



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

**INVESTIGAÇÃO SOBRE *gamification* PARA
AMENIZAÇÃO DAS DIFICULDADES NO
APRENDIZADO DE PROGRAMAÇÃO:
ENTREVISTA SEMI-ESTRUTURADA COM
ALUNOS DO CURSO DE ENGENHARIA DE
COMPUTAÇÃO DO CEFET-MG CAMPUS
TIMÓTEO**

IGOR OTONI RIPARDO DE ASSIS

Orientador: Aléssio Mirando Júnior
Instituição do orientador

TIMÓTEO
JUNHO DE 2019

IGOR OTONI RIPARDO DE ASSIS

**INVESTIGAÇÃO SOBRE *gamification* PARA
AMENIZAÇÃO DAS DIFICULDADES NO APRENDIZADO
DE PROGRAMAÇÃO: ENTREVISTA SEMI-ESTRUTURADA
COM ALUNOS DO CURSO DE ENGENHARIA DE COMPUTAÇÃO
DO CEFET-MG CAMPUS TIMÓTEO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Aléssio Mirando Júnior
Instituição do orientador

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
TIMÓTEO
JUNHO DE 2019

Igor Otoni Ripardo de Assis

**Investigação do Uso da Gamificação no Processo
de Aprendizagem de Programação**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia de
Computação do Centro Federal de Educação
Tecnológica de Minas Gerais, campus Timóteo,
como requisito parcial para obtenção do título de
Engenheiro de Computação.

Trabalho aprovado. Timóteo, 10 de dezembro de 2019



Prof. Me. Aléssio Miranda Júnior
Orientador

Prof. Me. Marlene Schettino
Professor Convidada

Prof. Dr. Viviane Cota Silva
Professor Convidada

Timóteo
2019



Emitido em 10/12/2019

FOLHA DE ROSTO (PLATAFORMA BRASIL) Nº 1/2019 - DCCTM (11.63.05)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 15/12/2020 19:19)

ALESSIO MIRANDA JUNIOR
PROFESSOR ENS BASICO TECN TECNOLOGICO
DCCTM (11.63.05)
Matrícula: 1713470

(Assinado digitalmente em 15/12/2020 20:03)

MARLENE SCHETTINO
PROFESSOR ENS BASICO TECN TECNOLOGICO
DCCTM (11.63.05)
Matrícula: 2144591

(Assinado digitalmente em 15/12/2020 19:36)

VIVIANE COTA SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DCCTM (11.63.05)
Matrícula: 2088439

Para verificar a autenticidade deste documento entre em <https://sig.cefetmg.br/documentos/> informando seu número:
1, ano: **2019**, tipo: **FOLHA DE ROSTO (PLATAFORMA BRASIL)**, data de emissão: **15/12/2020** e o código de
verificação: **5199d3ad16**

Agradecimentos

Primeiramente agradeço a meu orientador e professores da minha banca, por oferecem suas atenções para a produção deste trabalho.

Agradeço a todos os que foram meus professores e colegas de turma durante o curso de Engenharia de computação, tenho ciência de que todos contribuíram para a formação deste trabalho, seja direta ou indiretamente.

Também agradeço a meus familiares por toda estrutura e apoio durante esses anos.

Por fim agradeço a Deus pelo dom da criação e existência.

“Portanto, sede vós perfeitos, como perfeito é o vosso Pai celeste.” (Mt 5, 48)

Resumo

A evasão nas disciplinas de programação têm levantado necessidades de novas metodologias de ensino. A *gamification* possui berço nas empresas, mas têm sido aplicada a várias áreas, inclusive no ensino, promovendo motivação e engajamento. Após algumas dificuldades de aprendizado serem levantadas sobre as disciplinas de programação no curso de engenharia de computação do CEFET-MG Campus Timóteo, uma solução digital em *gamification* foi proposta e analisada neste trabalho, com objetivo de amenizar tais dificuldades. Têm se conhecimento que a *gamification* trabalha a parte cognitiva dos jogadores. Através da utilização do protótipo foi possível perceber que os elementos imersivos de *gamification* podem influenciar beneficemente no aprendizado do aluno; atendendo a vários perfis, classificando as habilidades desenvolvidas e mensurando o progresso do aluno. Assim como foi possível perceber que os elementos competitivos e dinâmicas de penalização podem atrapalhar ou inibir a participação do aluno. Há apreciação por parte dos alunos por tal estratégia, constatada pela aplicação de questionários e debates com os mesmos.

Palavras-chave: Gamification. Aprendizagem. Programação.

Lista de Figuras

Figura 1 – Domínio cognitivo da taxonomia de <i>bloom</i> e suas hierarquias.	6
Figura 2 – Analogia de Lister (2000) entre o ensino de programação e o domínio cognitivo da taxonomia de <i>bloom</i>	7
Figura 3 – Demonstração do enfoque da gamification por Deterding (2014).	13
Figura 4 – Visão do designer e do jogador pelo conceito de MDA de Hunicke et al. (2004).	14
Figura 5 – Demonstração do sistema de pontuação de prestígio social do <i>eBay</i> , para classificar a confiança em um vendedor por Zichermann et al. (2011).	15
Figura 6 – Demonstração da barra de progresso do <i>linkedIn</i> por Zichermann et al. (2011).	16
Figura 7 – Exemplo de tabela de ranque do jogo <i>Doodle Jump</i> por Zichermann et al. (2011).	17
Figura 8 – Sala de medalhas do <i>Foursquare</i> por Zichermann et al. (2011).	18
Figura 9 – Ciclo de engajamento social por Zichermann et al. (2011).	19
Figura 10 – Demonstração da relação entre conhecimento e <i>gamification</i> por Busarello (2016).	22
Figura 11 – Resultados das pesquisas nas bases científicas por Hamari, Koivisto e Sarsa (2014). A primeira coluna é o nome da base científica, a segunda é quantidade de trabalhos encontrados e a terceira a quantidade de trabalhos revisados.	23
Figura 12 – Framework para gamification de aprendizado por Caporarello, Magni e Pennarola (2017).	26
Figura 13 – Fluxo do desenvolvimento deste trabalho.	30
Figura 14 – Arquitetura da plataforma <i>Ludenos</i> , destacando funções da mesma e suas dinâmicas em interface com o aluno.	37
Figura 15 – Fluxo de atividade esperado pelo aluno na plataforma, destacando decisões que o mesmo pode tomar de acordo com seu progresso.	38
Figura 16 – Quantidade de respostas por opções da questão: Q1 - I1	41
Figura 17 – Quantidade de respostas por opções da questão: Q1 - I2	42
Figura 18 – Quantidade de respostas por opções da questão: Q1 - I3	42
Figura 19 – Quantidade de respostas por opções da questão: Q1 - I4	42
Figura 20 – Quantidade de respostas por opções da questão: Q1 - I5	43
Figura 21 – Quantidade de respostas por opções da questão: Q1 - I6	43
Figura 22 – Quantidade de respostas por opções da questão: Q1 - I7	43
Figura 23 – Quantidade de respostas por opções da questão: Q2 - I1	44

Figura 24 – Quantidade de respostas por opções da questão: Q2 - I2	44
Figura 25 – Quantidade de respostas por opções da questão: Q2 - I3	44
Figura 26 – Quantidade de respostas por opções da questão: Q2 - I4	45
Figura 27 – Quantidade de respostas por opções da questão: Q2 - I5	45
Figura 28 – Quantidade de respostas por opções da questão: Q2 - I6	45
Figura 29 – Quantidade de respostas por opções da questão: Q2 - I7	46
Figura 30 – Quantidade de respostas por opções da questão: Q2 - I8	46
Figura 31 – Quantidade de respostas por opções da questão: Q2 - I9	46
Figura 32 – Quantidade de respostas por opções da questão: Q2 - I10	47
Figura 33 – Quantidade de respostas por opções da questão: Q2 - I11	47
Figura 34 – Quantidade de respostas por opções da questão: Q2 - I12	47
Figura 35 – Quantidade de respostas por opções da questão: Q3 - I1	48
Figura 36 – Quantidade de respostas por opções da questão: Q3 - I2	48
Figura 37 – Quantidade de respostas por opções da questão: Q3 - I3	48
Figura 38 – Quantidade de respostas por opções da questão: Q3 - I4	49
Figura 39 – Quantidade de respostas por opções da questão: Q3 - I5	49
Figura 40 – Quantidade de respostas por opções da questão: Q4 - I1	49
Figura 41 – Quantidade de respostas por opções da questão: Q4 - I2	50
Figura 42 – Quantidade de respostas por opções da questão: Q4 - I3	50
Figura 43 – Quantidade de respostas por opções da questão: Q4 - I4	50
Figura 44 – Quantidade de respostas por opções da questão: Q4 - I5	51
Figura 45 – Quantidade de respostas por opções da questão: Q4 - I6	51
Figura 46 – Quantidade de respostas por opções do item: I1	57
Figura 47 – Quantidade de respostas por opções do item: I2	57
Figura 48 – Quantidade de respostas por opções do item: I3	57
Figura 49 – Quantidade de respostas por opções do item: I4	58
Figura 50 – Quantidade de respostas por opções do item: I5	58
Figura 51 – Quantidade de respostas por opções do item: I6	58
Figura 52 – Quantidade de respostas por opções do item: I7	59
Figura 53 – Quantidade de respostas por opções do item: I8	59
Figura 54 – Quantidade de respostas por opções do item: I9	59
Figura 55 – Quantidade de respostas por opções do item: I10	60
Figura 56 – Quantidade de respostas por opções do item: I11	60
Figura 57 – Quantidade de respostas por opções do item: I12	60
Figura 58 – Quantidade de respostas por opções do item: I13	61
Figura 59 – Quantidade de respostas por opções do item: I14	61
Figura 60 – Quantidade de respostas por opções do item: I15	61
Figura 61 – Quantidade de respostas por opções do item: I16	62
Figura 62 – Tabela de rank na plataforma.	71

Figura 63 – Pontuação de habilidades em programação na plataforma.	72
Figura 64 – Medalhas na plataforma.	73
Figura 65 – Feedback na plataforma.	74
Figura 66 – Estrutura de um curso na plataforma.	75
Figura 67 – Estrutura de um tópico na plataforma.	76
Figura 68 – Estrutura de um exercício na plataforma.	77
Figura 69 – Comentário de tópico (ou exercício) na plataforma.	77
Figura 70 – Solução de um exercício na plataforma.	78
Figura 71 – Barra de progresso na plataforma.	79
Figura 72 – Chat entre amigos na plataforma.	80

Lista de Tabelas

Tabela 1 – Estudo 1 - Conclusões	3
Tabela 2 – Classificação dos trabalhos por Hamari, Koivisto e Sarsa (2014). . .	85
Tabela 3 – Continuação da classificação dos trabalhos por Hamari, Koivisto e Sarsa (2014).	86

Lista de Abreviaturas e Siglas

CEFET	Centro Federal de Educação Tecnológica
IES	Instituto de Ensino Superior
SEMESP	Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior no Estado de São Paulo

Sumário

1 – Introdução	1
1.1 Objetivos	4
2 – Fundamentação Teórica	5
2.1 Aprendizado	5
2.2 Aprendizado em programação e suas dificuldades	6
2.3 Definindo gamification	11
2.4 Framework para entender o projeto de um jogo	13
2.5 Definido elementos ou mecânicas de gamification	14
2.5.1 Pontuação	14
2.5.2 Níveis	16
2.5.3 Tabela de ranque ou ranqueamento	17
2.5.4 Medalhas	18
2.5.5 Narrativa	18
2.5.6 Desafios e tarefas	19
2.5.7 Ciclos de enjamento social	19
2.5.8 Feedback	20
2.6 Impactos da gamification na psicologia	20
2.6.1 Área cognitiva	20
2.6.2 Área emocional	20
2.6.3 Área social	20
2.7 Gamification e o aprendizado/ensino	21
3 – Trabalhos Relacionados	23
3.1 Avaliação de gamification no aprendizado	27
4 – Procedimentos metodológicos	29
4.1 Ambiente de estudo: CEFET-MG Campus Timóteo	30
4.1.1 A disciplina escolhida	31
4.2 Questionário de verificação das dificuldades	31
4.2.1 Primeira seção do questionário	31
4.2.2 Segunda seção do questionário	32
4.2.3 Terceira e quarta seção do questionário	32
4.3 Implementação de uma plataforma com gamification	33
4.3.1 Elementos e dinâmicas de gamification utilizadas	34
4.3.2 Arquitetura da plataforma	35

4.3.3	Fluxo de atividades esperado do aluno na plataforma	36
4.4	Apresentação do protótipo e realização da entrevista semi-estruturada	39
4.5	Questionário qualitativo sobre o protótipo	39
4.5.1	Primeira seção	39
4.5.2	Segunda seção	39
5	– Análise e Discussão dos Resultados	40
5.1	Questionário sobre dificuldades no aprendizado de programação	40
5.2	Apresentação da plataforma os alunos	52
5.3	Questionário sobre dificuldades no aprendizado de programação	55
6	– Conclusão	63
6.1	Trabalhos Futuros	64
	Referências	66
	 Apêndices	 70
	 APÊNDICE A – Interfaces do Mundo do Código	 71
A.1	Rank	71
A.2	Pontuação de habilidades	72
A.3	Medalhas	73
A.4	Feedback	74
A.5	Estrutura de um curso	75
A.6	Tópico de um curso	76
A.7	Exercício de um tópico	77
A.8	Comentário de um tópico	77
A.9	Solução de um exercício	78
A.10	Barra de progresso do curso	79
A.11	Chat entre amigos	80
	 Anexos	 81
	 ANEXO A – Anexos da revisão bibliográfica de Hamari, Koivisto e Sarsa (2014)	 82
A.0.1	Referências dos trabalhos estudados	82
A.0.2	Tabelas com as classificações	84

1 Introdução

Têm-se verificado nas disciplinas de Linguagens e Lógica de programação uma alta taxa de reprovação ou evasão, em muitos casos já nos primeiros semestres dos cursos relacionados à tecnologia da informação e comunicação, nos diferentes níveis de ensino, [Deters \(2008\)](#). [Dehnadi e Bonart \(2006\)](#) afirmam que existe uma enorme taxa de reprovação nas disciplinas introdutórias de programação nas universidades britânicas. Relatam ainda que entre 30% e 60% dos alunos de ciências da computação reprovam na primeira disciplina de programação. [Lister \(2000\)](#) mostra que diversas universidades australianas apresentam taxas de reprovação em disciplinas introdutórias de programação como uma das piores taxas. [Deters \(2008\)](#) apresentam dados sobre o Instituto de Ensino Superior em que a taxa de desistência ou reprovação chega a 60% nas disciplinas introdutórias de programação. Segundo levantamento realizado em 2012 pelo Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior no Estado de São Paulo, [SEMESP \(2012\)](#), “a cada três alunos que entram no curso de sistemas de informação, apenas um recebe o diploma. Em Ciência da Computação, a cada quatro alunos que entram no curso, apenas um termina”.

