluno atinja algumas conquistas e realizações como completar a caderneta de cromos, alcançar estatutos/ níveis superiores e obter uma boa nota no final do semestre.

Finalmente os principais significados deste sistema é a motivação dos alunos para a aprendizagem de programação a entreajuda dos alunos e a resolução de problemas e desafios.

No geral o sistema tem características que possibilitam tanto o uso da parte esquerda do cérebro, associada à racionalidade e às motivações extrínsecas, como à parte direita do cérebro, associado à parte emocional e às motivações intrínsecas. Ainda que o fator da escassez tenha grande importância neste sistema, estas características estão mais concentradas na parte superior do octógono o que faz deste sistema “*white hat gamification*”, significando que existem mais motivadores positivos do que negativos.



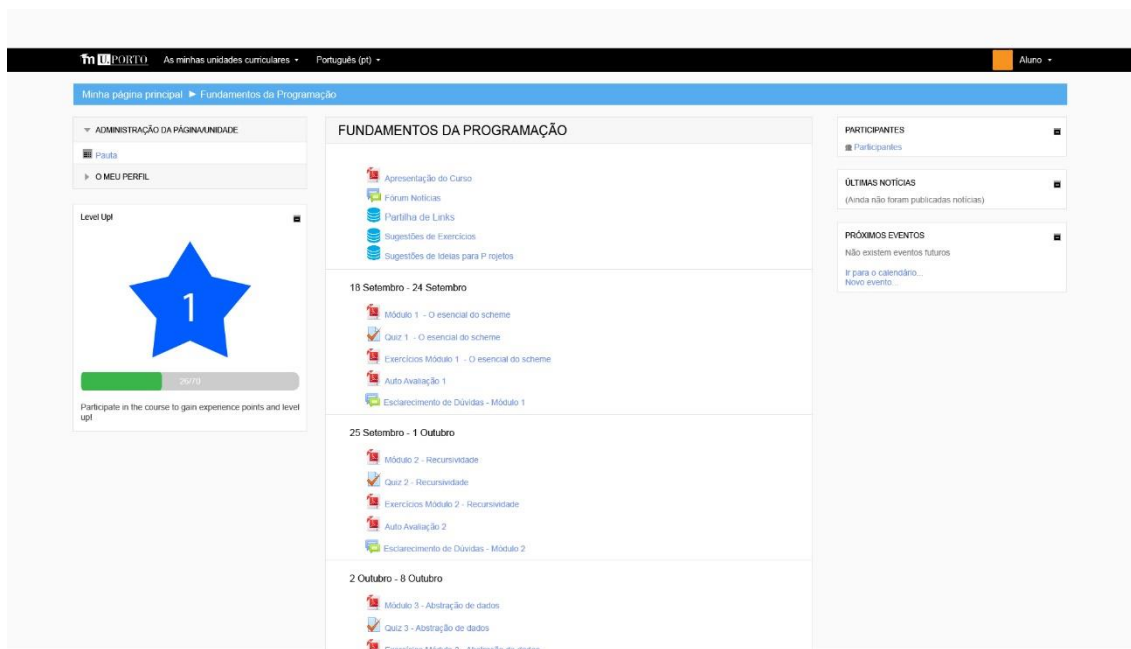
## 4. Implementação do Sistema no Moodle

Neste capítulo irão ser apresentadas representações gráficas dos elementos que fazem parte do sistema ludificado criado dando uma explicação geral de como cada um funciona e oferecendo algum apoio que possibilite uma possível implementação futura do sistema na plataforma *moodle* para que no futuro seja possível a realização de testes e investigações com os alunos que frequentem a UC de fundamentos de programação.

Esta secção irá falar dos seguintes elementos do sistema: página geral da UC de FPRO no *moodle* e *plugin* “Level Up!”, caderneta de cromos, fóruns, partilha de *links*, sugestões de exercícios, sugestões de ideias para projetos, *quizzes* e outras atividades.

### 4.1 Página de Fundamentos de Programação no Moodle

A página inicial da unidade curricular de FPRO, é onde os alunos poderão ter acesso a uma lista de todos os recursos disponíveis para a progressão do aluno dentro da disciplina, sejam estes *quizzes*, exercícios, autotestes, provas práticas para submissão, e também os fóruns, partilha de *links*, sugestões de exercícios e sugestões de projetos. Os recursos no geral irão sendo desbloqueados à medida que o aluno progride dentro da UC, ou seja, é necessário realizar primeiro certo tipo de atividades, e completa-las com sucesso, para depois serem desbloqueados novos desafios. Para ser possível realizar esse sistema de desbloqueio e conclusão e tarefas deve-se verificar se as opções de “Ativar rastreamento de conclusão” e “Permitir acesso condicional” estão habilitadas nas opções avançadas do *site*.



**Figura 6 - Representação do plugin Level Up! à esquerda e do formato semanal da UC no centro**

A lista de atividades que o aluno poderá realizar encontra-se no centro da página inicial do moodle e está dividida por semanas. A ordem dos desafios que irão aparecer irá ser guiada pelo ciclo semanal e o ciclo semestral apresentados no capítulo anterior. Por cada tarefa que o aluno concluir, irá receber uma certa quantidade de pontos que o ajudará a progredir na disciplina.

Do lado esquerdo do ecrã (Figura 6), encontra-se o *plugin* “Level Up!”. Esse *plugin* terá como função a gestão dos níveis do sistema gamificado e dos pontos que serão atribuídos a cada tarefa que o aluno realizar. Para a sua utilização deve-se fazer primeiro a sua ativação através da opção “*plugins*” no painel de administração do *site*.

Nas configurações do *plugin* “Level Up!” é possível definir quantos níveis irão existir e depois definir os pontos de início e de fim de cada nível, utilizando como base as pontuações de cada nível definidos no capítulo anterior, no tópico relacionado com os pontos de avaliação. É também possível substituir o símbolo que é apresentado por definição para cada nível por uma ilustração que possa fazer mais sentido. No caso deste sistema poderia estar relacionado com as designações de “Aprendiz”, “Mágico” e “Feiticeiro”.

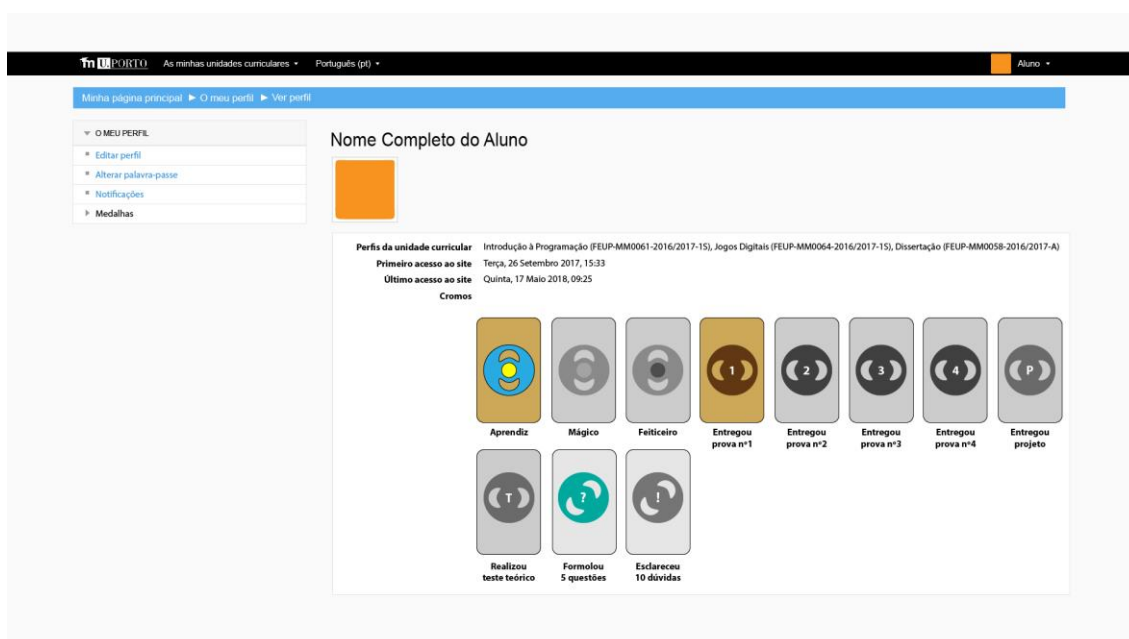
Uma das partes mais importantes neste *plugin* são as regras e onde é possível definir o número de pontos para cada ação realizada pelo jogador. O *plugin* já traz alguns pontos definidos por padrão, no entanto, caso as regras coincidam, as do utilizador prevalecem sempre perante as regras padrão.

Outras opções que o professor também tem à sua disposição são os relatórios onde pode visualizar as estatísticas de cada um dos alunos (nome, nível, pontos e barra de progressão) tal como pode alterar os pontos dos alunos manualmente; as informações dos níveis, onde aparece as descrições de todos os níveis numa tabela, mostrando o número do nível, os pontos de experiência necessários para entrar em determinado nível e o nome dado ao nível; e também o log que é um registo de todas as ações realizadas pelos alunos. Na tabela é identificada a data da ação, quem a realizou, quantos pontos ganhou e qual foi a tarefa realizada.

## 4.2 Caderneta de Cromos

O conceito de caderneta de cromos aparece no sistema como forma de motivar mais o aluno a realizar as tarefas pedidas, a alcançar determinados objetivos e também como forma de progredir na UC.