Isso se deve as grandes dificuldades na compreensão e aplicação de conceitos básicos por parte dos alunos; [Gomes et al. \(2008\)](#), [Lahtinen et al. \(2005\)](#), [Deters \(2008\)](#) e [Lister \(2000\)](#) citam esse fato. [Lahtinen et al. \(2005\)](#), [Bennedsen e Caspersen \(2005\)](#) e [Bennedsen \(2008\)](#) afirmam que a dificuldade em lidar com a abstração e definições de componentes básicos de programação, é a pior enfrentada pelos alunos nas disciplinas fundamentais. [Lister \(2000\)](#) aponta que as metodologias adotadas pelas universidades nas disciplinas introdutórias de programação não estão coerentes com a taxonomia de bloom e muitos alunos investem mais tempo em aprender conceitos técnicos, como sintaxe das linguagens de programação e configurações de ambiente, investindo pouco tempo no desenvolvimento da lógica e habilidades de programação. [Gomes et al. \(2008\)](#) abordam essa dificuldade no processo de ensino de programação, tomando como ponto a análise de vários autores sobre o tema. Os mesmos indicam que a escolha de linguagem de programação ou uso de ferramentas audiovisuais não ajuda a resolver o problema como um todo, e explicam que isso acontece, pois muitas dúvidas introdutórias nem chegam a ser contempladas nas ferramentas, além disso, cada aluno possui um perfil de aprendizado diferente, sendo favorecido ou desfavorecido por tais escolhas. Os autores concluem que uma metodologia ou ferramenta que motive o aluno e o acompanhe durante todo o processo de aprendizado é a melhor proposta. [Deters \(2008\)](#) também reforçam a necessidade de motivar o aluno, além da utilização das ferramentas e metodologias. [Lahtinen et al. \(2005\)](#) aplicaram questionários a alunos e professores, investigando dificuldades no processo de

ensino/aprendizagem de programação, e concluíram que questões técnicas como escolha de linguagem de programação não mudam as dificuldades sentidas pelos alunos. Perceberam que quanto mais for abstrato um tema de programação, maior é a dificuldade sentida pelos alunos em aprender o mesmo. Rocha et al. (2010) afirmam que independentemente da metodologia utilizada, lógica de programação é requisito fundamental nos cursos de computação, sendo base no desenvolvimento de raciocínio lógico e construção de algoritmos corretos.

A utilização de jogos como meio para motivar e engajar públicos é uma técnica recorrente e de sucesso. Segundo McGonigal (2011) os jogos conseguem fazer com que seus usuários permaneçam em uma tarefa por um longo período de tempo. Lazzaro (2004) afirma que as pessoas jogam tanto pela atividade do jogo quanto pelas experiências que ele proporciona. Buscando essa motivação e engajamento para outros serviços e atividades, através da utilização de elementos, estéticas e mecânicas de jogos, surgiu a *gamification*. Deterding (2014) define como *gamification* o uso de elementos de jogos em contextos não relacionados com os mesmos. Utilizando de tabelas de ranque, esquemas de pontuação, atribuição de níveis e medalhas, feedback aprimorado, narrativas diversas e ciclos de socialização, a *gamification* consegue situar o desempenho e habilidades dos seus jogadores, além de motivar, divertir e acompanhar os mesmos em toda a duração do jogo, Busarrello (2016).

Hamari, Koivisto e Sarsa (2014) constataram que o ensino é alvo das aplicações de *gamification*, e através de uma revisão literária, conseguiram concluir que o seu uso motivou positivamente os alunos. Majuri, Koivisto e Hamari (2018) confirmaram tal conclusão, apresentando uma revisão literária de trabalhos que usaram a *gamification* no ensino, mas que apresentaram dados empíricos. Foi identificado que poucos trabalhos têm mensurado o impacto da *gamification* no processo de aprendizagem; estes estão mais voltados para avaliação de motivação e engajamento.

Silva (2018) realizou uma investigação envolvendo disciplinas de programação do curso de engenharia do Centro Federal de Educação Tecnológica de Minas Gerais, com o intuito de descobrir o perfil de aprendizado dos alunos de programação, de acordo com o método *Index of Learning Styles*, proposto por Felder e Silverman (1988). Para isso foram elaborados questionários que envolviam lógica de programação. Após aplicar os questionários e realizar a classificação de perfil de aprendizado dos alunos, o autor conclui:

... observou-se diversas dificuldades voltadas à resolução de problemas e muitas dificuldades em resolver questões de cunho lógico-matemático. Estes problemas são descritos como dificuldades de interpretação de texto, falta de habilidade em tomar decisão, falta de habilidades voltadas à abstração, dificuldade em tratar lógica e raciocínio. Por dificuldades matemáticas específicas entende-se: falta de domínio de cálculos em geral, problemas em lidar com interpretação de fórmulas matemáticas e fundamentalmente problemas relacionados a Teoria de Números, ou seja, divisores comuns,

números primos, múltiplos e outros. (Silva , 2018, p. 36)

A tabela 1 foi retirada da monografia do autor e especifica as dificuldades dos alunos ao qual responderam o questionário.

Tabela 1 – Estudo 1 - Conclusões

Resolução de Problemas	Os alunos tinham dificuldades de interpretação do enunciado, logo tendo falta de habilidade em resolver os problemas Incapacidade em utilizar conceitos abstratos ou dificuldade em entendê-los. Os alunos não dominavam conceitos matemáticos básicos e dificuldades em realizar cálculos, além do uso indevido da Teoria dos Números. Apesar da dificuldade com os cálculos, tinham habilidade de responderem questões com conceitos geométricos.
------------------------	---

Fonte: Adaptado de (Silva , 2018, p. 40)

Essas dificuldades podem ter como causa diferentes fatores. Como mostra Castro (2018), um forte fator pode ser a diferença de características entre gerações; a geração Z, que compreende todos os que nasceram a partir do ano 2000 D.C., conhece o mundo sobre o olhar digital, enquanto a geração anterior estava naturalizada com o ambiente analógico. As diferenças tecnológicas contribuíram para um abismo de comunicação entre as gerações, que afetou o aprendizado. A autora apontou a *gamification* como uma estratégia para superar tal abismo no aprendizado. Castro (2018) realizou um *quiz* elementos de *gamification*. Este *quiz* foi aplicado a alunos do ensino técnico de informática do CEFET-MG Timóteo e, através de um questionário avaliativo aplicado aos mesmos, ela concluiu:

A pesquisa [...], mostrou que mais de 70% dos alunos tiveram um maior interesse em realizar uma atividade contendo elementos da *gamification* e 54,5% declararam que se as matérias em sala de aula fossem dadas em forma de atividades alvo de *gamification* eles teriam um melhor aprendizado. (Castro , 2018, p. 57)

Diante das dificuldades de aprendizado relatadas por Silva (2018) e pelo engajamento notado por Castro (2018) com as técnicas de *gamification* diante dos alunos do campus CEFET-MG Timóteo, este trabalho terá como objetivo conciliar o uso de técnicas de *gamification* no aprendizado de programação como uma possível solução para superação das dificuldades dos alunos. A solução será contemplada em uma plataforma, como um protótipo funcional, que será avaliado por alunos que cursam a disciplina de Algoritmos e Estrutura de Dados II do Campus CEFET-MG Timóteo em 2019, no segundo semestre. Para tal avaliação, será disponibilizado materiais e exercícios sobre grafos para os alunos, por um período de um mês. A turma escolhida já possui experiência em utilizar plataformas de submissão de algoritmos, utilizadas em maratonas de programação; esse ambiente será reaproveitado. Os elementos de jogos utilizados serão os propostos por Zichermann et

al. (2011), respeitando as definições de gamification propostas por Deterding (2014), e correlacionados ao ensino/aprendizado como foi explicado por Busarrelo (2016). A pergunta de pesquisa portanto é: como uso de técnicas de *gamification* no ensino de programação pode auxiliar na superação das dificuldades de aprendizagem?

1.1 Objetivos

O objetivo geral deste trabalho é: conciliar e avaliar o uso de elementos de *gamification* para o processo de aprendizado de programação.

Objetivos específicos:

- Identificar dificuldades no processo de aprendizado voltadas a programação na literatura e constatar a existência de tais dificuldades na turma do CEFET através de um questionário;
- Identificar quais técnicas de *gamification* podem ser usadas para superar as dificuldades analisadas;
- Projetar uma arquitetura de plataforma que utilize tais técnicas e implementar um protótipo;
- Validar o protótipo mediante apresentação aos alunos, disponibilizando o uso do mesmo em sala de aula e coletar a experiência dos alunos com a utilização da plataforma por meio de uma entrevista semi-estruturada e um questionário.

2 Fundamentação Teórica

2.1 Aprendizado

O aprendizado é um processo estudado pela área da psicologia, muitos autores tentaram conceituar tal processo. Realizando uma revisão bibliográfica, procurando explicitar o que é o aprendizado, [Signori e Guimarães \(2016\)](#) concluíram:

... a aprendizagem pode ser caracterizada como um processo de desenvolvimento intelectual, que ocorre por meio das estruturas de pensamento do sujeito e a sua interação com determinado meio. Assim, pode-se reconhecer três tipos de aprendizagem: a cognitiva, que resulta no armazenamento organizado de informações na mente do ser que aprende; a afetiva, resultado de sinais internos ao indivíduo e é identificada com experiências de dualidade como satisfação ou descontentamento, alegria ou ansiedade; e a psicomotora, que envolve respostas musculares adquiridas por meio de treino e prática.

Os autores também relatam, que embora sejam submetidos as mesmas condições, a literatura reconhece que cada indivíduo absorve o conhecimento em níveis diferentes. A taxonomia de *bloom* é considerada um instrumento que possibilita organizar e classificar tais níveis. A mesma é constituída por três domínios:

1. cognitivo;
2. afetivo;
3. psicomotor.

... as particularidades necessárias de cada domínio:

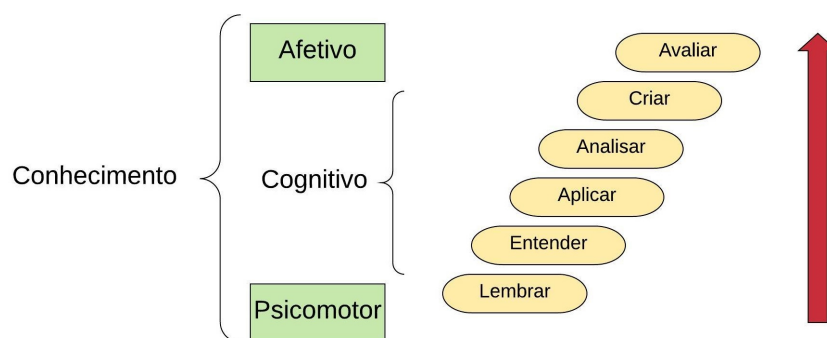
1. cognitivo: consiste em dominar um conhecimento relacionado ao aprender. Este domínio está conectado ao desenvolvimento intelectual, de habilidades e atitudes, ou seja, envolve a aquisição de um novo conhecimento. Reconhecimento de fatos específicos, procedimentos padrões e conceitos que estimulem o avanço constante do intelecto são fatores reconhecidos. Neste domínio existem 6 hierarquias de complexidade e dependência, em ordem crescente de complexidade. São elas:
 - a) lembrar;
 - b) entender;
 - c) aplicar;
 - d) analisar;
 - e) criar;
 - f) avaliar;
2. afetivo: consiste em sentimentos e posturas. As categorias identificadas correspondem a desenvolvimento da área emocional e afetiva, comportamento, atitude, responsabilidade, emoção, respeito e valores. As categorias correspondem:
 - a) receptividade;
 - b) resposta;
 - c) valorização;
 - d) organização ou conceituação de valores;

- e) internalização de valores;
- 3. psicomotor: consiste nas habilidades físicas específicas, que incluem:
 - a) imitação;
 - b) manipulação;
 - c) precisão;
 - d) habilidades articulação;
 - e) naturalização.

(Signori e Guimarães , 2016, p. 3)

A figura 1 exemplifica o domínio cognitivo e suas hierarquias.

Figura 1 – Domínio cognitivo da taxonomia de *bloom* e suas hierarquias.



Fonte: o próprio autor.

Além dos diferentes níveis de absorção, cada indivíduo também possui um perfil de aprendizado. Estes perfis reúnem diversas características comportamentais comuns a indivíduos no processo de aprendizagem. Vários autores conceituaram grupos de perfis. Neste trabalho não será abordado profundamente tal tema, pois a solução projetada objetiva atingir todos os perfis de alunos, não somente os perfis da turma em que foi avaliada, disponibilizando materiais de estudos de variadas formas e tipos: texto, imagem e vídeos.

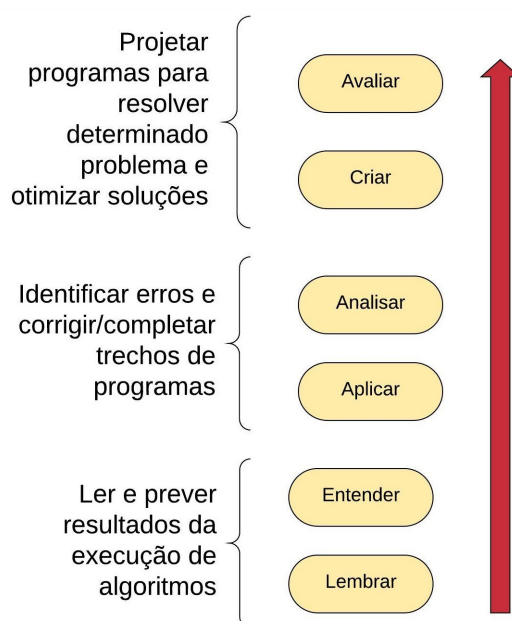
2.2 Aprendizado em programação e suas dificuldades

O aprendizado em programação encontra dificuldades em muitas universidades, compartilhando o mesmo problema do CEFET-MG. Realizando investigações sobre tais dificuldades, Lister (2000) chama atenção para a desmotivação e plágio que acarretam os alunos nas matérias introdutórias de programação. O autor afirma que o ensino de programação nas universidades não respeita a sequência da taxonomia de *bloom*; o aluno é forçado a absorver conceitos tecnológicos, que estão associadas aos últimos dois níveis da taxonomia, e o aprendizado depende do desenvolvimento de níveis anteriores, pois é neles que as habilidades em programação são desenvolvidas.

O autor apresenta uma versão modificada da taxonomia de *bloom*, exemplificada na figura 2, voltada para o aprendizado em programação. Nessa abordagem os dois primeiros

níveis enfatizam habilidades de ler e compreender códigos, os dois níveis intermediários enfatizam habilidades de escrever pequenos trechos de códigos para contextos bem definidos, e os dois últimos níveis enfatizam habilidades para desenvolver programas completos não triviais.

Figura 2 – Analogia de Lister (2000) entre o ensino de programação e o domínio cognitivo da taxonomia de bloom.



Fonte: o próprio autor.

Lister (2000) também aponta a opção por questões de múltipla escolha, em que trechos de código são exibidos no enunciado e as alternativas para resposta indicam o resultado da execução de tais trechos pelo computador. São exemplos de exercícios para os dois primeiros níveis. Para os níveis intermediários, também sugere questões de múltipla escolha onde os enunciados exibem parte de um código e as alternativas indicam trechos de código que se unidos ao código do enunciado, representarão um programa para determinado objetivo. Tal objetivo deve ser explicitado no enunciado. Também é sugerido a utilização de “algoritmos esqueletos”: que um trecho de código é apresentado para cumprir determinada função, porém está incompleto ou contém erros. O aluno deve corrigir ou acrescentar linhas de código ao mesmo, evitando assim tempo de escrita em que nenhum conhecimento novo é testado.

Lister (2000) deixa bem claro que a preocupação com o ensino não deve estar voltada para fatores tecnológicos, o que é reforçado por Dehnadi e Bonart (2006), estes também investigando soluções para as dificuldades no ensino, concluíram que alterar a linguagem de programação, paradigmas e ferramentas para o desenvolvimento têm pouco impacto no ensino. Estes autores demonstram índices de reprovação nas universidades britânicas entre 30% e 60%.

Essas dificuldades estão a nível mundial, como ressaltam [Gomes et al. \(2008\)](#). E há falta de metodologias para reverter a situação de reprovação nas disciplinas. Realizando uma revisão sobre trabalhos que estudam os problemas envolvidos no ensino de programação, o autores apontaram como grande dificuldade a compreensão e aplicação de noções básicas de lógica e que a mudança de ferramentas não muda o cenário de reprovação e evasão, fortalecendo demais citações. Os conceitos em programação são abstratos e fazem os alunos perderem o interesse. Nas disciplinas introdutórias os alunos ficam mais preocupados com questões tecnológicas como sintaxes e configurações de ambientes, não focando em desenvolver habilidades lógicas. Existem, também, fatores de ordem emocional e de transição que afetam os alunos no início de curso, que impactam no aprendizado das disciplinas introdutórias de programação, mas a falta de motivação é mais preocupante na literatura, afirmam [Gomes et al. \(2008\)](#).

Mencionam ainda, a falta de uma metodologia centrada no aluno, que promova ao mesmo supervisão e *feedbacks* adequados. Criticam a excessiva preocupação em ensinar sintaxes e familiaridades tecnológicas no início do curso, o que deveria ser ensinado somente depois que o aluno tenha desenvolvido habilidades de programação. Os autores defendem que para aprender a programar é necessário atividades práticas. Apontam dificuldades dos alunos em interpretar textos problemas e defasagem de conhecimento em outras áreas, que atrapalham desenvolver soluções em algoritmos, assim como [Silva \(2018\)](#) percebeu em seu trabalho. Também afirmaram que os alunos não refletem ou testam suas soluções produzidas de maneira coerente. Concluíram que nenhuma ferramenta de ensino de programação obteve sucesso, pois nenhuma até então incluiu uma vasta quantidade de perfis de aprendizado, para que envolvessem todos os alunos. [Gomes et al. \(2008\)](#) citaram algumas diretrizes que qualquer metodologia de ensino para programação ou ferramenta deve incluir:

- Contribuir para o aumento da motivação do aluno, através de um ambiente lúdico e estimulante.
- Animações e modelos dinâmicos que representem os vários conceitos de programação.
- Apresentar atividades de acordo com o estilo de aprendizagem de cada aluno, de acordo com o seu ritmo.
- Oferecer uma abordagem gradual de apresentação da programação, treinando, em primeiro lugar, competências básicas de resolução de problemas, de planejamento de soluções e mostrando a utilidade da programação, deixando para mais tarde os detalhes sintáticos das linguagens de programação
- Minimizar, os complexos detalhes sintáticos de uma linguagem de programação.
- Utilizar metáforas e exemplos concretos conhecidos do aluno para ajudar a diminuir a carga abstrata inerentemente associada à programação.
- Preparar um ambiente estruturado que certifique metodologias de estudo corretas

que levem à reflexão e questionamento recorrentes.

- Permitir um treino intensivo de resolução de problemas, forçando o aluno a seguir todas as etapas para uma correta resolução de problemas, que se resumem à compreensão do problema, caracterização do problema, representação do problema, solução do problema e reflexão sobre a solução obtida.
- Permitir um treino intensivo de conhecimentos matemáticos e lógicos úteis à programação.
- Possibilitar que os alunos ganhem uma certa experiência em programação, dando-lhe sugestões para atingir determinadas soluções, propondo atividades diversificadas (programas completos para os alunos analisarem, programas que contenham erros lógicos que foram cometidos pelos alunos também e programas incompletos para finalizar), apresentando modelos de forma a que os alunos adquiram as melhores práticas de programação.

Ainda investigando tais dificuldades, porém focando no caso de alunos repetentes nas disciplinas introdutórias do IES - Instituto de Ensino Superior, [Deters \(2008\)](#) descobriram três problemas gerais:

1. o elevado número de alunos por turma, que inviabiliza a realização de um acompanhamento individualizado;
2. avaliações que ocorrem apenas por meio de provas escritas ou trabalhos individuais, não promovendo uma evolução gradual da aprendizagem;
3. heterogeneidade da turma, disparidade de conhecimento e ritmo de aprendizagem.

Também descobriram dificuldades enfrentadas pelos professores em:

1. reconhecer habilidades inatas de seus alunos;
2. apresentar técnicas de resolução de problemas;
3. promover o desenvolvimento da capacidade de abstração do aluno, permitindo-o selecionar as estruturas de dados coerentes;
4. facilitar a cooperação e colaboração entre os alunos.

Os autores também enumeraram causas da desmotivação dos alunos:

1. a preocupação excessiva com detalhes de sintaxe da linguagem sendo usada;
2. a falta de uma visão sobre o que se quer solucionar, como idealizar soluções adequadas, e consequentemente mapear essas soluções em passos sequenciais e abstrair o funcionamento dos mecanismos escolhidos;
3. o estabelecimento de um raciocínio lógico visando à resolução de problemas, com base em um modelo incremental, em relação à complexidade e à estratégia de refinamentos sucessivos.

Com dados do IES, os autores mostram que a taxa de reprovação ou evasão pode chegar a 60% nas disciplinas de introdução a programação, em concordância com [Dehnadi](#)

e Bonart (2006). Levantaram alguns motivos que podem causar essa taxa:

- alunos estudam e trabalham, não se dedicando como deveria aos estudos;
- desmotivação dos alunos em resolver listas de exercício e ler materiais;
- dificuldades com interpretação dos exercícios, impossibilitando ou levando a solução errada;
- dificuldade na assimilação dos conceitos, agravada por defasagem de conhecimento em outras áreas;

Deters (2008) também destacam alguns objetivos que podem diminuir as dificuldades:

- Estimular o aluno a desenvolver as suas habilidades durante o período da disciplina.
- Promover ao aluno a visualização de seu progresso na capacidade de resolução de problemas.
- Oferecer aos alunos desafios práticos.

Por fim, concluem que:

Entretanto, percebe-se a necessidade de motivar o aluno repetente, sendo desenvolvidas algumas estratégias para estimular o seu aprendizado e a sua consciência como responsável pelo aprendizado, convertendo uma situação de fracasso em uma situação de sucesso.(Deters , 2008, p. 2)

Ainda diante das dificuldades, Lahtinen et al. (2005) realizaram uma pesquisa para classificar o quanto as mesmas eram sentidas pelos alunos e professores, e como os mesmo lidavam com elas. Aplicaram um questionário em 6 universidades, totalizando 559 alunos e 34 professores. Para cada pergunta do questionário, várias respostas eram apresentadas e cada uma podia variar de uma nota entre 1 a 5. O total de questões eram 4, seus temas eram:

- Quais atividades mais difíceis durante o aprendizado
- Quais conceitos são os mais difíceis de aprender
- Qual maneira mais confiante para se aprender
- Quais materiais mais ajudam a aprender

Analisando as respostas do questionário foi possível aos autores perceber que as atividades mais classificadas pelos alunos como grande dificuldade foram:

- Entender como projetar um programa para resolver uma determinada tarefa
- Dividir funcionalidades em procedimentos
- Achar erros nos próprios algoritmos

Essas atividades requerem um conhecimento sobre o programa como um todo, não somente sobre detalhes do mesmo. Os conceitos que os alunos mais classificaram como

difíceis foram:

- Recursão
- Ponteiros e referências
- Tipos abstratos de dados
- Tratamento de erros
- Uso de bibliotecas

Maioria destes conceitos são abstratos e não estão presentes na vida cotidiana, sendo difícil realizar comparações. De acordo com os autores as respostas dos professores foram coerentes com as respostas dos alunos. Porém os professores acrescentaram uma atividade: entender estruturas de programação. Quanto as maneiras, alunos também preferem estudar sozinhos, em vez de realizar leituras, e preferem programar por conta própria em vez de serem guiados por listas de exercícios ou seções práticas. O material mais apreciado pelos alunos eram códigos fontes ou exemplos de programas. Os professores escolheram as listas de exercícios e seções práticas como maneiras para estudo, e preferiram visualizações interativas como materiais. Os autores concluíram que os alunos gostam de mais independência ao aprender, enquanto os professores preferem guiar e supervisionar mais o aprendizado dos alunos.

Antes da aplicação do questionário os autores haviam perguntado aos alunos qual a linguagem que estes usavam para programar, a grande maioria foi C e JAVA, porém com as análises do questionário, a mudança de linguagem não afeta as dificuldades no aprendizado, corroborando com todos autores citados até aqui. Quase metade dos alunos já tinham experiência com programação antes do curso, o que também não alterou as dificuldades sentidas, ou seja, há um erro na metodologia de ensino.

2.3 Definindo gamification

Para entender a *gamification*, que será a estratégia usada para superar as dificuldades de aprendizado em programação neste trabalho, é necessário entender o que é um jogo. Há muita divergência sobre a definição de jogo, mas Kelley (1988) definiu um jogo de uma maneira interessante para este trabalho:

Um jogo é uma forma de recreação constituída sob um conjunto de regras que especifiquem a busca por um objetivo e os meios de atingi-lo. (Kelley, 1988, p. 40)

Percebe se por essa definição que um jogo é uma realidade abstrata, uma categoria, possui assim características; regras, objetivos, narrativa, etc. Em uma reflexão mais filosófica, Huizinga (2000) afirma que um jogo só pode ser reconhecido por suas características:

Devemos, portanto, limitar-nos ao seguinte: o jogo é uma função da vida, mas não é passível de definição exata em termos lógicos, biológicos ou estéticos. O conceito de jogo deve permanecer distinto de todas as outras formas de pensamento através das quais exprimimos a estrutura da vida espiritual e social. Teremos, portanto, de limitar-nos a descrever suas principais características. (Huizinga , 2000, p. 9)

E é justamente pela definição dessas características ou elementos de um jogo que a **gamification** fará sentido. Diante disso, Deterding (2014) define como **gamification** o uso de elementos de jogos em contextos não relacionados com os mesmos. Assim a *gamification* proporciona um jogo baseado em estratégia, estruturado por regras e competição, não apenas uma brincadeira em que respostas e comportamentos podem ser improvisados. A técnica foi muito utilizada por empresas, inicialmente, com o intuito de motivar e engajar determinados públicos. Ganhou destaque, pois se demonstrou como uma maneira eficiente de superar as diferenças entre seus participantes, ou seja, os jogadores. Dada a simplicidade e diversão proporcionada pelos jogos, estes foram estudados, com o objetivo de se retirar elementos que podem ser incorporados a outras atividades, objetivando o engajamento. A *gamification* não é restrita a sistemas digitais; qualquer atividade ou serviço pode ser contexto de uma estratégia de *gamification*.

O autor ainda define que elementos de jogos não são apenas os elementos necessários para construir os jogos, mas também não são quaisquer componentes que um jogo específico apresente, mas tudo o que caracteriza um jogo, presente, não necessariamente, em maioria dos jogos. Lembrando que, por jogos, entende-se aqui qualquer atividade ou serviço que busque cumprir metas ou objetivos através da motivação dos indivíduos, de maneira divertida e competitiva, seja digital ou não. É reconhecido que para tais definições utiliza-se muito de heurística, e mais estudos são necessários para a consolidação das definições de jogos, características ou elementos de jogos e *gamification*. Seguindo Deterding (2014), o autor define cinco níveis de projeto para a *gamification* em um determinado contexto. São eles:

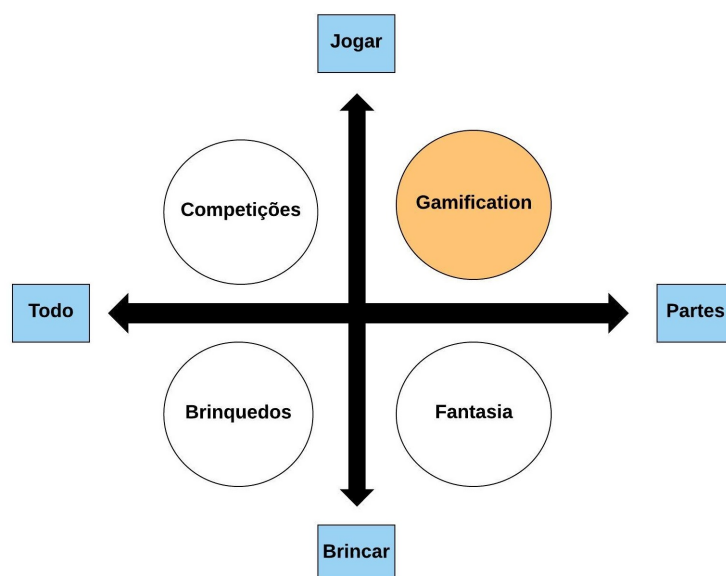
- Padrões de projeto de interfaces: esse nível define o valor que o jogador pode adquirir no jogo, exemplificados pelos elementos de atribuição de medalhas, ranqueamentos e nivelção.
- Padrões de projeto da mecânica do jogo: são as limitações do jogo, como por exemplo, limitações de tempo, de recursos ou rodadas.
- Padrões de projeto para os princípios ou heurísticas do jogo: define as metas ou objetivos do jogo, assim como estilos ou tipos permitidos para jogar.
- Modelos conceituais de projeto unitários do jogo: um nível abstrato que define quais comportamentos e respostas psicológicas o jogo que causar.
- Padrões de metodologia e processos do jogo: define a estrutura do jogo em si, correlacionando os outros níveis.

Expondo quatro sentidos para a produção de um jogo:

- “Partes” versus “Todo”
- “Jogar” versus “Brincar”

[Deterding \(2014\)](#) consegue definir que o motivo que caracteriza a *gamification* é o foco nas “Partes” e no “Jogar”. Brinquedos, por exemplo, estariam entre Todo e Brincar, comportamentos lúdicos, como dinâmicas e fantasias estariam entre Parte e Brincar, e disputas ou competições estariam entre Jogar e Todo. A figura 45 exemplifica esses sentidos.

Figura 3 – Demonstração do enfoque da gamification por [Deterding \(2014\)](#).



Fonte: adaptado de [Deterding , 2014](#), p. 5

2.4 Framework para entender o projeto de um jogo

Um processo para construção de jogos já existe; [Hunicke et al. \(2004\)](#) propuseram um *framework* para explicar e orientar o projeto de jogos. O *framework*, denominado MDA (*Mechanics Dynamics Aesthetics*), é uma abordagem formal para entender jogos. Para isso, a abordagem quebra o conceito de jogo em três conceitos.