Para aplicar o conceito de caderneta de cromos na plataforma *moodle*, o aluno deve, desde o princípio da unidade curricular, ter a possibilidade de visualizar todos os cromos que serão possíveis de adquirir durante todo o semestre, mostrando quais são os objetivos que se deve atingir para adquirir cada um dos cromos. Antes da sua aquisição, os cromos estariam na página do perfil do aluno, apresentados a preto e branco, e quando atingissem os objetivos que os permitissem desbloquear um cromo, este passaria a ser apresentado com as cores originais (ver Figura 7).



**Figura 7 - Caderneta de cromos no perfil do aluno, no qual os cromos a preto e branco são aqueles que ainda não foram adquiridos e os cromos a cor são aqueles que já foram conquistados**

Para fazer uso deste conceito de caderneta de cromos e para poder atribuí-los aos alunos, é necessário verificar nas opções avançadas, onde se pode ir através do painel da administração do *site*, se as opções de “Ativar badges” e de “Ativar rastreamento de conclusão” estão habilitadas. Estas opções irão permitir a utilização dos cromos e irão verificar se as tarefas que são necessárias realizar para obter determinado cromo estão concluídas, para que após a conclusão possa ser atribuído o cromo automaticamente.

No entanto, não é possível tornar este processo completamente automático apenas com as definições que o *moodle* nos disponibiliza. Então, para isto ser possível o professor teria de atribuir todos os cromos a preto e branco a todos os alunos manualmente no início da cadeira, depois atribuir os cromos de cor automaticamente aquando da finalização das tarefas necessárias à sua aquisição e após isso o professor teria de monitorizar os cromos dos alunos diariamente e, se algum aluno tivesse adquirido um cromo de cor, o docente teria de eliminar manualmente o cromo a preto e branco do aluno correspondente.

### **4.3 Fóruns de Esclarecimento de Dúvidas**

Com a criação de fóruns pretende-se proporcionar uma maior interação entre alunos e dar-lhes oportunidade de expor as dúvidas que vão encontrando ao longo da sua jornada semestral na UC de FPRO e permite a qualquer aluno que saiba a resposta ajudar os seus colegas com dificuldades. Estes fóruns seriam criados semanalmente permitindo uma melhor organização, fazendo com que as dúvidas apresentadas pelos alunos em cada fórum estejam relacionadas com as respetivas matérias lecionadas em cada semana.

Nesta atividade semanal poderão ser atribuídos pontos tanto aos alunos que expõem as suas dúvidas como aos alunos que respondem às dúvidas. Com isto pretende-se premiar tanto o interesse e curiosidade em aprender mais sobre a matéria como aqueles que têm vontade em ajudar os outros a terem sucesso também. Contudo, é necessário ter em atenção o uso abusivo desta atividade para angariação de pontos. É então necessário arranjar uma forma de verificar se as respostas dadas dos alunos às perguntas expostas são coerentes, claras e explicativas, antes da atribuição dos pontos.

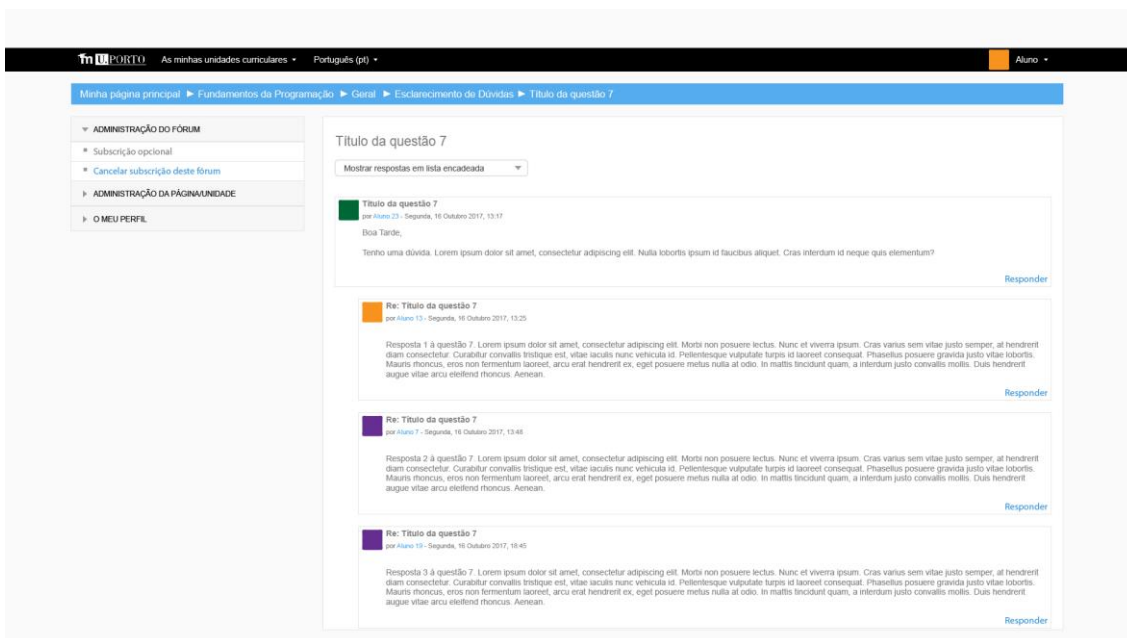
Tópico	Iniciado por	Respostas	Não lidas	Última mensagem
Título da questão1	Aluno 12	0	0	Aluno 12 Qui, 23 Nov 2017, 16:32
Título da questão2	Aluno 25	1	0	Aluno 11 Sex, 17 Nov 2017, 13:21
Título da questão3	Aluno 08	0	0	Aluno 08 Seg, 13 Nov 2017, 10:54
Título da questão4	Aluno 16	2	2	Aluno 24 Ter, 7 Nov 2017, 15:02
Título da questão5	Aluno 15	4	0	Aluno 04 Sex, 25 Out 2017, 14:17
Título da questão6	Aluno 22	6	0	Aluno 14 Sex, 20 Out 2017, 09:14
Título da questão7	Aluno 05	3	1	Aluno 19 Seg, 16 Out 2017, 18:45
Título da questão8	Aluno 17	2	1	Aluno 05 Qua, 11 Out 2017, 20:37
Título da questão9	Aluno 23	4	0	Aluno 20 Ter, 3 Out 2017, 16:32

**Figura 8 - Lista de tópicos no fórum para esclarecimento de dúvidas**

Na **figura 8 e 9** podem ser vistas as representações gráficas de como seria o fórum. Na primeira imagem está a lista de tópicos. Os tópicos são criados pelos alunos e cada um refere-se a uma dúvida que determinado aluno quer ver esclarecida. Os alunos deixam de poder criar tópicos num fórum semanal quando uma nova semana se iniciar, no entanto, podem continuar a responder às questões já feitas anteriormente. Para uma distinção mais facilitada, o título do fórum indica a semana de aulas a que corresponde esse fórum. O aluno conseguirá notar que existem respostas novas quando existe um realce diferente na cor de um tópico.

Já na segunda imagem, pode-se verificar o título na parte superior, onde normalmente aparece a pergunta em questão que se deseja ver esclarecida. Por baixo tem uma breve descrição da pergunta feita pelo aluno para oferecer mais alguma informação e detalhe a esta. Os outros comentários são as respostas dadas pelos alunos que querem ajudar. Em caso de sermos um estudante que está a responder à pergunta, apenas podemos ver a resposta dos outros estudantes quando nós próprios respondermos à pergunta.

Quando se for criar o fórum deve-se ter em atenção em como é atribuída a avaliação (deve ser dada pelo professor e atribuída uma escala de 1 a 8 para cada resposta dada e 10 pontos para cada tópico criado) e também restringir a data em que cada fórum irá aparecer para fazer coincidir com o início de cada semana respetiva.



**Figura 9 - Tópico do fórum onde é discutida a dúvida de um aluno e os outros a tentam esclarecer**

## 4.4 Quizzes

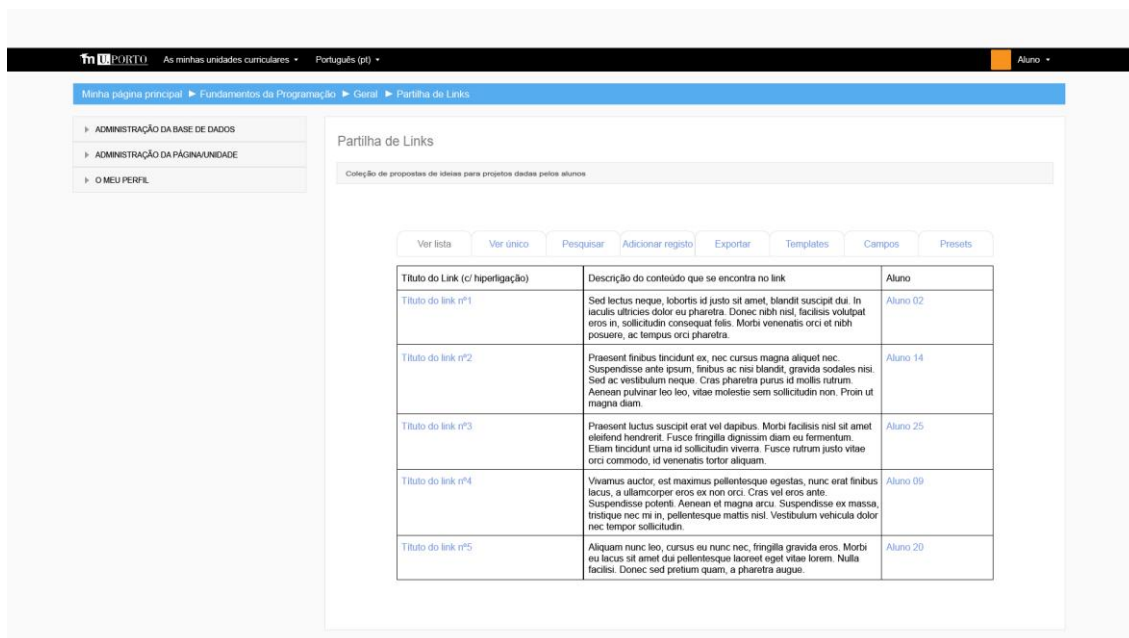
Os *quizzes* servem como forma de testar os conhecimentos adquiridos dos alunos após a leitura dos módulos respetivos. Pode-se também considerá-los como sendo uma maneira mais divertida para aprender os conteúdos quando comparado com a leitura do e-book. Estes desafios podem ser encontrados no *moodle* pelos alunos em formato de questionário e são feitos com uma regularidade semanal.