- Mecânicas: descreve os componentes específicos do jogo, no nível da representação de dados e algoritmos.
- Dinâmicas: descreve o comportamento em tempo de execução da mecânica que atua nas entradas do jogador e nas saídas uns dos outros ao longo do tempo.
- Estéticas: descreve as respostas emocionais desejáveis evocadas no jogador, quando ele interage com o sistema de jogo.

Do ponto de vista do projetista, a mecânica dá origem ao comportamento dinâmico do sistema, que por sua vez leva a experiências estéticas particulares. Do ponto de vista

do jogador, a estética dá o tom, que nasce na dinâmica observável e, eventualmente, na mecânica operável. Como demonstra a figura 4

Figura 4 – Visão do designer e do jogador pelo conceito de MDA de Hunicke et al. (2004).



Fonte: adaptado de Hunicke et al. , 2004, p. 2

2.5 Definido elementos ou mecânicas de gamification

Definindo alguns elementos de jogos, Zichermann et al. (2011) exemplificaram como estes foram usados por jogos ou atividades utilizando *gamification* em ambientes digitais. Esses elementos serão abordados a seguir e correlacionados com a aplicação deste trabalho, que será uma aplicação digital também. Os autores partiram do *framework* proposto por Hunicke et al. (2004), tais elementos equivalem as mecânicas do *framework* MDA.

2.5.1 Pontuação

É um sistema para avaliar ou marcar as ações do jogador durante o jogo. Não é preciso necessariamente que estejam visíveis a quem joga, mas a pontuação é um elemento imperativo pois é a partir dela que é possível determinar os comportamentos do jogador e realizar ajustes. Pode ser usado para trocas ou interações especiais do jogador com o sistema. Exemplos de sistemas de pontuação conhecidos nos jogos:

- **Pontuação em dinheiro - Caixa:** representa bem o status do jogador, pois indica sua capacidade de aquisição e quantidade de ações possíveis no jogo. Normalmente, uma pontuação mantida em segredo pelo jogador, assim todos jogadores precisam analisar suas ações para determinar o status de cada um.
- **Pontuação de Video Game - Score:** representa o quanto perto um jogador está de concluir objetivo, alcançar um nível ou vencer o jogo. Normalmente mostrado a todos que jogam.
- **Prestígio social:** representa o nível de prestígio social de um jogador ou de uma rede de jogadores. Um exemplo é o *Facebook* e o *twitter*, que divulgam, respectivamente, número de amigos e seguidores, sendo assim uma espécie de pontuação implícita. Exemplificação na figura 5

Figura 5 – Demonstração do sistema de pontuação de prestígio social do eBay, para classificar a confiança em um vendedor por Zichermann et al. (2011).



Fonte: Zichermann et al. , 2011, p. 41

Qualquer metrificação ou mensuração pode se tornar pontuação na *gamification*, que por sua vez pode ser categorizada em cinco tipos de sistemas de pontuação:

- **Pontos de experiência:** o mais importante de todos, pois mensura o nível de experiência que o jogador possui no jogo. É utilizado para guiar o jogador, atribuir seu nível e distinguir suas atitudes. Pontos de experiência não podem ser perdidos e não possuem um valor limite; conforme o jogador interage com o sistema mais ele ganha experiência. Existe a possibilidade de assumir os pontos de experiência como temporários, expirando mensalmente ou semanalmente, forçando o jogo a voltar em um estado original e ao jogador a sempre interagir. Não são utilizados como troca no sistema.

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: conforme o aluno resolve problemas de algoritmos, classificados em níveis de dificuldade, este recebe mais ou menos pontos de experiência.

- **Pontos resgatáveis:** são pontos de troca para o jogador, conseguidos como recompensa pelo cumprimento de objetivos ou desafios. Possuem um valor limite e exigem uma regulamentação para serem usados, para não causar desequilíbrios no jogo, similarmente a inflação em um mercado.

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: conforme o aluno resolve problemas aplicando algoritmos, ele ganha pontos, para poder trocar em dicas de como resolver problemas futuros.

- **Pontos de habilidades:** são pontos específicos de certas atividades, que servem para recompensar jogadores capazes das mesmas. Podem ser utilizados como pontos bônus para pontos de experiência e resgatáveis.

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: analisando a solução de um aluno a um problema, e possível classificar seu

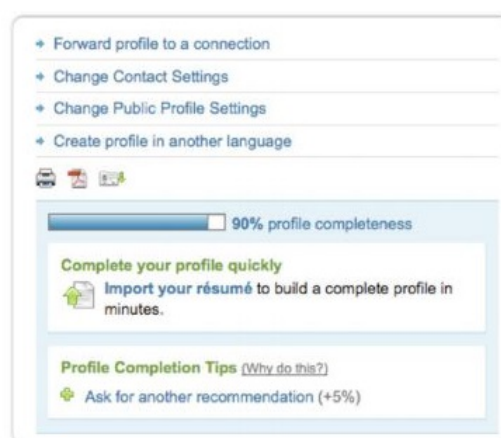
algoritmo em legibilidade, redigibilidade, complexidade de tempo, complexidade de memória e etc para atribuir pontos específicos a cada classificação, ressaltando as habilidades naquela solução.

- **Pontos de *Karma*:** raramente utilizado nos jogos; trata se de uma pontuação na qual o jogador precisa mantê-la zerada para não perder benefícios ou outras pontuações no jogo. Têm o propósito de forçar certos comportamentos nos jogadores: conforme os jogadores realizam tais comportamentos, seus pontos de *Karma* são perdidos. Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: caso o aluno não resolva uma certa quantidade de problemas, ele é punido nos seus pontos resgatáveis.
- **Pontos de reputação:** são os mais difíceis de serem implementados, pois têm o objetivo de mensurar comportamentos sociais dos jogadores, logo levam em consideração muitos fatores sobre certas circunstâncias: não podem ser determinados levando em consideração apenas uma atividade ou desafio, mas de um conjunto dos mesmos.

2.5.2 Níveis

Na maioria dos jogos os níveis indicam o progresso do jogador. Baseado no nível em que o jogador se encontra o fator de complexidade do jogo muda, tornando objetivos mais difíceis, abrindo mais desafios e aumentando as pontuações. São importantes pois situam o jogador, preparando para melhor tomar suas decisões. O progresso pode ser exibido pontualmente, ao longo de um objetivo por exemplo, ou longo de todo jogo. Barras de progresso podem ser utilizadas para informar o jogador, como a barra de progresso utilizado pelo *LinkedIn*, abordada na figura 6

Figura 6 – Demonstração da barra de progresso do *LinkedIn* por Zichermann et al. (2011).



Fonte: Zichermann et al. , 2011, p. 49

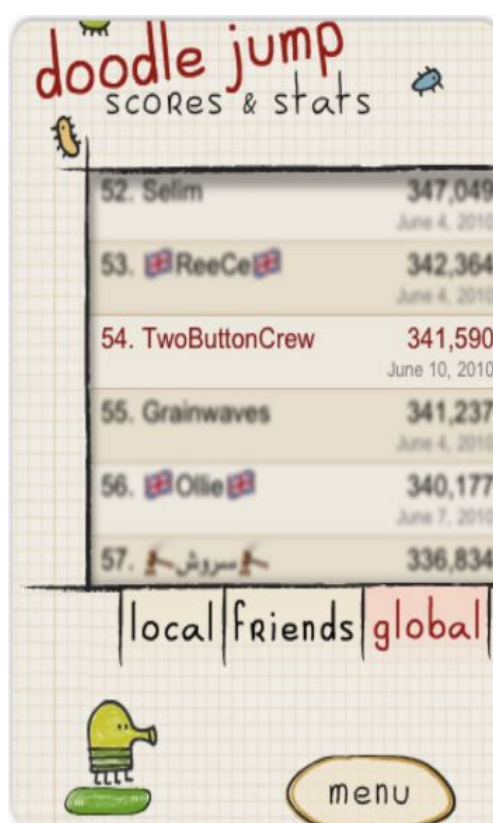
Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: conforme o jogador vai resolvendo problemas, problemas mais difíceis sobre o

mesmo assunto lhe serão direcionados e o jogador ganha uma estrela, que será visível na interface. Cinco estrelas significa a complexidade máxima dos problemas.

2.5.3 Tabela de ranque ou ranqueamento

O objetivo principal das tabelas de ranqueamento é permitir comparações entre os jogadores sobre alguma espécie de pontuação. Um exemplo de tabela é demonstrado na figura 7.

Figura 7 – Exemplo de tabela de ranque do jogo Doodle Jump por Zichermann et al. (2011).



Rank	Player	Score	Date
52.	Selim	347,049	June 4, 2010
53.	ReeCe	342,364	June 4, 2010
54.	TwoButtonCrew	341,590	June 10, 2010
55.	Grainwaves	341,237	June 4, 2010
56.	Ollie	340,177	June 7, 2010
57.	سروش	336,834	

local | friends | global

menu

Fonte: Zichermann et al. , 2011, p. 51

Existem dois tipos de tabelas de ranque:

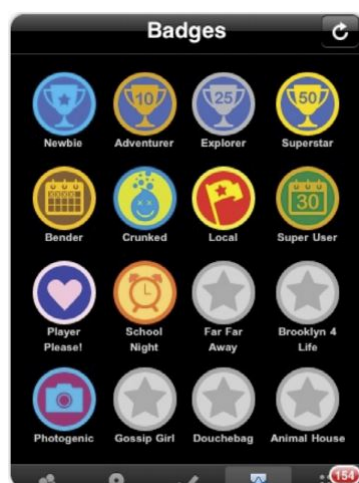
- **Tabela não desincentivadora:** trata-se de uma tabela que sempre colocará a posição do jogador no centro da tabela, independente de sua colocação. Abaixo de sua posição, serão mostrados jogadores que estão muito próximos de ultrapassar sua colocação e logo acima serão mostrados jogadores das colocações mais próximas. O objetivo desta tabela é incentivar o jogador a interagir, mostrando seus avanços e o quanto pode progredir.
- **Tabela infinita:** trata-se de uma tabela com o objetivo de mudar sua apresentação, conforme o jogador sobe ou desce colocações, ampliando ou diminuindo o número de comparações: globais ou locais, ou apresentando outros critérios e classificações.

É necessário cuidado ao planejar uma tabela de ranque, pois dependendo da pontuação que será utilizada para classificar os jogadores na tabela, suas privacidades podem ser feridas, o que diminui o nível de motivação. Se bem montada, a tabela é um dos elementos mais fortes para a motivação, pois está diretamente ligada a competição.

2.5.4 Medalhas

As medalhas são uma excelente maneira de encorajar e promover socialmente os jogadores; são recompensas surpresas conforme o jogador faz progresso e este pode desejar seja por colecionar, por satisfação, por status ou curiosidade. A medalha não se relaciona com a pontuação ou tabela, necessariamente. São elementos estéticos, que podem ser usados para marcar o cumprimento de tarefas e progressos. Analogamente, podemos recorrer as marcas que um carro possui; descrevendo suas peças, status, tipo de piloto, informações de manutenção e etc. para entender como as medalhas podem classificar e descrever os jogadores. A figura demonstra 8 um exemplo de sala de medalhas.

Figura 8 – Sala de medalhas do Foursquare por Zichermann et al. (2011).



Fonte: Zichermann et al. , 2011, p. 57

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: se o aluno resolve uma certa quantidade de problemas de algoritmos em todos os níveis de dificuldade de um determinado assunto ou tema, ele recebe uma medalha.

2.5.5 Narrativa

Esse elemento é utilizado para guiar o jogador, e possui papel importante nos momentos iniciais do jogo; pois precisa oferecer uma tutoria que treine o jogador, o motive e que não o sobrecarregue, para que não desanime de tentar logo nas decisões iniciais. No primeiro momento de jogo, não recomendado ensinar nada sobre o jogo, mas levar o jogador a ter experiências com o mesmo, e isso pode ser feito através de perguntas direcionadas,

ligadas ao cumprimento de tarefas simples que não se pode falhar, recompensado o jogador para que seja cativado logo nos momentos iniciais e para que ajustes sejam feitos para o perfil do novo jogador.

As narrativas também podem ser usadas de maneira implícita, conforme o progresso do jogador ocorre, o sistema se ajusta ao progresso do mesmo, de maneira a sempre manter a motivação e engajamento do jogador.

2.5.6 Desafios e tarefas

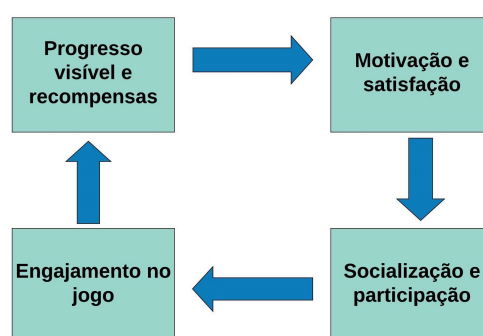
Desafios são a melhor maneira de manter o interesse e a curiosidade do jogador. Estes precisam ser eventos esporádicos, aleatórios com mais riscos e sem muito detalhes.

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: a cada tema, um problema difícil é aberto com grande recompensa e pontuação para o primeiro aluno que o resolver até que o professor passe para o próximo tema.

2.5.7 Ciclos de engajamento social

A projeção de ciclos sociais é muito importante para o sistema, pois define como o jogador deixará o mesmo e como irá retornar. Este ciclo começa com uma motivação emocional do jogador, que o leva ao reengajamento, que gera uma ação social e o resultado desta gera uma recompensa ou progresso visível ao jogador, gerando uma nova motivação emocional. A figura 9 exemplifica esse conceito.

Figura 9 – **Ciclo de engajamento social por Zichermann et al. (2011).**



Fonte: adaptado de Zichermann et al. , 2011 , p. 68

Um exemplo, no twitter, este ciclo pode ser demonstrado da seguinte maneira:

- Motivação emocional: conectar e se expressar
- Reengajamento do jogador: Menções (@)
- Ação social: *Tweets*
- Recompensa ou progresso visível: Seguidores

2.5.8 Feedback

Um elemento crucial para o jogo, pois matem o jogador situado sobre o seu presente momento no jogo. São utilizados para guiar o jogador ao real objetivo do jogo, assim *feedbacks* podem melhorar a interação com o jogador.

Um exemplo da aplicação deste elemento para a *gamification* no ensino de programação: após a submissão de uma solução o sistema avalia a solução, indicando o quanto distante tal solução ficou da solução esperada, expondo os parâmetros se necessário.

2.6 Impactos da gamification na psicologia

Relacionando a *gamification* com o processo de aprendizado, [Domínguez et al. \(2000\)](#) identificam que todas as três áreas são abrangidas no ato do indivíduo jogar: cognitiva, emocional e social. Muitos dos elementos utilizados de *gamification* abrangem todas essas áreas ao mesmo tempo, gerando ciclos entre as mesmas.

2.6.1 Área cognitiva

De acordo com os autores, a área cognitiva é envolvida ao passo que, conforme o indivíduo joga, este executa tarefas orientadas por regras buscando atingir um objetivo. A cada tentativa de finalizar uma tarefa, o nível das habilidades do jogador se eleva. Para que esse processo flua é necessário prover liberdade ou jogador sobre quais tarefas deseja executar e que lhe seja oferecido as informações necessárias para usar suas habilidades.