Os questionários não deverão ultrapassar as 10 perguntas e podem ter diversos formatos, como escolha múltipla, verdadeiro ou falso ou até mesmo fazer correspondências, consoante aquilo que mais convém ao professor. Outra das características destes questionários é o facto de poderem ser realizados um número de vezes ilimitadas até o aluno acertar em todas as questões. Os pontos são atribuídos no final do *quiz* consoante o número de respostas certas e erradas. O total de pontos que podem ser adquiridos em todos os *quizzes* é de 40 pontos, sendo que para descobrir quanto vale cada *quiz*, terá de se dividir 40 pelo número de semanas em que irá existir estes desafios para resolver.

Quando se for criar os questionários deve-se ter em atenção que a atividade só pode ser mostrada aos alunos na semana em que a matéria contida neles é lecionada e só será considerada como finalizada quando o aluno visualizar o questionário e receber uma nota.

## 4.5 Partilha de Links

A partilha de conteúdos adicionais (vídeos, tutoriais, artigos) foi criada com o propósito de incentivar o aluno a pesquisar mais na internet para além daquilo que é transmitido na unidade curricular e partilhar esses conteúdos com os outros alunos para que todos possam trocar conhecimentos e aprender em conjunto. Esta atividade serve como forma de complemento para os alunos expandirem o seu conhecimento e entender melhor a matéria.



Titulo do Link (c/ hiperligação)	Descrição do conteúdo que se encontra no link	Aluno
Titulo do link nº1	Sed lectus neque, lobortis id justo sit amet, blandit suscipit dui. In iaculis ultricies dolor eu pharetra. Donec nibh nisi, facilisis volutpat eros in, sollicitudin consequat felis. Morbi venenatis orci et nibh posuere, ac tempus orci pharetra.	Aluno 02
Titulo do link nº2	Praesent finibus tincidunt ex, nec cursus magna aliquet nec. Suspendisse ante ipsum, finibus ac nisi blandit, gravida sodales nisi. Sed ac vestibulum neque. Cras pharetra punis id mollis rutrum. Aenean pulvinar leo leo, vitae molestie sem sollicitudin non. Proin ut magna diam.	Aluno 14
Titulo do link nº3	Praesent luctus suscipit erat vel dapibus. Morbi facilisis nisi sit amet eleifend hendrerit. Fusce fringilla dignissim diam eu fermentum. Etiam tincidunt urna id sollicitudin viverra. Fusce rutrum justo vitae orci commodo, id venenatis tortor aliquam.	Aluno 25
Titulo do link nº4	Vivamus auctor, est maximus pellentesque egestas, nunc erat finibus iaculis, a ullamcorper eros ex non orci. Cras vel eros ante. Suspendisse potenti. Aenean et magna arcu. Suspendisse ex massa, tristique nec ma in, pellentesque mattis nisi. Vestibulum vehicula dolor nec tempor sollicitudin.	Aluno 09
Titulo do link nº5	Aliquam nunc leo, cursus eu nunc nec, fringilla gravida eros. Morbi eu luctus sit amet dui pellentesque laoreet eget vitae lorem. Nulla facilisi. Donec sed pretium quam, a pharetra augue.	Aluno 20

Figura 10 - Lista de conteúdos partilhados pelos alunos numa base de dados

Na Figura 10 pode-se visualizar um exemplo de uma lista de *links* partilhados pelos alunos que está organizada numa tabela. Este formato pode ser criado no separador de *templates*.

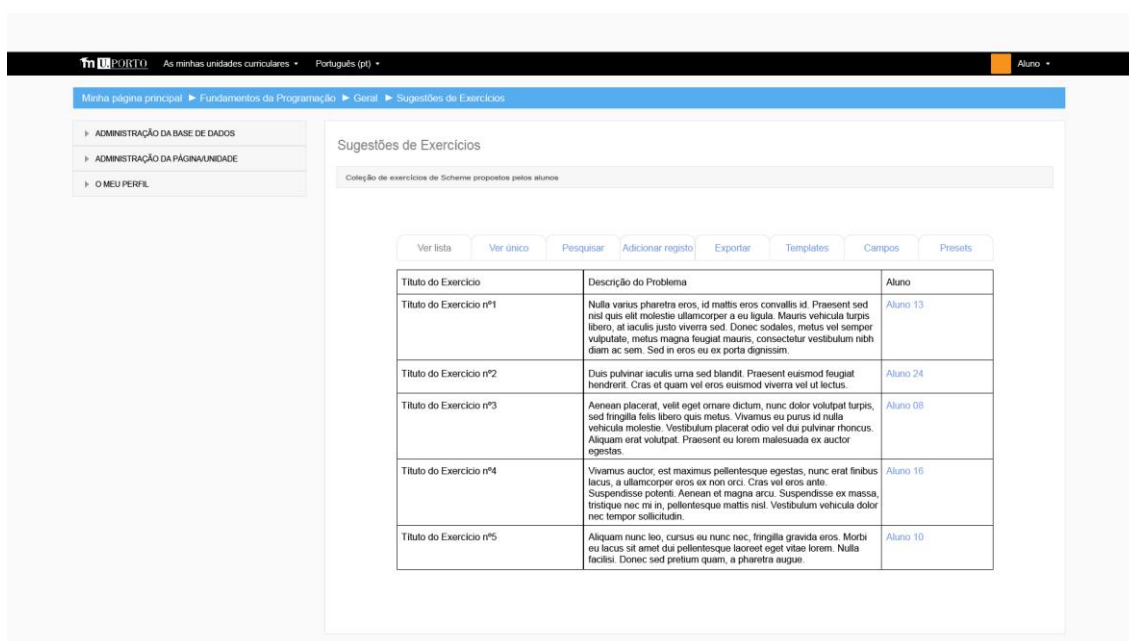
Para a criação desta atividade no *moodle* deve ser escolhida a base de dados como forma de organizar os conteúdos. Ao se criar a base de dados o docente deve criar os campos que irão permitir ao aluno criar novos registos. Os campos a serem criados seriam: uma entrada de texto onde o aluno poderá dar um título ou definir um assunto acerca do conteúdo que está a partilhar, um URL que será o *link* onde se encontra os conteúdos e uma área de texto que servirá para o aluno explicar melhor o que se poderá encontrar ao abrir o *link*.

Por cada *link* partilhado o aluno receberá 10 pontos. No entanto, deve também existir um controlo dos registos antes de serem publicados para os outros alunos, logo sugere-se que ao criar a base de dados se diga que é necessária a aprovação do docente primeiro. Por fim, esta atividade só deverá aparecer na página inicial da disciplina para os alunos quando a 1ª parte da matéria estiver lecionada.

## 4.6 Sugestões de Exercícios

O espaço para sugestão de exercícios tem como objetivos desafiar outros alunos a resolver os exercícios propostos na sala de aula ou em casa. Esta atividade permite utilizar a criatividade do aluno para ele criar exercícios simples que gostaria de ver resolvidos, e ao criar exercícios, o aluno experiencia, em parte, o papel do professor ao criar desafios para os alunos resolverem. Esta atividade pode também fazer com que o professor tenha também uma panóplia maior de exercícios para oferecer aos alunos nos anos seguintes.

Pretende-se premiar o aluno pela sua criatividade e originalidade, então aquele que fizer uma entrada de registo de um exercício para resolver, e este for aprovado pelo professor, o aluno receberá 20 pontos e o exercício irá aparecer para os outros alunos.



Ver lista Ver único Pesquisar Adicionar registo Exportar Templates Campos Presets

Titulo do Exercício	Descrição do Problema	Aluno
Titulo do Exercício nº1	Nulla varius pharetra eros, id mattis eros convallis id. Praesent sed nisi quis elit molestie ullamcorper a eu ligula. Mauris vehicula turpis libero, at iaculis justo viverra sed. Donec sodales, metus vel semper vulputate, metus magna feugiat mauris, consectetur vestibulum nibh diam ac sem. Sed in eros eu ex porta dignissim.	Aluno 13
Titulo do Exercício nº2	Duis pulvinar aculis urna sed blandit. Praesent euismod feugiat hendrerit. Cras et quam vel eros suscipit viverra vel ut lectus.	Aluno 24
Titulo do Exercício nº3	Aenean placerat, velit eget ornare dictum, nunc dolor volutpat turpis, sed fringilla feles libero quis metus. Vivamus eu purus id nulla vehicula molestie. Vestibulum placerat odio vel dia pulvinar rhoncus. Aliquam erat volutpat. Praesent eu lorem malesuada ex auctor egestas.	Aluno 08
Titulo do Exercício nº4	Vivamus auctor, est maximus pellentesque egestas, nunc erat finibus lacus, a ullamcorper eros ex non orci. Cras vel eros ante. Suspendisse potenti. Aenean et magna arcu. Suspendisse ex massa, tristique nec ma in, pellentesque mattis nisi. Vestibulum vehicula dolor nec tempor sollicitudin.	Aluno 16
Titulo do Exercício nº5	Aliquam nunc leo, cursus eu nunc nec, fringilla gravida eros. Morbi eu lacus. Sit amet dia pellentesque laoreet eget vitae lorem. Nulla facilis. Donec sed pretium quam, a pharetra augue.	Aluno 10

**Figura 11 - Base de dados contendo uma lista de sugestões de exercícios propostos pelos alunos**

Tal como foi referido no tópico da partilha de *links* e como irá ser utilizado também nas sugestões de projetos, foi escolhida a base de dados pois considera-se ser a melhor opção para organizar a informação. Os dados são também apresentados em forma de tabela. Os campos que devem ser criados para esta base de dados são: uma entrada de texto onde o aluno poderá definir um título para o exercício e uma área de texto para o aluno descrever o enunciado do exercício.

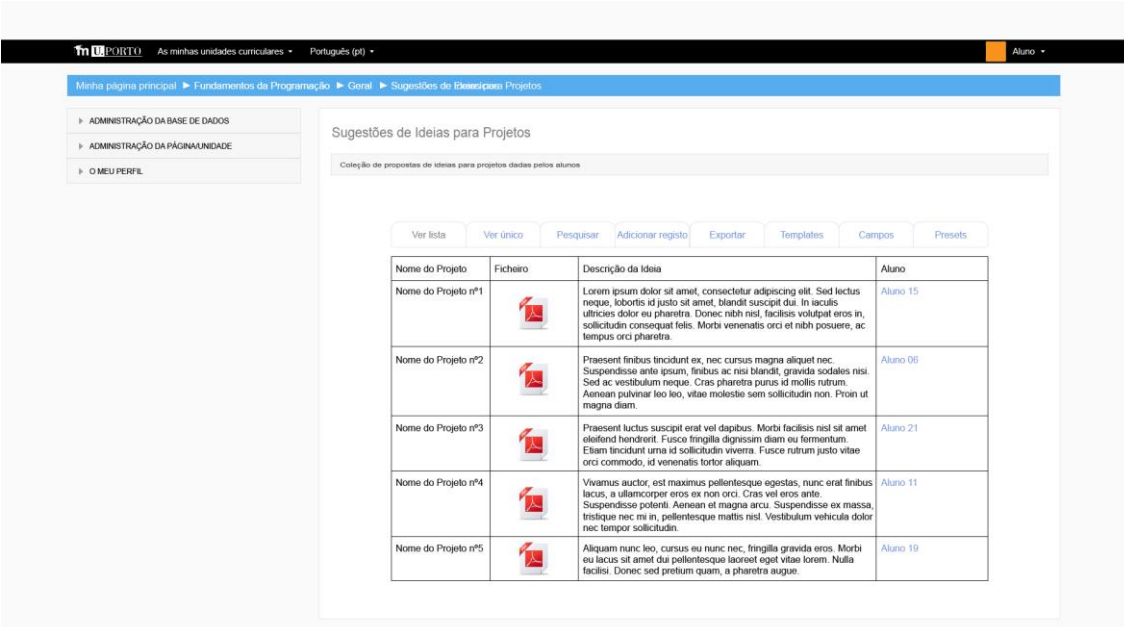
Esta atividade só deverá aparecer na página inicial da disciplina para os alunos quando a 2ª parte da matéria estiver lecionada.






## 4.7 Sugestões de Projetos

Com o objetivo de promover a criatividade e a geração de ideias, este espaço foi criado para que todos os alunos possam partilhar as suas ideias de projetos com os outros alunos e para que os alunos feitores que decidam realizar projeto e não tenham noção daqui que querem fazer possam escolher uma ideia para implementar. Caso haja uma grande quantidade de ideias, estas poderão entrar na lista de escolha nos anos seguintes.

Como para criar uma ideia para um projeto é mais complexo e demora mais tempo do que simplesmente criar novos exercícios simples, a pontuação atribuída a esta atividade é de 50 pontos e estes serão atribuídos aos alunos que inserirem um novo registo na base de dados e que este seja aceite pelo docente.

Os campos que devem ser disponibilizados para o aluno adicionar um registo são: uma entrada de texto onde o aluno poderá definir o nome para o projeto, uma imagem que poderá ser de preenchimento opcional e poderá ajudar o aluno a ilustrar melhor a sua ideia, um arquivo caso se queira dar uma explicação mais extensiva da ideia de projeto e uma área de texto para o aluno descrever o projeto com maior detalhe.



Nome do Projeto	Ficheiro	Descrição da Ideia	Aluno
Nome do Projeto nº1		Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed lectus neque, lobortis id justo sit amet, blandit suscipit du. In iaculis ultrices dolor eu pharetra. Donec nibh nisi, facilisis volutpat eros in, sollicitudin consequat felis. Morbi venenatis orci et nibh posuere, ac tempus orci pharetra.	Aluno 15
Nome do Projeto nº2		Praesent finibus tincidunt ex, nec cursus magna aliquet nec. Suspendisse ante ipsum, finibus ac nisi blandit, gravida sodales nisi. Sed ac vestibulum neque. Cras pharetra purus id mollis rutrum. Aenean pulvinar leo leo, vitae molestie sem sollicitudin non. Proin ut magna diam.	Aluno 00
Nome do Projeto nº3		Praesent luctus suscipit erat vel dapibus. Morbi facilisis nisi sit amet eleifend hendrerit. Fusce fringilla dignissim diam eu fermentum. Etiam tincidunt urna id sollicitudin viverra. Fusce rutrum justo vitae orci commodo, id venenatis tortor aliquam.	Aluno 21
Nome do Projeto nº4		Vivamus auctor, est maximus pellentesque egestas, nunc erat finibus lacus, a ullamcorper eros ex non orci. Cras vel eros ante. Suspendisse potenti. Aenean et magna arcu. Suspendisse ex massa, tristique nec mi in, pellentesque mattis nisi. Vestibulum vehicula dolor nec tempor sollicitudin.	Aluno 11
Nome do Projeto nº5		Aliquam nunc leo, cursus eu nunc nec, fringilla gravida eros. Morbi eu lacus sit amet du pellentesque laoreet eget vitae lorem. Nulla facilisi. Donec sed pretium quam, a pharetra augue.	Aluno 19

**Figura 12 - Base de dados contendo uma lista de sugestões de ideias para projetos propostos pelos alunos**

Na Figura 12 pode ser visto um exemplo de uma lista de ideias organizadas numa tabela, incluindo os campos referidos acima, não estando apenas disponível a coluna da imagem representativa do projeto.

Quando se for implementar esta atividade no *moodle* deve-se restringir o acesso dos alunos até a 3ª parte da matéria do *e-book* estar lecionada.

## 4.8 Outras Atividades

Para além dos elementos já descritos neste capítulo é importante também fazer referência a outras atividades como os exercícios propostos pelo professor, os autotestes, as provas práticas e os projetos.

Os exercícios têm como objetivo fazer com que os alunos testem na prática os conhecimentos teóricos adquiridos. Estes exercícios ou TPC podem ser realizados em casa mas também nas aulas teórico práticas com a ajuda do docente e são desafios de carácter opcional, não contabilizando para a nota final. No entanto, sendo que se pode fazer os *quizzes* de forma ilimitada até acertar em todas as perguntas, decidiu-se que ao adiciona-los ao *moodle*, deve-se restringir o aparecimento dos exercícios apenas após a conclusão do *quiz* da semana.

A realização dos autotestes é um momento de avaliação onde se irá testar o conhecimento dos alunos a produzir código e a resolver determinados problemas e serão realizados no final de cada ciclo semanal. O aluno deve submeter o autoteste num determinado formato na tarefa criada no *moodle* para esse efeito. Posteriormente eles passarão por um processo de correção automática e será dada a nota que irá ser adicionada à pontuação do aluno. A tarefa deve estar restringida ao aluno e só deverá aparecer após o aluno fazer o questionário ou após a data da aula teórico-prática semanal.

As provas práticas e o projeto são elementos de avaliação de grande importância contabilizando no seu conjunto 50% da nota final do aluno. Para adicionar à página do *moodle* estes arquivos para que o aluno os possa ver e realizar, terá de se fazer uso das tarefas. Nestes desafios não existirão restrições de acesso para além da data de início do começo destas. No entanto, a tarefa relacionada com o projeto terá um tipo diferente de restrição pois para poder ser feita, é necessário ter média de igual ou acima de 85% da nota das duas primeiras provas práticas. Contudo, para se concluir estas tarefas, todas elas precisam de receber uma nota.