2.6.2 Área emocional

A área emocional é estimulada mediante o resultado do indivíduo nas tarefas. Caso o resultado seja bem sucedido, o jogador sente uma emoção positiva. A realização de muitas tarefas pode gerar ansiedade no jogador; esta deve ser explorada até que não se transforme em frustração, assim é necessário que cada tarefa se adapte a qualquer nível de habilidade de um jogador. Uma outra maneira de evitar a frustração pode ser diminuindo as penalidades pela falha, incentivando a tentativa, de acordo com os autores.

2.6.3 Área social

A área social diz respeito as interações dos indivíduos durante o jogo, seja na socialização, colaboração ou competição. A competição pode gerar comportamentos negativos ou positivos, mas não há relação direta entre algum elemento da *gamification* e comportamentos negativos da competição.

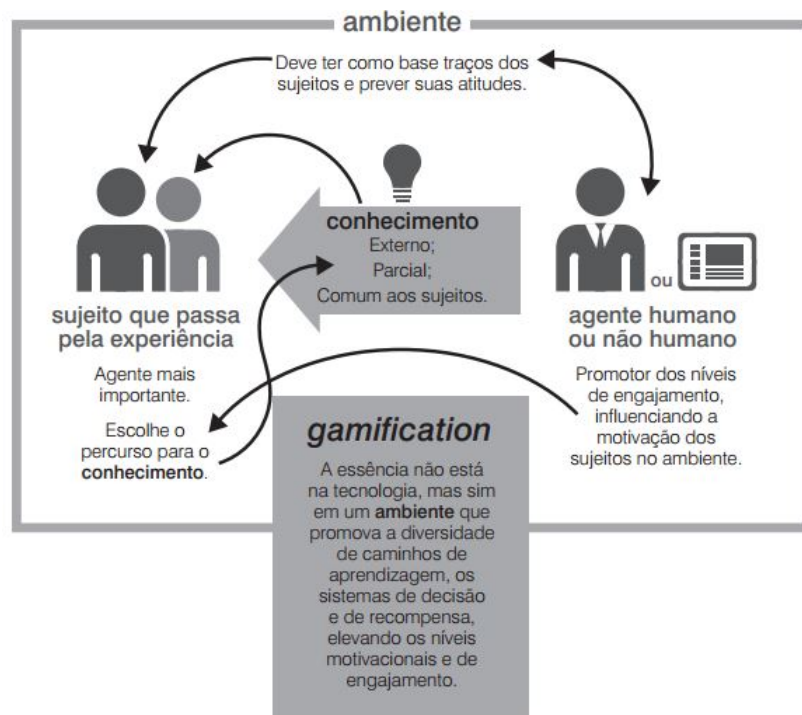
2.7 Gamification e o aprendizado/ensino

Relacionando de maneira direta os elementos de *gamification* com o ambiente de ensino, [Busarrelo \(2016\)](#) expõem que além da motivação, a imersão proporcionada pela *gamification* pode contribuir para o processo de aprendizagem; com a utilização de narrativas, sistemas de recompensas e exploração, tarefas rotineiras podem ficar mais divertidas e diferentes perfis de aprendizados podem ser alcançados pelas diferentes formas de jogar, que no caso do contexto de ensino, significam aprender. O autor sustenta uma hipótese de que a *gamification* pode ser entendida como uma teoria pedagógica contemporânea, e isso se deve a três processos que podem ser contemplados pela *gamification*:

- utilizar um sistema de avaliação baseado no envolvimento de uma comunidade: na escola têm-se o aluno e o professor, no sistema têm-se o jogador e o *feedback*, por exemplo.
- permitir diferentes caminhos, tanto individuais como coletivos para chegar ao objetivo: nas escolas os alunos podem criar grupos de estudo, e aprender um mesmo tema por áreas e autores diferentes, assim como resolver diferentes níveis de exercícios. Isso também pode ser imitado pela utilização de elementos de narrativa, enredo, desafios e recompensas.
- permitir a visualização do progresso: pelas notas e conquistas, o aluno consegue se situar quanto a seu aprendizado, identificando progressos e habilidades, a *gamification* permite essa situação por meio de elementos como tabela de ranque, medalhas e demais estratégias de classificação.

A figura 10 demonstra como a *gamification* propicia esse ciclo, onde o agente ou ferramenta guia o jogador as possibilidades de maneira pré-configurada inicialmente, o jogador, então, toma decisões, aprende, o agente ou ferramenta, avaliam tal conhecimento, contribuindo de maneira mais eficiente para o aprendizado do jogador, e evidenciando duas habilidades para as próximas tomadas de decisões.

Figura 10 – Demonstração da relação entre conhecimento e *gamification* por Busarrello (2016).



Fonte: Busarrello , 2016, p. 42

3 Trabalhos Relacionados

Com a intenção de descobrir se a *gamification* é alvo de trabalhos científicos e se os resultados com a utilização da mesma são positivos, [Hamari, Koivisto e Sarsa \(2014\)](#) realizaram uma revisão bibliográfica. Tais autores realizaram pesquisas nas seguintes bases científicas: Scopus, ScienceDirect, EBSCOHost, Web of Science, ACM Digital library, AISel, Google Scholar, e Proquest com os seguintes termos: *gamification*, *gamif**, *gameful* e *“motivational affordance”*. Os resultados foram catalogados pelos autores em uma tabela, que é exibida na imagem 11.

Figura 11 – **Resultados das pesquisas nas bases científicas por Hamari, Koivisto e Sarsa (2014). A primeira coluna é o nome da base científica, a segunda é quantidade de trabalhos encontrados e a terceira a quantidade de trabalhos revisados.**

EBSCOHost	399	17
Proquest	3423	124
Web of Science	56	56
Scopus	330	293
ScienceDirect	93	93
Google Scholar	3480	N/A
ACM Digital library	239	196
AISel	30	30

Fonte: Adaptado de [Hamari, Koivisto e Sarsa , 2014](#), p. 2

Diante dos resultados os autores aplicaram critérios para eliminar alguns trabalhos, os critérios foram:

- Trabalhos publicados e revisados em conferências internacionais
- Experimentação com *gamification*
- Metodologia explícita
- Busca por engajamento motivacional
- O trabalho focou em *gamification* e não em utilização de jogos

Com a aplicação dos critérios, 23 trabalhos foram escolhidos, a quantidade proveniente de cada base é mostrada a seguir:

- ACM Library: 13
- Scopus: 5
- ScienceDirect: 2
- AISel: 2
- Google Scholar: 1

Os autores catalogaram também os trabalhos que não passaram pelo critérios em 4 grupos:

- Trabalhos conceituais
- Trabalhos descrevendo projetos de engenharia, sem avaliação do projeto
- Trabalhos que apenas mencionaram a *gamification*
- Trabalhos curtos ou resumidos, com pesquisas não terminadas

Com esse passo, os autores escolheram um trabalho que não havia passado pelos critérios para se juntar os outros 23 trabalhos selecionados para um estudo mais profundo. O incremento deste último trabalho não foi bem explicado, apenas relatado como muito relevante. As referências dos trabalhos podem ser encontradas em [A](#).

Após a análises dos trabalhos os autores identificaram quais as técnicas de *gamification* utilizadas em cada trabalho, assim como qual a atividade ou serviço que utilizou a *gamification*, qual a categoria motivacional, quais elementos dos jogos foram utilizados, quais repostas psicológicas, quais respostas comportamentais, quais resultados concluídos, qual o tipo de estudo e qual método de avaliação utilizado. As tabelas [2](#) e [3](#) demonstra essa classificação de cada trabalho de acordo com sua numeração atribuída em [A](#).

Através dessa categorização realizada pelos autores foi possível descobrir quais técnicas de *gamification* foram mais utilizadas:

- Ranqueamento: 10
- Pontuação: 9
- Conquistas, medalhas: 9
- Desafios: 7
- Níveis: 6
- Enredo: 6
- *Feedback*: 6
- Recompensas: 4
- Progressos: 4
- Objetivos: 4

Os trabalhos estudados focaram mais em respostas comportamentais do que respostas psicológicas e os questionários foram a maneira mais utilizada para chegar aos resultados. Os autores chegaram então a conclusão que a utilização *gamification* têm produzido bons resultados. De acordo com os mesmos, muitos aspectos da *gamification* não puderam ser verificados pois seu engajamento dependeu de uma série de outros fatores além dos elementos de jogos utilizados. Foi relatado também que em alguns trabalhos a *gamification* foi bem sucedida, não por méritos da técnica em si, mas pelo fato de ser uma novidade onde foi implementada, perdendo seus efeitos motivacionais com o tempo. Porém nesses mesmos trabalhos, ao retirar a *gamification*, comportamentos negativos foram notados.

Os autores também relataram que a *gamification* está sendo uma técnica muito

abrangente, pois foi aplicada em áreas muito diversificadas, dos 24 trabalhos:

- 1 foi aplicado em comércio
- 9 trabalhos aplicaram em aprendizado/educação
- 1 trabalho aplicado na área de saúde
- 4 trabalhos aplicados a sistemas internos de companhias
- 1 trabalho aplicado em compartilhamento
- 1 trabalho aplicado em gestão de consumo
- 4 trabalhos aplicados em serviço/ambiente de trabalho
- 1 aplicado em área de inovações/ideias
- 1 aplicado em mineração de dados

Boa parte dos trabalhos também utilizou métodos quantitativos; foram dezessete trabalhos. Dois trabalhos utilizaram técnicas qualitativas e cinco utilizaram métodos mixos. Os autores expuseram então suas conclusões sobre a revisão:

- A *gamification* está em alta mais como uma novidade/fenômeno do que como uma metodologia em si;
- É necessário técnicas mais rigorosas para pesquisas com *gamification* que permita avaliar o engajamento resultante da aplicação direta dos elementos dos jogos;
- Os trabalhos precisam se adaptar melhor aos diferentes perfis de jogadores, não concentrando apenas nos elementos dos jogos; estes sozinhos não cumprem o objetivo.

Discutindo sobre a aplicação de *gamification* no ensino, [Faiella e Ricciardi \(2015\)](#) comentam a divergência que existe entre os principais autores influentes sobre as seguintes perguntas: quais elementos dos jogos contribuem para o processo de aprendizado? Será que é somente combinar tais elementos para cumprir esse objetivo? Não é possível responder essas perguntas com os trabalhos que se têm produzido até o momento. Após uma revisão sobre os trabalhos que utilizaram *gamification* no ensino, as autoras reforçam os ganhos positivos nas áreas cognitiva, social e emocional dos indivíduos, assim como a melhora de motivação e engajamento. Porém alertam sobre o risco do objeto de aprendizado se tornar trivial com o uso da técnica; estudantes podem apenas operar conceitos para atingir os objetivos em vez de assimilá-los. Ressaltam também que a utilização pura da técnica, em um jogo completo, não pode resolver problemas de aprendizado.

As autoras enfatizaram o impacto que a *gamification* têm no aprendizado, principalmente pelo elemento de níveis e progresso que podem ser implementados. Conforme um jogador faz progresso resolvendo problemas e desafios, estes vão ficando cada vez mais difíceis, se ajustando ao perfil do jogador, melhorando suas habilidades. Com o progresso, o jogador adquire confiança e explora melhor as possibilidades que um tópico de ensino pode ofertar. O *feedback* também é uma técnica que apresenta impacto relevante na aprendizagem, pois é uma resposta rápida direcionada ao erro particular do jogador. Logo,

um ambiente que permita vários perfis de jogadores, com diferentes habilidades, que se ajuste ao comportamento dos mesmos em tempo real pode sim impactar na experiência de aprendizado dos mesmos. Aconselham que intrutores ou professores que desejam utilizar a *gamification* em serviços ou atividades, devem primeiro traçar quais objetivos querem alcançar, e depois escolher quais elementos irão utilizar, ou seja, os elementos serão apenas os meios ou ferramentas do processo, mas não o processo de aprendizagem em si. Também aconselham a deixar cada elemento da *gamification* como voluntário para o jogador, evitando assim desmotivação por obrigação.

Analizando também as possibilidades de *gamification* aplicada a um processo de aprendizado, Caporarello, Magni e Pennarola (2017) propuseram um framework, mostrado na imagem 12. Esse framework é uma especialização do framework proposto por Hunicke et al. (2004), voltado a *gamification* no ensino. Os autores ressaltaram a necessidade de uso de elementos imersivos.

Figura 12 – Framework para *gamification* de aprendizado por Caporarello, Magni e Pennarola (2017).



Fonte: adaptado de Caporarello, Magni e Pennarola, 2017, p. 4

Após a constatação do sucesso da *gamification*, Majuri, Koivisto e Hamari (2018) também realizaram uma revisão bibliográfica sobre trabalhos com *gamification* direcionados a aplicações no ensino. Realizando pesquisas na base *Scopus* pelo termo: *gamif**, excluindo documentos não acadêmicos, os autores chegaram a oitocentos e sete trabalhos. Cinco trabalhos foram excluídos por serem duplicatas ou estarem indisponíveis, restando assim oitocentos e dois. Trinta e dois trabalhos foram excluídos por serem apenas resumos e não conseguiu se chegar ao trabalho completo ou não havia versão na língua inglesa,

restaram assim setecentos e setenta e três trabalhos. Duzentos e setenta trabalhos foram escolhidos entre esses restantes, por conduzirem pesquisas empíricas ou exploratórias. Desses, por fim, foram selecionados cento e vinte e oito por se tratarem de trabalhos voltados ao ensino/aprendizagem.

Analisando os trabalhos, os autores identificaram que pontuação, desafios, medalhas e ranqueamento são os elementos de jogos mais utilizados pelos trabalhos. Observa-se, assim, uma tendência por elementos orientados a progresso e conquista. Elementos orientados ao social foram o segundo grupo mais notado. Elementos orientados a imersão foram os menos utilizados. Na média, cada trabalho analisado implementou pelo menos 4 elementos de jogo. As respostas psicológicas mais apresentadas foram: percepções de uso e experiência, respostas de diversão, motivação, engajamento e utilidade também foram comuns. As respostas de comportamento mais apresentadas foram: notas, participação e performance. O comportamento social foi raramente notado nos trabalhos. Dos 128 trabalhos, 91 utilizaram métodos quantitativos: 65 (71,43%) dos trabalhos apresentaram resultados positivos, 23 (25,27%) apresentaram resultados iguais ou negativos e 3 (0,03%) apresentaram resultados muito negativos. Grande parte dos trabalhos com método qualitativo apresentaram resultados positivos.

Os autores mencionaram que apenas 7 trabalhos mensuraram os efeitos de um elemento por vez, o que dificulta determinar o real potencial de cada elemento. Todos os trabalhos estavam voltados para as mesmas respostas psicológicas e comportamentais, então ainda existem aplicações da *gamification* que não foram medidas. Muitos trabalhos foram quantitativos, então reações sobre a *gamification* não foram cobertas.

Foi aconselhado 3 observações para futuras pesquisas:

- Planejar elementos da *gamification* voltados para os perfis dos jogadores, levando em consideração fatores contextuais e variações;
- Analisar respostas psicológicas e comportamentais sobre cada elemento de jogo implementado;
- Explorar elementos orientados a imersão e social.