# 5. Apresentação e Discussão dos Resultados

## 5.1 Questionários aos Alunos

Após o desenvolvimento do sistema ludificado seguindo os 6 passos de Werbach e utilizando posteriormente a *framework* octalsys, decidiu-se fazer um questionário aos alunos que frequentaram a unidade curricular de FPRO no ano letivo 2016/ 2017. O questionário estava dividido essencialmente em duas partes: a primeira pretendia obter a opinião dos alunos sobre os elementos de ludificado que lhes foram apresentados no 1º semestre e a segunda parte continha afirmações que os alunos poderiam avaliar numa escala de 1 a 6, com a intenção de dar pistas sobre como melhorar este sistema de forma a motivar mais os alunos na aprendizagem de programação e potenciar a individualização dos seus percursos dentro da unidade curricular.

O questionário foi enviado para 152 alunos, dos quais 36 responderam. As afirmações presentes no questionário foram configuradas para serem respondidas numa escala de Likert, onde o aluno poderia escolher o seu grau de concordância ou discordância em relação a estas. A escala estava definida da seguinte forma:

- 1 – Discordo Completamente
- 2 – Discordo
- 3 – Não Concordo nem Discordo
- 4 – Concordo
- 5 – Concordo Completamente
- 6 – Não Sei/ Sem Opinião

As proposições colocadas na primeira parte do questionário foram:

1. O sistema de pontos que foi estabelecido para poder anular as piores provas práticas é claro
2. Os níveis de aprendiz, mágico e feiticeiro são relevantes para a minha aprendizagem
3. É injusto que só estudantes com média de 85% à PP1 e PP2 possam chegar a feiticeiro
4. A autoavaliação não deveria acrescentar pontos
5. Havia poucos desafios para ganhar pontos
6. Todos os estudantes deveriam ter as mesmas condições de avaliação independentemente do nível a que chegaram
7. Esta estratégia motivou-me a trabalhar mais semanalmente

A Tabela 5 apresenta os resultados obtidos na primeira parte do questionário realizado aos alunos, relacionado com os elementos de ludificação introduzidos em FPRO. São apresentadas as respostas dadas por proposição em cada valor da escala de Likert, e ainda a média de cada proposição excluindo as respostas “Não sei/ sem opinião”.

**Tabela 5 - Resultados da primeira parte do questionário realizado aos alunos**

Questão	1	2	3	4	5	Média	6
1	0	1	4	12	19	4,36	0
2	3	5	11	10	7	3,36	0
3	6	12	5	10	1	2,64	2
4	11	8	11	3	0	2,18	3
5	4	20	6	5	0	2,34	1
6	5	12	11	6	2	2,66	0
7	1	4	4	19	8	3,81	0

Analisando os resultados dos alunos em relação ao sistema usado anteriormente em FPRO, 31 dos 36 que responderam concordaram ou concordaram completamente que o sistema de pontos estabelecido para eliminar as piores provas práticas era simples de entender (questão 1). Já na questão que refere que é injusto apenas os estudantes com média de 85% às PP1 e PP2 poderem chegar a feiticeiro (questão 3) há diferentes opiniões sendo que apenas pouco mais de metade dos estudantes discordaram que era injusto. Apenas 3 concordaram que a autoavaliação não deveria

acrescentar pontos (questão 4) o que pode significar que os alunos gostam que o seu trabalho seja valorizado.

A maior parte considera que os níveis têm relevância na sua aprendizagem (questão 2), no entanto, parte considerável das respostas (11) é da opinião que os níveis não tiveram interferência. Vem-se a provar também que os níveis são relevantes no momento de escolher o tipo de avaliação (questão 6), sendo que os alunos mais empenhados podem optar por ter condições de avaliação diferentes da dos demais alunos.

Por fim, verificou-se que ao aplicar elementos de ludificação no ensino de programação, os alunos tendem a estudar com mais regularidade (questão 7).

As proposições colocadas na segunda parte do questionário foram:

1. Os diversos materiais de apoio e exercícios deveriam ser desbloqueados por cada estudante consoante o seu progresso e não de forma igual para todos
2. Um sistema de pontuação bem elaborado pode dar uma ideia clara da classificação final à disciplina
3. A avaliação contínua deveria contar mais, tendo em conta o mecanismo de ludificação
4. A participação nos fóruns para tirar dúvidas deveria também ter atribuição de pontos
5. Os estudantes que partilhem recursos adicionais de aprendizagem (vídeos e páginas Web) deveriam ter pontos adicionais
6. Gostaria de ter um conjunto de estatísticas que analisem a minha progressão na aprendizagem da unidade curricular
7. Gostaria de poder lançar desafios aos restantes estudantes
8. Gostaria de ter mais *quizzes* para testar os meus conhecimentos semanalmente
9. Gostaria de ter torneios no final de cada módulo que me possibilitassem competir com os outros estudantes

Na Tabela 6 são apresentados os resultados da segunda parte do questionário feito aos alunos, onde são feitas afirmações genéricas sobre possíveis melhorias ao sistema ludificado a que os alunos foram expostos anteriormente. Os dados são apresentados da mesma forma que foram apresentados na Tabela 5.

**Tabela 6 - Resultados da segunda parte do questionário realizado aos alunos**

Questão	1	2	3	4	5	Média	6
1	14	9	5	7	1	2,22	0
2	1	1	5	22	5	3,85	2
3	1	5	13	12	2	3,27	3
4	20	11	1	3	0	1,63	1
5	15	7	7	6	0	2,11	1
6	0	1	7	23	4	3,86	1
7	1	10	14	6	2	2,94	3
8	0	3	8	20	5	3,75	0
9	2	6	7	12	8	3,51	1

Os estudantes consideram que a avaliação contínua é importante e deveria ser mais valorizada tendo em conta os mecanismos de ludificação (questão 3) e que um sistema de pontos simples e inteligível pode ajuda-las a definir melhor a sua estratégia e percurso na unidade curricular (questão 2).

Mais de metade dos alunos discorda que os materiais de apoio e exercícios devam ser desbloqueados consoante a progressão do aluno (questão 1) o que faz crer que seja preferível o seu desbloqueio à medida que as aulas vão sendo lecionadas. Também a atribuição de pontos ao partilhar recursos adicionais de aprendizagem e na participação dos fóruns semanais de esclarecimento de dúvidas (questões 4 e 5) não foi bem recebida, devido à possibilidade dos alunos utilizarem estes espaços indevidamente para ganhar pontos mais facilmente.

Por outro lado, quase todos os alunos que responderam ao questionário acharam interessante a ideia do conjunto de estatísticas como forma de analisar a progressão dentro da UC (questão 6). Outro dos pontos que teve um reação bastante positiva por parte dos alunos foi a implementação dos *quizzes* semanais como forma de testar os seus conhecimentos teóricos (questão 8), tendo apenas 3 alunos discordado desta ideia.

A maior parte mostrou-se indiferente ou discordou da ideia de propor desafios a outros estudantes (questão 7) podendo significar que esta opção não seja muito relevante no percurso das suas aprendizagens na UC. Finalmente, verificou-se que a competição em torneios no final de cada módulo lecionado (questão 9) agrada aos alunos de uma forma geral, mas alguns deles não concordam com este tipo de desafios pois a competição pode ser prejudicial para a motivação dos alunos.

## 5.2 Questionários aos Especialistas

Para complementar, e como forma de obter uma opinião mais especializada sobre sistema ludificado desenvolvido, foi feita uma apresentação a 6 especialistas na área do ensino da programação que no final responderam a um questionário que visa entender os pontos fortes e fracos do sistema bem como os fatores que poderiam ser sujeitos a melhorias para tornar o sistema ainda envolvente e motivacional para os alunos aprenderem programação.

Os 6 especialistas são docentes da FEUP que lecionam unidades curriculares introdutórias de programação nos cursos de MM, MIEIC e MIEEC e têm entre 1 a 30 anos de experiência no ensino da área. As linguagens de programação ensinadas são o scheme (que é a linguagem que os alunos aprendem em FPRO), C, C++, Javascript, Matlab e Java.

Primeiramente foi feita uma pequena avaliação sobre a situação atual das disciplinas os especialistas lecionam e sobre o comportamento dos seus alunos em relação a elas. As afirmações apresentadas para serem avaliadas numa escala de Likert foram:

1. Os estudantes sentem-se motivados na UC
2. A documentação de apoio à UC é suficiente
3. Os estudantes estudam de forma regular semanalmente
4. As aulas teóricas têm um nível de participação elevado
5. As aulas teórico-práticas têm um nível de participação elevado
6. Os estudantes podem realizar percursos de aprendizagem/avaliação diferenciada consoante as suas competências

Na Tabela 7 são apresentados os resultados das respostas dadas pelos especialistas à situação atual das UC de programação que lecionam bem como a média dessas respostas.

**Tabela 7 - Resultados sobre a situação atual dos alunos nas unidades curriculares de programação que os especialistas lecionam**

Questão	1	2	3	4	5	Média	6
1	0	0	2	3	1	3,83	0
2	0	0	0	6	0	4,00	0
3	0	3	2	1	0	2,67	0
4	0	0	3	0	1	3,50	2
5	0	0	1	3	2	4,17	0
6	1	2	0	2	1	3,00	0

Ao analisar a situação atual é unânime que a documentação e material de apoio à UC é suficiente (questão 2), no entanto nenhum dos especialistas respondeu que concorda completamente com esta afirmação o que pode dar margem para melhorias da documentação atual.