3.1 Avaliação de gamification no aprendizado

Como já explicado, todo contexto de *gamification* envolve um público alvo, portanto é importa desenvolver instrumentos para avaliar a estratégia por seus usuários, como fizeram [Scholtz et al. \(2016\)](#). Os autores desenvolveram e avaliaram uma solução de *gamification* para divulgar e ensinar sobre carreiras na área de ciência da computação. Para o desenvolvimento do trabalho, os autores se basearam no *framework* MDA. A solução foi testada com jovens alunos da África do Sul. Um questionário desenvolvido pelos autores avaliava tanto a experiência de usuário como a usabilidade educacional da solução por eles

desenvolvida. As perguntas foram separadas em 7 conjuntos:

- Conjuntos voltados a experiência de usuário:
 - Questões emocionais (5 perguntas)
 - Engajamento (4 perguntas)
 - Recurso/Apelação visual (3 perguntas)
 - Satisfação (4 perguntas)
- Conjuntos voltados a usabilidade educacional:
 - Clareza de metas, objetivos e resultados (1 pergunta)
 - Reconhecimento, diagnóstico e recuperação de erros (2 perguntas)
 - Feedback, orientação e avaliação (3 perguntas)

Cada conjunto continha perguntas ou afirmações que eram respondidas/avaliadas com uma nota de 1 a 5. Esse questionário possibilitou aos autores mensurar o quanto a solução dos mesmos impactou no processo de aprendizado sobre carreiras de ciências da computação. As perguntas foram elaboradas a partir de um *framework* proposto por [Harpur e De Villiers \(2015\)](#), MUUX-E (a letras indicam: *M-Learning*, *Usability*, *User Experience* e *Educational* respectivamente), que concilia critérios de usabilidade, experiência de usuário e aprendizado mediado por tecnologias. Com os resultados da aplicação do questionário, os autores perceberam que o *feedback* da solução poderia ser melhorado (a menor nota foi 3), que os elementos de *gamification* foram satisfatórios (a menor nota foi 4) e que os alunos apreciaram bem a aparência da solução (a menor nota foi 4).

4 Procedimentos metodológicos

Este trabalho pode ser entendido como um trabalho de caráter exploratório segundo Wazlawick (2009) ou ainda pesquisa aplicada, pela definição de Marconi e Lakatos (2012). Diante dos problemas, que acarretam a área de ensino e aprendizagem de programação, foi feito um estudo sobre as dificuldades no processo de aprendizagem dos alunos e foi escolhido uma estratégia que influencie nessas dificuldades, sendo essa avaliada em uma aplicação com a mesma.

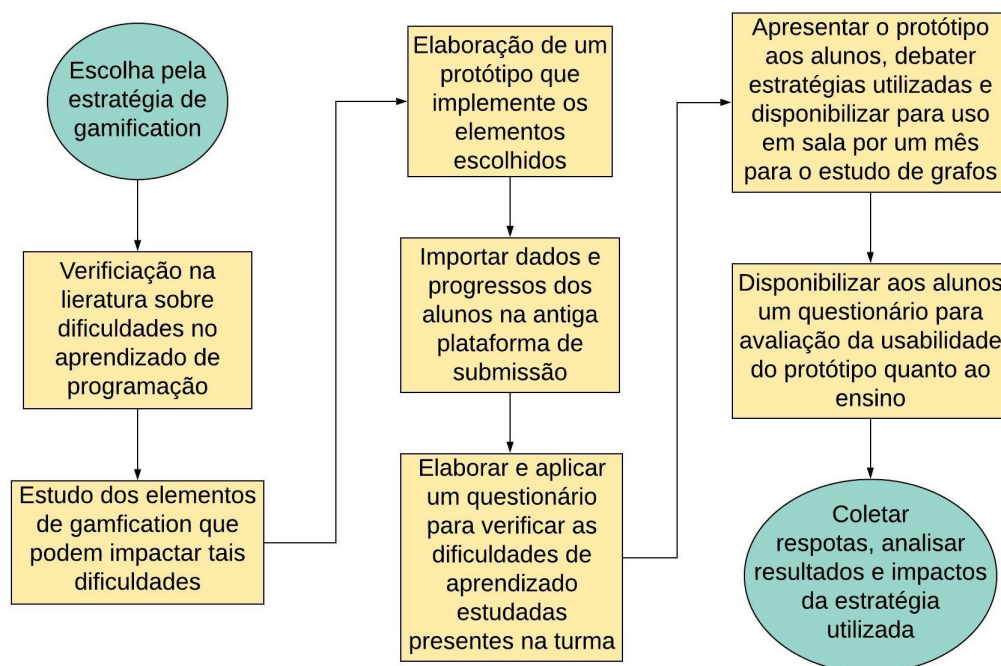
O primeiro passo para o início deste trabalho, que têm como objetivo investigar uma solução de *gamification* para amenização das dificuldades no ensino de programação, foi realizar pesquisas na literatura sobre trabalhos que identificaram tais dificuldades. Com esse passo foi possível perceber quais dificuldades eram gerais ou inerentes as metodologias de ensino utilizadas até então e quais fatores influenciavam para tais dificuldades. No segundo passo foi buscado também na literatura trabalhos que utilizaram *gamification*, conceituaram seus elementos e os relacionaram com o ambiente de ensino. Os elementos que indicavam possíveis impactos benéficos no ensino, que podiam auxiliar na superação das dificuldades, foram escolhidos e estudados mais profundamente.

No terceiro passo, dinâmicas foram criadas, seguindo conceitos específicos para produção de jogos e uma arquitetura de plataforma voltada para o ensino de programação foi projetada, contemplando todas essas dinâmicas. Foi implementado um protótipo de plataforma, com conteúdo e materiais para o ensino de programação. No quarto passo, dados reais de uma outra plataforma foram utilizados para fazer ajustes e testes; isso será explicado mais detalhadamente a seguir. No quinto passo, um questionário abordando as dificuldades notadas e fatores que influenciam nas mesmas foi adaptado e aplicado aos alunos da turma escolhida, para que verificação da coerência entre a literatura e o estudo de caso deste trabalho. No sexto passo, o protótipo foi apresentado aos alunos e uma entrevista semi-estruturada foi feita com os mesmos, analisando a viabilidade do uso de tal solução.

Em um sétimo passo, um questionário de usabilidade educacional foi adaptado para a plataforma e disponibilizados para que os alunos que a desejassem avaliar, pudessem fazê-lo. Esse último passo não foi obrigatório, pois na entrevista foi possível coletar muitas informações que já continham no mesmo. Por fim foi analisado todos os dados recolhidos dos últimos três passos, que serão expostos nas próximas seções. A figura 13 representa num fluxograma esse procedimentos que foram explicados. A determinação do fluxograma seguido foi feita antes da execução direta de cada passo, pois independente do resultado que seria obtido com a execução do mesmo, os passos seguintes poderiam ser executados.

Mesmo que as dificuldades estudadas na literatura não fossem observadas na turma escolhida, o que não aconteceu, pois a estratégia poderia melhorar outros aspectos como o engajamento dos alunos e participação dos mesmos, que é o que normalmente se espera de resultado da utilização da *gamification*.

Figura 13 – Fluxo do desenvolvimento deste trabalho.



Problemas no ensino/aprendizagem de programação

Fonte: o próprio autor.

4.1 Ambiente de estudo: CEFET-MG Campus Timóteo

O Campus CEFET-MG Timóteo oferece dois cursos de graduação: engenharia de computação e metalurgia. O curso de computação possui muita familiaridade com a área de ciências da computação, enfatizando a lógica de programação. Durante os 4 primeiros semestres do curso, 4 disciplinas fundamentais do eixo de lógica de programação serão ofertadas:

- Programação de computadores I: que prioriza o ensino das estruturas básicas de programação, como declaração de variáveis, vetores, matrizes, condicionais e laços de repetição.
- Programação de computadores II: que prioriza o ensino de conceitos como herança, polimorfismo, classes, objetos e demais temas de programação orientada a objetos.
- Algoritmos e estruturas de dados I e II: que prioriza o ensino de algoritmos de ordenação, recurção, tipos abstratos de dados (listas, pilhas, filas), grafos e *hashing*.

Todo eixo de programação do curso depende dos conceitos envolvidos nessas 4 disciplinas.

4.1.1 A disciplina escolhida

Foi escolhido a disciplina de algoritmos e estrutura de dados II pois os temas envolvidos em tal disciplina são os mais abstratos e espera-se que sejam considerados como os mais difíceis pelos alunos. Se espera também que os alunos dessa turma tenham mais maturidade para identificar dificuldades durante seu aprendizado, pois já passaram por outras três matérias de ensino de programação. Será avaliado somente o tópico de grafos com alunos, que é a estrutura de dados mais complexa a nível de computação ensinada no curso. Os alunos que cursam a disciplina, já estão acostumados a utilizar, na própria disciplina, uma plataforma *open-source* de submissão de algoritmos (*DOMJudge*), usada em maratonas de programação. Essa experiência pode diminuir a resistência e dificuldades dos mesmos ao avaliar o protótipo desenvolvido. Pode-se dizer que o protótipo desenvolvido neste trabalho é uma modificação da plataforma *DOMJudge*, em que a mesma foi reprojeta para a utilização de elementos de *gamification*, voltada para o ensino de programação em vez de ambientes de maratona de programação. Porém muito pouco código fonte foi reaproveitado do *DOMJudge*, sua contribuição foi a base de dados com exercícios resolvidos pelos alunos, analisada para auxiliar a construção do protótipo desenvolvido.

4.2 Questionário de verificação das dificuldades

Como já apresentado, [Lahtinen et al. \(2005\)](#) desenvolveram um questionário que pode ser aplicado tanto a professores como a alunos com o objetivo de identificar dificuldades com o aprendizado em programação. O mesmo questionário foi aplicado aos alunos da disciplina escolhida. Isso foi feito, para determinar se a turma escolhida compartilha das mesmas dificuldades apresentadas nos demais trabalhos citados.

O questionário é dividido em 4 seções, que somam um total de 30 itens. Em cada seção, o aluno deve analisar os itens apresentados, com uma nota de 1 a 5.

4.2.1 Primeira seção do questionário

Na primeira seção são apresentados aos alunos, cenários ou situações que podem ocorrer durante o aprendizado de programação, em 7 itens. E o aluno precisa avaliar qual seu grau de dificuldade em aprender em cada um. Será questionado se o aluno sente dificuldade em:

- preparar ambientes para desenvolvimento
- acessar as ferramentas para programar (ex: computadores, redes)

- utilizar estruturas de dados para programar (ex: pilhas, grafos)
- aprender a sintaxe da linguagem
- projetar lógica de programas para uma determinada tarefa
- analisar e corrigir seus próprios programas

O objetivo dessa seção é avaliar o impacto de fatores externos ao aprendizado de programação, que possibilitam a mesma.

4.2.2 Segunda seção do questionário

Na segunda seção são apresentados 12 conceitos de programação necessários para a construção de programas, que são ensinados durante o curso. O aluno deve avaliar o grau de dificuldade em aprender cada conceito. Os conceitos de programação são incrementos uns dos outros, ficando cada vez mais abstratos ao longo do curso. O objetivo dessa seção é avaliar a grau de dificuldade de acordo com a complexidade do conceito.

4.2.3 Terceira e quarta seção do questionário

Na terceira e quarta seção são apresentados materiais ou formas de estudo possíveis ao aluno durante o curso. O aluno deve avaliar quais materiais ou formas mais impactam de maneira benéfica no seu aprendizado a programação. Quanto aos materiais, é avaliado as possibilidades por:

- Livros
- Apresentações montados pelo professor
- Listas teóricas de exercícios
- Listas práticas

Quanto as formas, é avaliado as possibilidades em:

- Estudo em turma
- Estudo com amigos
- Estudo individual

São 11 itens no total, sendo que a terceira seção enfoca nas formas e as quartas nos materiais, e nesta última, o aluno pode avaliar materiais que não foram utilizados durante seu aprendizado; isso para que indique fontes que considera melhor que as que utilizou. O objetivo dessa seção é relacionar os materiais e formas diretamente com o impacto no aprendizado.

4.3 Implementação de uma plataforma com gamification

Este trabalho se propõem a projetar e avaliar uma solução de *gamification* para áreas que envolvem o ensino de programação, utilizando dinâmicas que envolvam os elementos de *gamification*. Estes elementos foram implementados em uma plataforma online. As aulas disponibilizadas pela plataforma, que contemplou tal solução, foram adaptadas de materiais de outras plataformas de ensino a distância de programação, assim como materiais do próprio professor da disciplina. Reconhece-se que isso pode impactar na avaliação das técnicas de *gamification*, mas o propósito aqui é avaliar puramente o quanto for possível os elementos e dinâmicas. A plataforma desenvolvida foi nomeada “Mundo Do Código”. Como inspiração para o desenvolvimento da mesma, foram estudadas outras plataformas de ensino de programação: o *CodCad* e o *SoloLearn*.

Dois tipos de usuários podem autenticar na plataforma: professor e aluno. O professor pode cadastrar alunos, cursos, tópicos, exercícios, casos de teste, dificuldades dos exercícios, dificuldades dos casos de teste e habilidades de programação.

O cadastro de cursos é uma funcionalidade mais conceitual; serve apenas para agrupar os tópicos. No cadastro de tópicos, o professor pode adicionar vídeos, códigos fonte de exemplo, materiais em PDF e textos. Cada tópico pode ou não ter exercícios. No cadastro de exercícios o professor informa o enunciado, as habilidades de programação que o exercício envolve e suas respectivas pontuações, uma pontuação para o exercício e sua dificuldade (estas também são cadastradas por ele). Um exercício tem vários casos de teste; um caso de teste consiste em um grupo de entradas e saídas associados. Quando uma solução em algoritmo é alimentada com as entradas do caso de teste, ele deve ter como resposta as mesmas saídas do caso de teste para ser considerado uma solução válida. O professor ao cadastrar o caso de teste, pode informar uma pontuação para o mesmo, uma dificuldade, habilidades de programação envolvidas assim como suas respectivas pontuações.

Um aluno cadastrado na plataforma pode alterar seus dados, visualizar os cursos, acessar os tópicos e seus materiais (foto no apêndice A.5), acessar os exercícios (foto no apêndice A.7) dos tópicos, enviar soluções para tais exercícios (foto no apêndice A.9) e obter o *feedback* dos mesmos (foto no apêndice A.4).

Os exercícios utilizados na plataforma foram exercícios de maratonas de programação, que já estavam sendo resolvidos pelos alunos da disciplina pelo *DOMJudge*. Escolhendo as maratonas como referência, é possível determinar com mais clareza o nível de dificuldade dos exercícios, pois nessas maratonas os exercícios são divididos em vários níveis: aquecimento, nível 1, nível 2, etc. E os exercícios são elaborados de maneira a explorar várias habilidades de programação em um único problema. Utilizou-se exercícios dos seguintes eventos:

- Maratona Regional SBC
- Maratona Final SBC
- Maratona Mineira
- OBI Nivel 1
- OBI Nivel 2
- OBI Nivel 3
- CTBC

4.3.1 Elementos e dinâmicas de gamification utilizadas

Foi implementado uma tabela de *rank* (foto no apêndice [A.1](#)), onde são mostradas as pontuações por cada exercício resolvido pelo aluno, assim como a quantidade de soluções enviadas, a quantidade de problemas resolvidos (que obtiveram uma solução que passou em todos os testes) e a quantidade de medalhas obtidas. Essa pontuação pode ser parcial, depende de quantos casos de teste a solução do aluno resolveu. O objetivo dessa tabela é situar o aluno em relação a turma, assim como instigar ciclos de amizades que favoreçam o aprendizado. Os nomes dos alunos exibidos na plataforma são nomes fantasias escolhidos pelos menos, assim a integridade do aluno fica assegurada e a imersão propiciada da plataforma é acrescida.