Em relação à questão relacionada com a motivação dos estudantes na UC que lecionam (questão 1), 66,7% dos especialistas concorda que os estudantes estão motivados enquanto que os restantes não concordam nem discordam. No entanto, podemos verificar que, apesar de estarem motivados, pelo menos metade têm a opinião de que eles não estudam semanalmente (questão 3) o que pode significar que reservam o estudo apenas para as vésperas ou antevésperas de entrega de trabalhos ou da realização de testes.

Pode-se também verificar que as aulas teórico-práticas têm um interesse muito maior por parte dos alunos (questão 5) do que as aulas teóricas (questão 4), sendo as percentagens de concordância 83,3% e 16,7% respetivamente.

A última pergunta relacionada com a possibilidade de realizar percursos de aprendizagem e avaliação diferenciada consoante as competências dos alunos (questão 6) gerou uma opinião bastante diferenciada entre os especialistas sendo que metade concordou com a afirmação e a outra metade discordou. Este resultado pode-se explicar se considerarmos que os especialistas lecionam UC diferentes com características e modos de aprendizagem e avaliação distintos e que parte dos especialistas leciona a unidade curricular de FPRO, onde já foram implementados alguns elementos de ludificação anteriormente com o objetivo de individualizar os percursos dos alunos.

Posteriormente, com o objetivo de receber a opinião dos especialistas sobre o sistema apresentado presencialmente, foram apresentadas as seguintes questões para serem classificadas numa escala de Likert:

1. O sistema está bem desenvolvido
2. O sistema é complexo
3. Este sistema influencia positivamente o comportamento dos alunos para aprender programação
4. A atribuição de pontos em fóruns pode incentivar um uso excessivo e inadequado por parte dos alunos
5. A partilha de recursos adicionais (vídeos, artigos) de aprendizagem, pode ser um complemento para os alunos expandirem o seu conhecimento e entenderem melhor a matéria
6. A partilha destes recursos deveria atribuir pontos aos alunos
7. A possibilidade de sugestão de projetos de desenvolvimento pode facilitar a escolha de um aluno não tão criativo que decida optar por fazer um projeto
8. É interessante a possibilidade de alunos sugerirem exercícios para os outros tentarem solucionar
9. A curiosidade do aluno é suscitada se o desbloqueio de materiais de apoio à matéria e de exercícios for feito consoante a progressão do aluno

10. A disponibilização constante e atualizada de um conjunto de estatísticas do aluno na unidade curricular é importante para este receber feedback da sua progressão durante todo o semestre
11. O conceito de caderneta de cromos irá provocar maior motivação no aluno
12. A caderneta de cromos irá dar ao estudante uma maior sensação de progressão na aprendizagem
13. Os torneios podem fazer com que o aluno se motive a aprender mais
14. A competitividade dos torneios pode desmotivar o aluno

Os resultados das respostas da opinião dos especialistas em relação às características e elementos implementados no sistema podem ser consultados na Tabela 8 bem como o valor da média de cada questão.

**Tabela 8 - Resultados da opinião dos especialistas sobre o sistema ludificado**

Questão	1	2	3	4	5	Média	6
1	0	0	1	5	0	3,83	0
2	0	1	3	2	0	3,17	0
3	0	0	3	2	1	3,67	0
4	0	0	2	1	2	4,00	1
5	0	0	0	4	2	4,33	0
6	0	1	0	5	0	3,50	0
7	0	0	1	4	0	3,80	1
8	0	0	1	4	1	4,00	0
9	0	0	2	4	0	3,67	0
10	0	0	0	2	4	4,67	0
11	0	0	0	3	2	4,40	1
12	0	0	0	4	2	4,33	0
13	0	0	0	5	1	4,17	0
14	0	1	0	4	0	3,60	1

A opinião dos especialistas em relação ao sistema é que este está bem desenvolvido, mas que pode de certa forma estar um pouco complexo para ser explicado aos alunos na primeira aula do semestre (questão 1 e 2). Metade não concorda nem discorda que este sistema pode influenciar positivamente o comportamento dos alunos para aprender programação (questão 2), sendo que a outra metade responde afirmativamente a esta questão.

Todos os especialistas concordam que a partilha de recursos adicionais de aprendizagem pode complementar e expandir o conhecimento dos alunos (questão 5) e apenas um discordou na atribuição de pontos ao partilhar estes recursos (questão 6). Isto vai contra o que os alunos

responderam nos seus questionários quando lhe foi perguntado se concordavam com a atribuição de pontos ao partilhar destes tipos de conteúdo. Admite-se também que a atribuição de pontos em fóruns pode incentivar ao uso excessivo e inadequado destes por parte dos estudantes (questão 4).

Pode-se também apurar que a possibilidade de sugerir exercícios para os outros alunos tentarem solucionar é uma ideia interessante (questão 8) e que a sugestão de ideias de projetos por parte de outros alunos pode facilitar a escolha dos feiticeiros não tão criativos na hora de escolherem o tema do projeto que irão desenvolver (questão 7).

A realização de torneios no final de cada módulo pode tornar o aluno mais autónomo e motiva-lo a aprender mais (questão 13), no entanto, a pressão de tempo e o facto de os participantes poderem acabar a prova nos últimos lugares, pode fazer com que o aluno se sinta desmotivado ao se deparar com esta parte negativa da competição (questão 14).

A estatísticas que analisam a progressão do aluno na disciplina durante o semestre (questão 10) são das ideias mais apoiadas tanto pelos especialistas como pelos alunos, podendo-se concluir que a existência de um *feedback* constante sobre o que o desempenho dos alunos é um fator essencial a considerar e deve ser implementado nas edições seguintes da UC.

À questão que refere que o desbloqueio de conteúdos da matéria ao ser feito consoante a progressão do aluno pode suscitar a curiosidade no aluno, 66,67% concordou e os restantes não concordaram nem discordaram (questão 9).

Por fim, os especialistas são da opinião que o conceito de caderneta de cromos irá provocar uma maior motivação nos alunos (questão 11) na medida em que o aluno irá querer completá-la e para isso terá de participar nos desafios e atividades do *moodle*, o que fará ao mesmo tempo que este acabe sempre por aprender algo. Irá também dar uma maior sensação de progressão na aprendizagem (questão 12).

# 6. Conclusão e Trabalho Futuro

## 6.1 Conclusão

O propósito desta dissertação foi o de encontrar uma forma de incentivar e motivar os estudantes que entram para o ensino superior e que irão frequentar uma unidade curricular de programação a se interessarem e envolverem mais na disciplina. Ao mesmo tempo, entende-se que cada turma é heterogénea e as competências de cada um podem não estar ao mesmo nível no início da UC, então torna-se também importante arranjar uma forma de individualizar os percursos de aprendizagem e de avaliação dos alunos.

Este projeto foi focado na unidade curricular de Fundamentos de Programação do MIEIC na FEUP e para atingir os objetivos acima descritos fez-se uso da *gamification* – o uso de elementos de jogos em contextos não relacionados com jogos. Da pesquisa de *frameworks* que poderiam ser usadas para implementar a ludificação, duas delas destacaram-se e foram usadas para a criação do sistema ludificado: os 6 passos de Werbach e o Octalsys. A primeira é um processo passo a passo para implementar gamificação que pode ser generalizado para qualquer área de atuação sendo que os 6 passos passam pela definição dos objetivos, comportamentos-alvo dos utilizadores, os tipos de utilizadores que participam na ludificação, os ciclos de ações que se irão criar para manter o utilizador envolvido, a diversão que irá proporcionar e por fim as ferramentas que irão ser utilizadas para ajudar a alcançar os objetivos definidos no primeiro passo. Com a *framework* Octalsys foi possível analisar o sistema criado através de vários fatores como por exemplo a escassez, a impressibilidade, influência social, realização e *feedback* e ao atribuir pontos de 0 a 10 a cada um desses fatores, foi possível verificar os pontos fracos e fortes do sistema bem como perceber se o sistema fazia mais uso da parte esquerda ou direita do cérebro (se os alunos teriam de usar mais a lógica e racionalidade ou mais a parte criativa e emocional).

A intenção da criação deste sistema ludificado era de o implementar posteriormente na plataforma *moodle* para ser utilizado em edições futuras da UC. Então, apesar de não ter sido feita a implementação do sistema na plataforma devido a limitações de tempo, foi realizada uma pesquisa que culminou num guia passo-a-passo para a que futuramente seja possível uma implementação. Este guia descreve essencialmente como se pode adicionar os recursos e atividades e como se pode aplicar os elementos de ludificação tais como as restrições de acesso a

certas atividades até outras estarem completas, a implementação da caderneta de cromos, a atribuição de pontos ao concluir tarefas e as subidas de nível.

Foram feitos dois questionários após o desenvolvimento do sistema. O primeiro foi destinado aos alunos que frequentaram FPRO e estava dividido em duas partes, sendo que a primeira tinha a intenção de obter a opinião dos alunos em relação ao sistema de ludificação a que foram sujeitos quando frequentaram a disciplina e a segunda parte pretendia obter pistas sobre como melhorar ainda mais esse sistema. As principais conclusões retiradas deste questionário foram que a *gamification* realmente os motivou mais a trabalhar semanalmente; que se devia ter em atenção a questão da atribuição de pontos ao esclarecer dúvidas em fóruns e ao partilhar recursos de aprendizagem adicionais pois os alunos poderiam usar essas atividades excessivamente e inadequadamente como método para angariar pontos mais rapidamente; e a opinião foi quase unânime quando se fez referência sobre a implementação de um conjunto de estatísticas como forma dos alunos analisarem a sua progressão constante dentro da UC.

Já o segundo questionário foi destinado a especialistas na área do ensino da programação, onde primeiramente foi feita uma apresentação presencial para 6 especialistas com o intuito de explicar todos os detalhes do sistema e posteriormente responderam a um questionário que tinha como objetivo a validação do deste e também a possibilidade de sugestões de melhorias. Os resultados determinaram que o sistema criado estava bem desenvolvido e que a maior parte das características e elementos contidos nele eram interessante e poderiam motivar e envolver o aluno a aprender programação de uma maneira mais entusiasmante e divertida, fazendo também uma ressalva (tal como no questionários dos alunos) para o facto do possível uso abusivo dos fóruns para obter pontos mais rapidamente. Torna-se assim necessário encontrar soluções para controlar este problema.

No geral, com a obtenção dos resultados dos dois questionários, pensa-se que o sistema desenvolvido poderá proporcionar uma motivação maior no aluno para aprender programação e individualizará a aprendizagem e avaliações de cada um na disciplina, no entanto, para obter uma conclusão mais precisa será necessário fazer testes que comprovem que o sistema é realmente benéfico e trás bons resultados

## **6.2 Possíveis Aperfeiçoamentos ao Sistema Ludificado Desenvolvido**

Um produto, serviço ou sistema nunca chega a ser completamente perfeito, no entanto, a iteração entre os seus criadores e o público-alvo faz com que haja sempre um feedback e através deste pode-se ir melhorando aos poucos o sistema para ir de encontro as necessidades dos alunos e professores que o vão utilizar. Com isso em mente, e tendo em conta os questionários feitos aos

antigos alunos de FPRO e aos especialistas no ensino da programação, chegou-se à conclusão que algumas melhorias ainda poderiam ser feitas no sistema ludificado.

Um dos elementos a ter em conta é a atribuição de pontos nos fóruns de esclarecimento de dúvidas e na partilha de *links*. Estas duas atividades podem ser usadas abusivamente e inadequadamente pelos alunos para receberem pontos extra sem terem de fazer muito esforço, então ver-se-ia com bons olhos uma forma de controlar ou filtrar apenas as respostas que são úteis para esclarecer a dúvida de um aluno e avaliar se os conteúdos partilhados pelos alunos serão mesmo relevantes para um melhor entendimento sobre a matéria que estão a aprender na UC.

Outro ponto importante a considerar, e que foi sugerido por um especialista no questionário, é a realização de 2 ou 3 torneios ao mesmo tempo no final de cada módulo (um torneio para cada nível ou um torneio para os melhores alunos e outros para os alunos com mais dificuldades). Isto pode evitar a desmotivação dos alunos com pior desempenho ao ver que, mesmo não sendo dos melhores da turma, podem participar em todas as atividades.

Uma outra característica a ter em conta é os pontos extras. Alguns alunos poderão conseguir atingir nota máxima e a partir desse patamar terão pontos a mais que apenas servirão para serem acumulados ou, mesmo não atingindo a nota máxima, há pontos que se ganham mas que não contam para a nota final. Logo, vê-se a necessidade de dar uso a estes pontos, como por exemplo, distribuir os pontos extras pelos outros alunos, ou usar os pontos para eliminar perguntas no teste teórico.

Por fim, outros aspetos a ter em consideração são o bom esclarecimento das regras do sistema, tornar o sistema de pontuação o mais claro e simples possível e a criação de desafios mais colaborativos como trabalhos de grupo ou criação de *workshops* por parte dos alunos.

### **6.3 Trabalho Futuro**

Com esta proposta, os passos seguintes serão a implementação deste sistema no *moodle* e a realização posterior de um estudo de campo durante um semestre na unidade curricular de Fundamentos de Programação para avaliar se realmente este sistema trará resultados significativos em termos de motivação para aprender a programar, liberdade para aprender ao seu próprio ritmo e à sua própria maneira, e se isso se irá traduzir em melhores notas e resultados por parte dos alunos.



## 7. Referências

- [1] 06 de Fevereiro de 2017). Fonte: CODECADEMY: <https://www.codecademy.com/>
- [2] Werbach, K., & Hunter, D. (2012). *For the win: How game thinking can revolutionize your business*. Wharton Digital Press.
- [3] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments (pp. 9-15). ACM.
- [4] Hamari, J., Koivisto, J., & Sarsa, H. (2014, January). Does gamification work?--a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (pp. 3025-3034). IEEE.
- [5] Silva, T., Melo, J., & Tedesco, P. (2016, November). Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando gamification. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 5, No. 1, p. 71).
- [6] Regalado, M. R., da Silva, T. R., & da Silva Aranha, E. H. Um Mapeamento Sistemático sobre o uso da Gamificação em Ambientes Virtuais de Aprendizagem. *RENOTE*, 13(2).
- [7] Swacha, J., & Baszuro, P. (2013, July). Gamification-based e-learning platform for computer programming education. In *X World Conference on Computers in Education* (pp. 122-130).

- [8] Mora, A., Riera, D., Gonzalez, C., & Arnedo-Moreno, J. (2015, September). A literature review of gamification design frameworks. In *Games and Virtual Worlds for Serious Applications (VS-Games), 2015 7th International Conference on* (pp. 1-8). IEEE.
- [9] Chou, Y.-k. (1 de Março de 2015). *Octalysis – complete Gamification framework*. Fonte: Yu-kai Chou: Gamification & Behavioral Design: <http://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>
- [10] Claro, M. (26 de Setembro de 2014). *Badges no Moodle - Gamification*. Fonte: Moodle Livre: <https://www.moodlelivre.com.br/tutoriais-e-dicas/570-badges-no-moodle-gamification>
- [11] Claro, M. (18 de Julho de 2016). *Configurar Condicionais no Moodle*. Fonte: Moodle Livre: <https://www.moodlelivre.com.br/tutoriais-e-dicas/1606-configurar-condicionais-no-moodle>

## **8. Anexos**

# ANEXO A

2017-6-14

Ludificação do Ensino da Programação

## Ludificação do Ensino da Programação

O presente questionário, elaborado para uma dissertação do Mestrado em Multimédia da Universidade do Porto, visa recolher a opinião dos alunos que frequentaram a unidade curricular de Fundamentos de Programação acerca dos elementos de ludificação usados durante o período em que a disciplina foi lecionada no primeiro semestre. Tem também a intenção de dar pistas sobre como melhorar este sistema de forma a motivar mais os alunos na aprendizagem de programação e potenciar a individualização dos seus percursos (os alunos poderem ter mais poder de escolha) dentro da unidade curricular.

Desde já agradeço a ajuda para esta investigação!

**\*Obrigatório**

**1. Idade: \***

*Marcar apenas uma oval.*

- < 18  
 18 - 24  
 25 - 34  
 > 35

**2. Género \***

*Marcar apenas uma oval.*

- Feminino  
 Masculino

**3. Distrito: \***

\_\_\_\_\_

**4. Conselho:**

\_\_\_\_\_

**5. Estatuto: \***

*Marcar apenas uma oval.*

- Estudante  
 Trabalhador-estudante

**Avalie o grau de concordância ou discordância com as seguintes afirmações...**

**6. Relativamente ao sistema usado em FPRO neste ano letivo: \****Marcar apenas uma oval por linha.*

	Discordo completamente	Discordo	Não discordo nem Concordo	Concordo	Comcordo Completamente	Não sei/ Sem opinião
O sistema de pontos que foi estabelecido para poder anular as piores provas práticas é claro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os níveis de aprendiz, mágico e feiticeiro são relevantes para a minha aprendizagem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É injusto que só estudantes com média de 85% à PP1 e PP2 possam chegar a feiticeiro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A autoavaliação não deveria acrescentar pontos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Havia poucos desafios para ganhar pontos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Todos os estudantes deveriam ter as mesmas condições de avaliação independentemente do nível a que chegaram	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Esta estratégia motivou-me a trabalhar mais semanalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Avalie o grau de concordância ou discordância com as seguintes afirmações...**

**7. Pensando em futuras edições da unidade curricular, como acha que se poderia melhorar este sistema: \***

Marcar apenas uma oval por linha.

	Discordo completamente	Discordo	Não discordo nem Concordo	Concordo	Comcordo Completamente	Não sei/ Sem opinião
Os diversos materiais de apoio e exercícios deveriam ser desbloqueados por cada estudante consoante o seu progresso e não de forma igual para todos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Um sistema de pontuação bem elaborado pode dar uma ideia clara da classificação final à disciplina	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A avaliação contínua deveria contar mais, tendo em conta o mecanismo de ludificação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A participação nos fóruns para tirar dúvidas deveria também ter atribuição de pontos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os estudantes que partilhem recursos adicionais de aprendizagem (vídeos e páginas Web) deveriam ter pontos adicionais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gostaria de ter um conjunto de estatísticas que analisem a minha progressão na aprendizagem da unidade curricular	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gostaria de poder lançar desafios aos restantes estudantes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ludificação do Ensino da Programação

	Discordo completamente	Discordo	Não discordo nem Concordo	Concordo	Comcordo Completamente	Não sei/ Sem opinião
Gostaria de ter mais quizzes para testar os meus conhecimentos semanalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gostaria de ter torneios no final de cada módulo que me possibilitassem competir com os outros estudantes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**8. Sugestões de melhorias ao sistema que gostaria de ver implementado:**

---

---

---

---

---

# ANEXO B

2017-6-23

Introdução à Programação - Entrevista para teses de mestrado MIEIC/MM

## Introdução à Programação - Entrevista para teses de mestrado MIEIC/MM

Inquérito relativo à entrevista realizada no contexto de projetos para tese de mestrado dos alunos Luis Reis (MIEIC) e Jorge Monteiro (MM)

\*Obrigatório

### 1. Idade \*

Marcar apenas uma oval.