Foi implementado pontuações em habilidades: conforme o aluno resolve problemas e seus casos de teste, ele acumula pontos de habilidades em programação, atribuídos aos mesmos. Assim o aluno consegue ter conhecimento de qual suas melhores habilidades e quais habilidades precisa desenvolver mais. Para incentivar o aluno a desenvolver as mesmas, estas só aparecem para ele quando ele atinge pelo menos 1 ponto na mesma, ou seja, ele só descobre todas as habilidades de programação se resolver vários tipos de problemas (foto no apêndice [A.2](#)).

Foi implementado um quadro de medalhas, que o aluno adquire conforme resolve os exercícios na plataforma (foto no apêndice [A.3](#)). Foram criadas 28 medalhas, considerando a seguinte distribuição:

- 7 medalhas concedidas conforme o número de envios de solução do aluno, ou seja, quantos exercícios ele tentou resolver
- 7 medalhas concedidas conforme o número de exercícios resolvidos pelo aluno
- 4 medalhas concedidas conforme o número de exercícios resolvidos pelo aluno em para uma determinada dificuldade
- 4 medalhas concedidas conforme o número de habilidades descobertas pelo aluno
- 4 medalhas concedidas conforme a pontuação em exercícios resolvidos pelo aluno
- 1 medalha se o aluno resolver todos os exercícios da plataforma
- 1 medalha se o aluno possui a maior pontuação da plataforma

Outra maneira de recompensa utilizada, para motivar o aluno a resolver uma quantidade mínima de exercícios de um curso, é a emissão de um certificado falso, atestando que o aluno completou o curso pela plataforma. Foi implementado uma barra de progresso para cada curso, onde 100% da barra é alcançado quando o aluno resolve todos os exercícios do curso (foto no apêndice [A.10](#)).

Foi utilizado um julgador para automatizar o sistema de correção dos exercícios, não sendo necessário a correção manual ou fiscalização do professor. O julgador utilizado é o mesmo que a plataforma *DOMJudge* utiliza. Com o intuito de ajudar o aluno na escolha de qual exercício resolver ou melhorar a solução, todos os exercícios são apresentados com a sua dificuldade, a qual tópico ele pertence e sua situação (Não corrigido, corrigido, resolvido). Quando o aluno submete uma solução pela plataforma, assim que corrigida, ele pode ver quais testes acertou ou errou, o quanto computacionalmente viável foi sua solução (analisando tempo de execução e memória gastos) e se há algum erro de compilação (fotos nos apêndice [A.4](#)).

Para instigar os ciclos sociais na plataforma, foi concedido ao aluno possuir um avatar (uma foto de perfil para sua conta) e também criar uma lista de amigos. Por essa lista ele pode conversar com os mesmos por chat (foto no apêndice [A.11](#)). Também é possível ao aluno adicionar comentários aos tópicos e exercícios, ficando visível a todos; uma espécie de fórum (foto no apêndice [A.8](#)). Assim incentiva-se o aprendizado em comunidade.

Para avaliar o fluxo de acessos do aluno pela plataforma, foi marcado a ocorrência de visualizações de cada grupo de telas relacionados a um elemento de *gamification*. Assim têm-se um indício do quanto o quadro de medalhas, *rank*, habilidades e amigos foram atraentes para os alunos.

Todos esses elementos implementados, excluindo o chat e fórum, foram validados utilizando dados passados de alunos que já usaram a plataforma *DOMJudge* no curso do CEFET-MG Timóteo. Também foi realizada uma apresentação da plataforma aos alunos, explicando o que é a *gamification*, qual o propósito de cada elemento de *gamification* utilizado na plataforma. A cada elemento demonstrado, era feito um debate com alunos sobre o possível impacto desse elemento no ensino de programação, assim como perguntado se os mesmos aprovavam a utilização de tal elemento, se tinham alguma oposição a utilização ou se acreditavam que o elemento não contribuiria para superar as dificuldades no aprendizado.

4.3.2 Arquitetura da plataforma

A plataforma foi projetada para que sua base de dados forneça condições para que um curso possua vários tópicos e que cada tópico possua exercícios próprios. Assim, é possível desenvolver as dinâmicas de narrativas: cada exercício dentro de um tópico

pode possuir enunciados diferentes, abordando as mesmas habilidades em programação, atingindo se assim mais perfis de alunos (como descrito em 2.7). A plataforma também fornece condições para correção automática: cada exercício possui vários testes, que validam as habilidades de programação desenvolvidas pelo aluno com a solução que este enviou. É possível que em um exercício, o aluno desenvolva de forma mais ampla uma habilidade do que em outro exercício (de um mesmo tópico ou não), assim além de identificar as habilidades que podem ser desenvolvidas no exercício, estas também possuem uma pontuação que é proporcional a seu desenvolvimento.

A figura 14 demonstra a arquitetura da plataforma desenvolvida. Os blocos em cinza agrupam funções da plataforma, que são:

- Prover para o aluno materiais para o estudo, com diferentes mídias e exibições (textual, imagens ou vídeos)
- Prover uma classificação para os exercícios em grau de dificuldade e quais habilidades de programação foram desenvolvidas
- Prover um sistema de *feedback* ao aluno e usar do mesmo para a distribuição de pontuações, medalhas e classificação no *rank*.

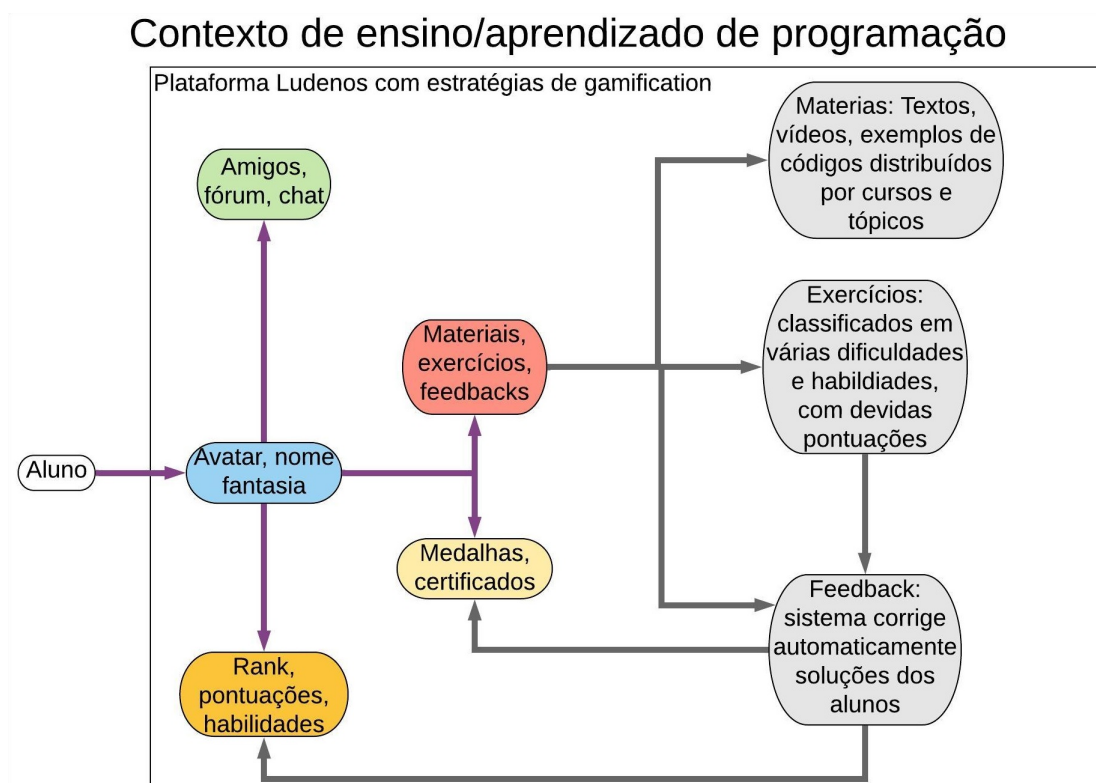
As setas cinzas são informações usadas ou geradas pelas funções das plataforma. O bloco azul destaca a dinâmica de imersão utilizada na plataforma, que possibilita ao aluno escolher um avatar e nome fantasia. O bloco verde destaca a estratégia para desenvolvimento social do aluno na plataforma, promovendo ciclos sociais: se dará pela possibilidade de crias ciclos de amizade, conversar por amigos com chat e o fórum que permite comentários nos exercícios e tópicos. O bloco amarelo destaca dinâmicas de recompensas utilizadas para motivar o aluno: as medalhas e o certificado. O bloco laranja são dinâmicas utilizadas para situar o aluno quanto a seu progresso e também para promover competição: pontuações em *rank* e habilidades. O bloco branco é o aluno e suas interações com os blocos da plataforma são indicadas pelas setas roxas.

4.3.3 Fluxo de atividades esperado do aluno na plataforma

A figura 15 demonstra o fluxo esperado de atividades do aluno na plataforma, uma vez que este esteja autenticado e pertença a pelo menos uma turma na mesma. O fluxo inicia com a autenticação do aluno, após isso o aluno escolhe um curso, um tópico e estuda pelo material que é disponibilizado. Quando conclui o estudo, o aluno escolhe um exercício de seu agrado para resolver, baseado na sua dificuldade. Após o aluno enviar a solução do exercício o sistema irá retornar a correção da mesma para o aluno. Aqui existem duas possibilidades: o aluno se contenta com sua solução ou não.

Para o aluno resolver um exercício na plataforma, basta que sua solução passe em um dos testes para a mesma. Porém, um exercício pode ter mais testes, cada um com

Figura 14 – **Arquitetura da plataforma Ludenos, destacando funções da mesma e suas dinâmicas em interface com o aluno.**



Fonte: o próprio autor.

suas habilidades de programação, logo o aluno pode pontuar mais se explorar mais suas habilidades em programação, buscando passar em todos os testes. Essa dinâmica incentiva o aluno que quer ir além do pedido e testar suas habilidades.

Se o aluno considerar seu *feedback* positivo, ele visitará as telas:

- de *rank*, para saber sua situação em relação aos outros jogadores;
- de habilidades, para saber quais as novas habilidades desenvolvidas e as pontuações;
- de quadro de medalhas, para apreciar suas conquistas.

Esses dados apresentados ao aluno podem motivá-lo a continuar os estudos, seja por conquista, competição ou progresso, atingindo assim diretamente a área emocional do aluno (como exposto em 2.6.2). Após isso, o aluno irá divulgar sua experiência, seja fazendo comentários nos tópicos ou exercícios, ou pelo chat com seu ciclo de amigos, adicionados pelo próprio aluno.

A plataforma oferece um certificado falso para o aluno que resolve uma quantidade mínima de exercícios por tópicos, sendo este uma das recompensas principais da plataforma. Caso o aluno tenha atingido essa quantidade, pode finalizar seus estudos pela plataforma, realizando a desautenticação, pois cumpriu o curso disponível. O aluno pode querer continuar competindo, para desenvolver mais habilidades, ganhar mais medalhas,

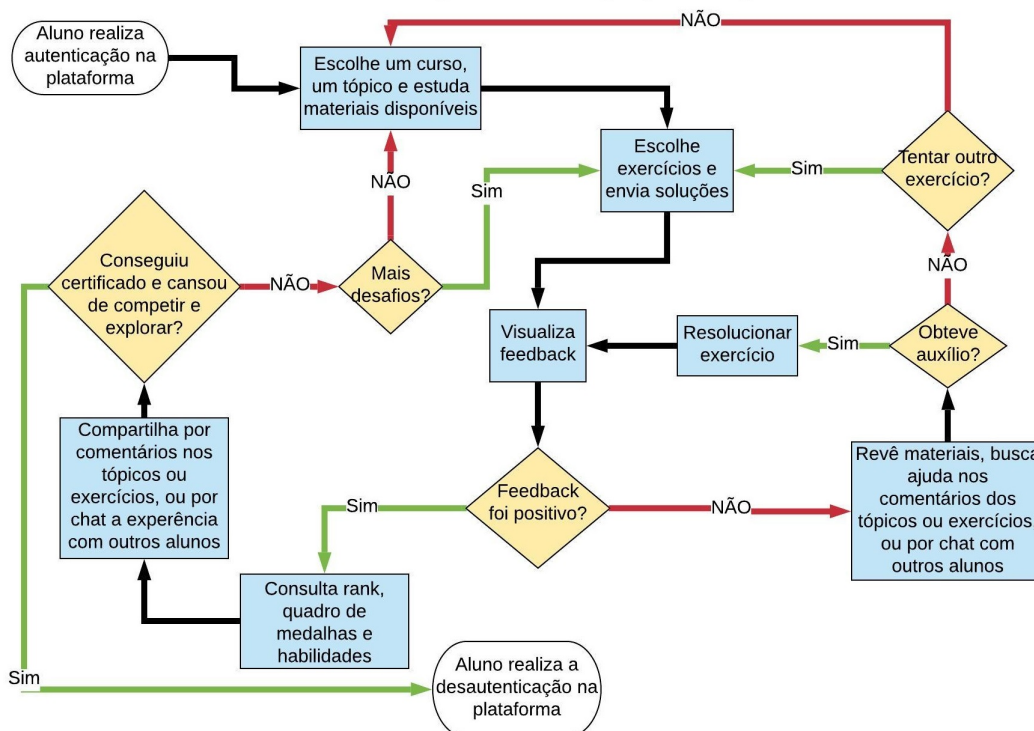
ou subir no *rank*.

Caso não tenha desbloqueado o certificado, o aluno poderá escolher entre resolver mais exercícios do tópico, ou estudar um novo tópico. Mas, pode ser que o *feedback* obtido pelo aluno na resolução dos exercícios foi desaprovado pelo mesmo. Este aluno irá buscar auxílio para melhorar a sua solução, seja pelos materiais, ou pelos comentários e chat. Caso ele consiga melhorar, irá resolver o exercício, acarretando em outra correção. A plataforma não mantém um histórico de soluções do aluno, sendo a última a ser enviada, a utilizada pelo sistema. Isso permite ao aluno acobertar tentativas ruins de solução. Caso o aluno não consiga melhorar sua solução, resta duas opções ao mesmo: trocar de exercício, para um que seja mais fácil ou mais compatível com seu perfil, ou trocar de tópico, iniciando assim novos estudos. A plataforma não tranca tópicos, todos estão abertos ao aluno. É reconhecido que um tópico pode depender de outro quanto a conteúdo, mas para permitir a exploração do aluno e avaliar a mesma, os tópicos estão sem requisitos.