- < 25  
 25 - 34  
 35 - 44  
 45 - 54  
 > 54

### 2. Género \*

Marcar apenas uma oval.

- Masculino  
 Feminino

### 3. Categoria \*

---

### 4. Experiência no Ensino Superior (anos) \*

---

### 5. Experiência no ensino da Programação (anos) \*

---

### 6. Cursos que lecciona \*

Marcar tudo o que for aplicável.

- MIEIC  
 MIEEC  
 MM

### 7. Tipologia das aulas dadas \*

Marcar tudo o que for aplicável.

- Teórica  
 Teórico-Prática  
 Regência

**8. Linguagem que leciona \*****Situação Atual**

Considerando a situação atual das UC de Programação que leciona classifique as seguintes afirmações. Pode, se pretender, deixar uma sugestão para outros problemas existentes.

**9. Os estudantes sentem-se motivados na UC \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**10. A documentação de apoio à UC é suficiente \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**11. Os estudantes estudam de forma regular semanalmente \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**12. As aulas teóricas têm um nível de participação elevado \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**13. As aulas teórico-práticas têm um nível de participação elevado \***

Marcar apenas uma oval.

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**14. Os estudantes podem realizar percursos de aprendizagem/avaliação diferenciada consoante as suas competências \***

Marcar apenas uma oval.

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo Completamente
- Não sei / Não respondo

**15. Sugestões de outros problemas atuais (opcional)**

---

---

---

---

---

**Luis Reis**

Programa Híbrido para o ensino de programação e desenvolvimento de jogos.

**16. 1. O ensino paralelo de tópicos relacionados com desenvolvimento de jogos é uma mais valia para um programa de ensino de introdução à programação \***

Marcar apenas uma oval.

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo completamente
- Não sei / Não respondo

17. 2. Uma abordagem de Problem Based Learning (PBL), neste caso em que cada problema apresentado ao aluno é o desenvolvimento de um jogo, é apropriada para o ensino da programação \*

Marcar apenas uma oval.

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

18. As afirmações em 1. e 2. são mais corretas no contexto de cursos não ligados à área da informática \*

Marcar apenas uma oval.

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

19. As afirmações em 1. e 2. continuam corretas no contexto específico de cursos de informática e computação onde existirá um ensino mais formal e completo de programação \*

Marcar apenas uma oval.

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

## Lista de topicos

---

Mencionado na proxima pergunta.

1. Fundamentos da Programação
  - 1.1. Sintaxe e semântica de uma linguagem de programação
  - 1.2. Variáveis e tipos de dados primitivos (ex., números, caracteres, booleanos)
  - 1.3. Expressões e atribuição
  - 1.4. I/O básico, incluindo ficheiros
  - 1.5. Estruturas de controlo condicionais e iterativas
  - 1.6. Funções e passagem de parâmetros
  - 1.7. Recursividade
  - 1.8. Teste e correção de programas
2. Estruturas de Dados
  - 2.1. Vetores (arrays)
  - 2.2. Cadeias de caracteres (strings)
  - 2.3. Listas, Pilhas, filas e tabelas
  - 2.4. Abstrações de dados e sua implementação
  - 2.5. Seleção da estrutura de dados mais adequada
3. Projeto de Software

- 3.1. Conceitos básicos (Abstração, encapsulamento, acoplamento e reutilização);
- 3.2. Programação estruturada;
- 3.3. Abordagem top-down;
- 3.4. Programação baseada em abstrações.
4. Gráficos e Visualização
  - 4.1. Representação de imagem: modelo vetorial e raster;
  - 4.2. Cor e modelos de cor
  - 4.3. Desenvolvimento de aplicações gráficas
  - 4.4. Interfaces gráficas
5. Linguagens de Programação
  - 5.1. Programação Orientada por Objetos
  - 5.2. Programação baseada em eventos
6. Algoritmos e complexidade
  - 6.1. Complexidade algorítmica
  - 6.2. Diferenças no comportamento de um algoritmo - melhor/esperado/pior caso; Definição da notação O-grande;
  - 6.3. Complexidade temporal e espacial;
  - 6.4. Complexidade constante, linear, quadrática, logarítmica, e exponencial.
  - 6.5. Algoritmos de pesquisa (pesquisa sequencial e pesquisa binária);
  - 6.6. Comparação da complexidade algorítmica;
7. Fundamentos e design de Jogos Digitais
  - 7.1. Conceito de jogo e fatores humanos;
  - 7.2. Narrativas interativas e não lineares;
  - 7.3. Mecânicas de jogo;
  - 7.4. Design dos mundos de jogo;
8. Programação de Jogos Digitais
  - 8.1. Arquitetura de um motor de jogo
  - 8.2. Programação gráfica
  - 8.3. Programação de áudio
  - 8.4. Lógica de jogo
  - 8.5. Input e gestão de eventos
  - 8.6. Física e detecção de colisões
  - 8.7. Inteligência artificial

**20. A lista de tópicos apresentada está completa e inclui todos os conceitos que são importantes para um programa de introdução à programação \***

*Marcar apenas uma oval.*

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

**21. Se não, o que falta?**

\_\_\_\_\_

**22. Que linguagem(s) seria a mais indicada ou gostaria de lecionar se tivesse material de apoio?**

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

## Matriz de competencias

<https://paginas.fe.up.pt/~ei12085/diss/matriz/>

23. **A matriz de competências apresentada é um método eficaz de planejar os problemas a abordar classificando-os ao nível dos tópicos escolhidos \***

*Marcar apenas uma oval.*

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

24. **Espaço para comentários à pergunta anterior (opcional)**

---

---

---

---

---

## Tutorial

---

[https://paginas.fe.up.pt/~ei12085/diss/tutoriais/rock\\_paper\\_scissors.html](https://paginas.fe.up.pt/~ei12085/diss/tutoriais/rock_paper_scissors.html)

25. **O tutorial apresentado faz uma boa primeira introdução à linguagem Javascript e a um ambiente de trabalho sobre o qual futuros tutoriais possam construir? \***

*Marcar apenas uma oval.*

- Concordo completamente  
 Concordo  
 Não concordo nem discordo  
 Discordo  
 Discordo completamente  
 Não sei / Não respondo

26. **E em específico para cursos mais ou menos técnicos? + espaço para comentários (opcional)**

---

---

---

---

---

27. **Um ciclo de aulas com iterações compostas por Introdução -> Desenvolvimento de um Jogo -> Desafios (Onboarding -> Scaffolding -> EndGame) é adequado para um processo de introdução à programação \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo completamente
- Não sei / Não respondo

28. **Ter um percurso não linear de progressão com recursos diferenciados para cada estudante seria uma mais valia para os estudantes \***

*Marcar apenas uma oval.*

- Concordo completamente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo completamente
- Não sei / Não respondo

### **Jorge Monteiro**

Sistema ludificado para o ensino da programação.

**Avalie o grau de concordância ou discordância com as seguintes afirmações..**

---

29. \*

Marcar apenas uma oval por linha.

	Discordo completamente	Discordo	Não concordo nem discordo	Concordo	Concordo completamente	Não sei / Não respondo
O sistema está bem desenvolvido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O sistema é complexo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Este sistema influencia positivamente o comportamento dos alunos para aprender programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A atribuição de pontos em fóruns pode incentivar um uso excessivo e inadequado por parte dos alunos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A partilha de recursos adicionais (vídeos, artigos) de aprendizagem, pode ser um complemento para os alunos expandirem o seu conhecimento e entenderem melhor a matéria	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A partilha destes recursos deveria atribuir pontos aos alunos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A possibilidade de sugestão de projetos de desenvolvimento pode facilitar a escolha de um aluno não tão criativo que decida optar por fazer um projeto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É interessante a possibilidade de alunos sugerirem exercícios para os outros tentarem solucionar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Discordo completamente	Discordo	Não concordo nem discordo	Concordo	Concordo completamente	Não sei / Não respondo
A curiosidade do aluno é suscitada se o desbloqueio de materiais de apoio à matéria e de exercícios for feito consoante a progressão do aluno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A disponibilização constante e atualizada de um conjunto de estatísticas do aluno na unidade curricular é importante para este receber feedback da sua progressão durante todo o semestre	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O conceito de caderneta de cromos irá provocar maior motivação no aluno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A caderneta de cromos irá dar ao estudante uma maior sensação de progressão na aprendizagem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os torneios podem fazer com que o aluno se motive a aprender mais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A competitividade dos torneios pode desmotivar o aluno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**30. Sugestões de melhorias ao sistema que o poderiam tornar mais interessante:**

---



---



---



---



---

2017-6-23

Introdução à Programação - Entrevista para teses de mestrado MIEIC/MM

**31. Observações que não tenham sido abordadas anteriormente pelo questionário:**

---

---

---

---

---

---

Com tecnologia  
 Google Forms