As setas vermelhas na figura 15 indicam que a decisão do aluno foi um não em relação a pergunta feita nos blocos de decisão; os blocos amarelos. As setas verdes indicam que a decisão foi um sim. As setas pretas indicam o fluxo sem decisão, os blocos azuis são atividades sem decisão também. Os blocos brancos são de início e fim do fluxo.

Figura 15 – Fluxo de atividade esperado pelo aluno na plataforma, destacando decisões que o mesmo pode tomar de acordo com seu progresso.

Fluxo de atividades do aluno na plataforma (esperado)



Fonte: o próprio autor.

4.4 Apresentação do protótipo e realização da entrevista semi-estruturada

Com o desenvolvimento do protótipo, o mesmo foi apresentado aos alunos, tanto para explanação de sua utilização como para observar a recepção dos alunos ao mesmo. Como não era possível prever a reação dos alunos a plataforma, foi planejado uma entrevista semi-estruturada, baseada em um questionário qualitativo de experiência explicado a seguir. Assim foi possível perceber o impacto de cada elemento utilizado na plataforma diante dos alunos.

4.5 Questionário qualitativo sobre o protótipo

Foi aplicado um questionário adaptado do trabalho de [Scholtz et al. \(2016\)](#) para avaliar a experiência dos alunos com a plataforma. O questionário é composto por duas seções principais, que se subdividem em quatro e três subseções respectivamente. Cada subseção apresenta itens de usabilidade para serem classificados em notas de 1 a 5, somando um total de 16 itens.

4.5.1 Primeira seção

Na primeira seção o aluno avalia a plataforma nos critérios de usabilidade geral, que podem ser aplicados a qualquer *software*. Cada subseção apresenta itens, respectivamente, para avaliar o impacto do uso da plataforma em questões emocionais, de engajamento, visuais e de satisfação sentidas pelo aluno. São 14 itens no total, que podem ser avaliados com uma nota de 1 a 5, indicando, de maneira crescente, o quanto foi benéfico o uso da plataforma.

4.5.2 Segunda seção

Na segunda seção o aluno avalia a plataforma nos critérios de usabilidade voltados para o aprendizado, especificamente. Cada subseção permitirá aos alunos avaliar os materiais, exercícios e correções oferecidas pela plataforma. São 6 itens no total, que podem ser avaliados com uma nota de 1 a 5, indicando, de maneira crescente, o quanto foi benéfico o uso da plataforma para o aprendizado.

5 Análise e Discussão dos Resultados

5.1 Questionário sobre dificuldades no aprendizado de programação

O questionário foi aplicado no mesmo dia em que foi apresentado a plataforma aos alunos. Foi impresso 10 folhas e deixadas a disposição daqueles que estavam presentes para escolher responder. Não foi requerido nenhum dado pessoal do aluno que respondeu o questionário. As dez cópias foram respondidas pelos mesmos, na presença do aplicador (o próprio autor), caso houvesse alguma dúvida para ser sanada. A seguir é apresentado o questionário, a média de resposta dos alunos e também o desvio padrão.

Questionário para identificação de dificuldades no aprendizado de programação

Orientação: Responda cada item com um número inteiro de 1 a 5 (1 é a nota mínima e 5 a máxima)

Q1 - Com quais cenários você sente mais dificuldade ao aprender programação?

- I1 Usar o ambiente de desenvolvimento de programação: média de 1.6, desvio padrão de 0.69921
- I2 Obter acesso aos computadores e redes: média de 1.6, desvio padrão de 0.96609
- I3 Entender as estruturas de programação: média de 3.3, desvio padrão de 1.2517
- I4 Aprender a programar na sintaxe da linguagem: média de 2.4, desvio padrão de 0.5164
- I5 Projetar um programa para executar determinada tarefa: média de 3, desvio padrão de 0.4714
- I6 Dividir tarefas em funções ou procedimentos: média de 2.2, desvio padrão de 0.91894
- I7 Encontrar erros no seu próprio programa: média de 2.4, desvio padrão de 0.96609

Q2 - Quais conceitos de programação você têm mais dificuldade para aprender?

- I1 Variáveis (ciclo de vida, escopo): média de 1.2, desvio padrão de 0.42164
- I2 Estruturas condicionais ou de seleção: média de 1.3, desvio padrão de 0.48305
- I3 Estruturas de laço ou *loops*: média de 1.6, desvio padrão de 0.69921
- I4 Recursão: média de 3.1, desvio padrão de 0.99443

- I5 Vetores: média de 1.4, desvio padrão de 0.96609
- I6 Ponteiros e referências: média de 3.2, desvio padrão de 1.0328
- I7 Tipos de parâmetros: média de 2.7, desvio padrão de 1.3375
- I8 Tipos de estruturas de dados: média de 3, desvio padrão de 1.2472
- I9 Tipos abstratos de estruturas de dados: média de 3.6, desvio padrão de 0.96609
- I10 Manipulação de entradas e saídas: média de 2.1, desvio padrão de 0.56765
- I11 Tratamento de erros: média de 3.3, desvio padrão de 0.48305
- I12 Usar bibliotecas da linguagem: média de 3.1, desvio padrão de 0.73786

Q3 - Quando você sente que aprende assuntos de programação?

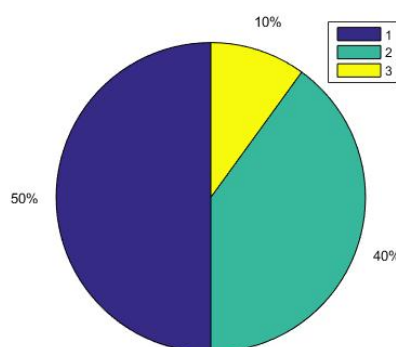
- I1 Nas leituras de livros: média de 1.9, desvio padrão de 1.2867
- I2 Em pequenas listas de exercícios: média de 3.9, desvio padrão de 0.8756
- I3 Em atividades/aulas práticas: média de 3.6, desvio padrão de 1.3499
- I4 Estudando/pesquisando sozinho: média de 3.2, desvio padrão de 1.3984
- I5 Quando realiza os trabalhos sozinho: média de 3.3, desvio padrão de 1.4181

Q4 - Quais materiais te ajudaram ou ajudariam a aprender programação?

- I1 Livros de programação: média de 2.4, desvio padrão de 1.3499
- I2 Apresentações/materiais da sala de aula: média de 2.8, desvio padrão de 0.63246
- I3 Questões de exercícios e suas soluções: média de 3.8, desvio padrão de 1.0328
- I4 Exemplos de programas ou trechos de fonte: média de 4.3, desvio padrão de 1.0593
- I5 Imagens da estrutura de programação: média de 3.5, desvio padrão de 1.0801
- I6 Visualizações interativas e dinâmicas dos algoritmos e suas lógicas: média de 4, desvio padrão de 1.1547

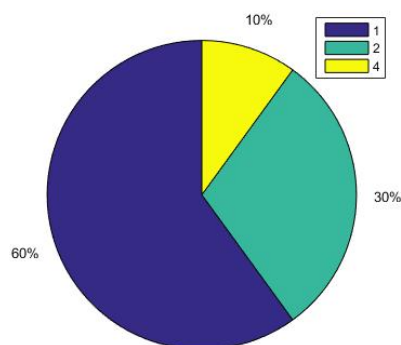
A seguir são apresentados gráficos com a quantidade de respostas por opção, para cada item das questões.

Figura 16 – Quantidade de respostas por opções da questão: Q1 - I1



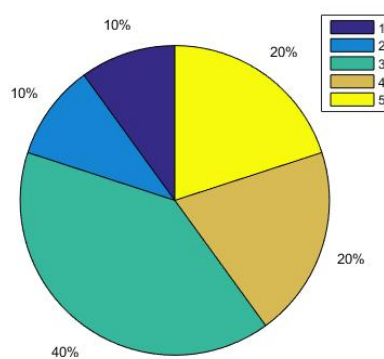
Fonte: o próprio autor.

Figura 17 – Quantidade de respostas por opções da questão: Q1 - I2



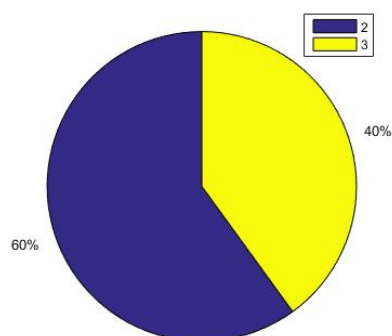
Fonte: o próprio autor.

Figura 18 – Quantidade de respostas por opções da questão: Q1 - I3



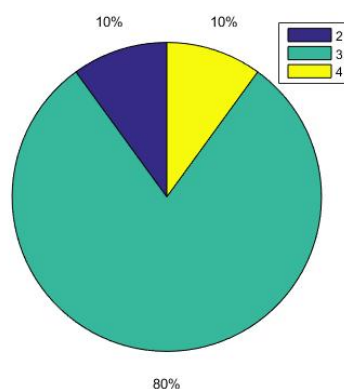
Fonte: o próprio autor.

Figura 19 – Quantidade de respostas por opções da questão: Q1 - I4



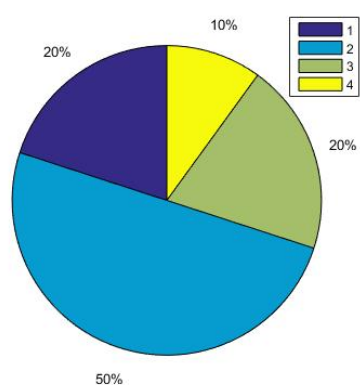
Fonte: o próprio autor.

Figura 20 – Quantidade de respostas por opções da questão: Q1 - I5



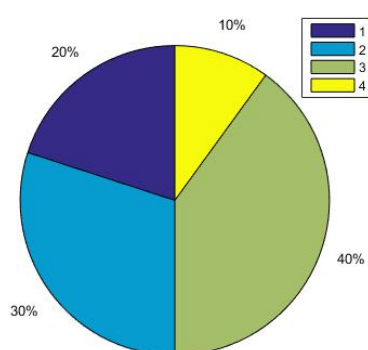
Fonte: o próprio autor.

Figura 21 – Quantidade de respostas por opções da questão: Q1 - I6



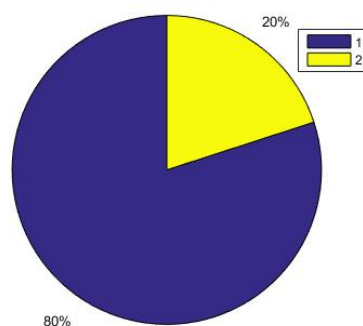
Fonte: o próprio autor.

Figura 22 – Quantidade de respostas por opções da questão: Q1 - I7



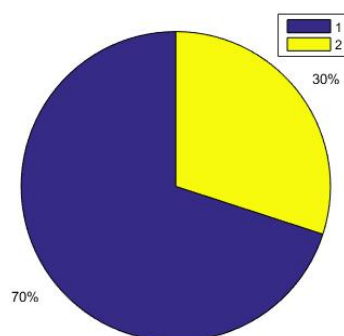
Fonte: o próprio autor.

Figura 23 – Quantidade de respostas por opções da questão: Q2 - I1



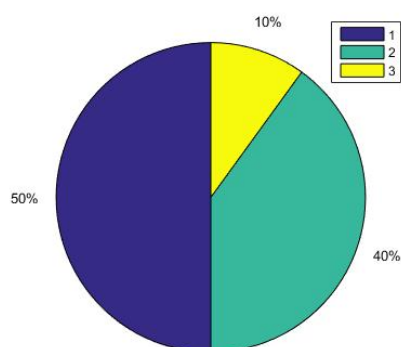
Fonte: o próprio autor.

Figura 24 – Quantidade de respostas por opções da questão: Q2 - I2



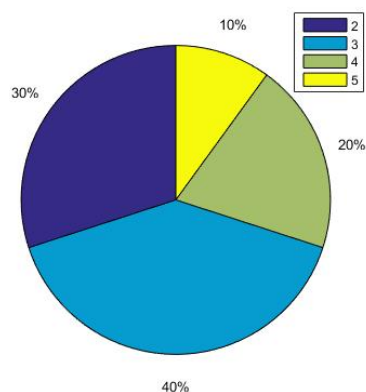
Fonte: o próprio autor.

Figura 25 – Quantidade de respostas por opções da questão: Q2 - I3



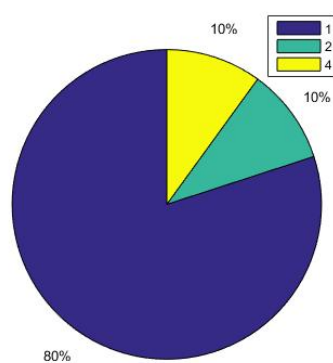
Fonte: o próprio autor.

Figura 26 – Quantidade de respostas por opções da questão: Q2 - I4



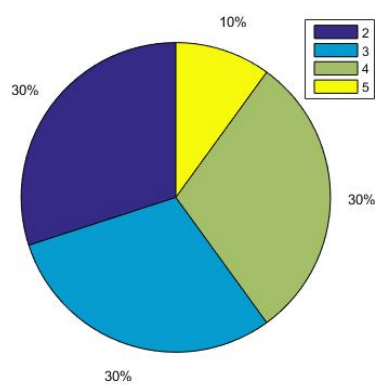
Fonte: o próprio autor.

Figura 27 – Quantidade de respostas por opções da questão: Q2 - I5



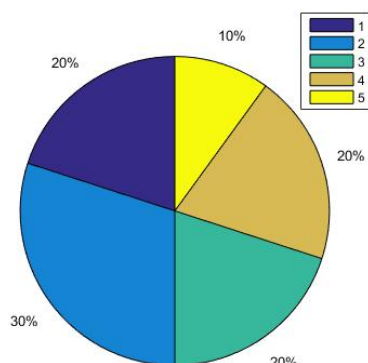
Fonte: o próprio autor.

Figura 28 – Quantidade de respostas por opções da questão: Q2 - I6



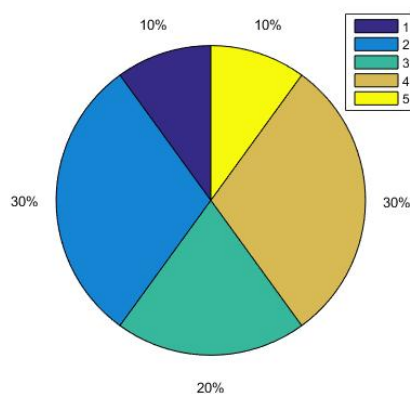
Fonte: o próprio autor.

Figura 29 – Quantidade de respostas por opções da questão: Q2 - I7



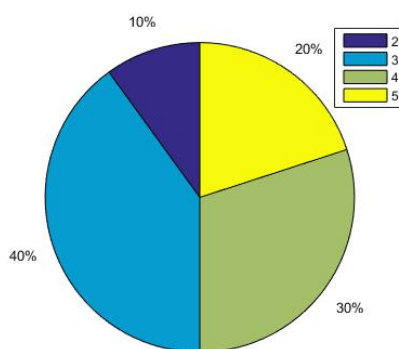
Fonte: o próprio autor.

Figura 30 – Quantidade de respostas por opções da questão: Q2 - I8



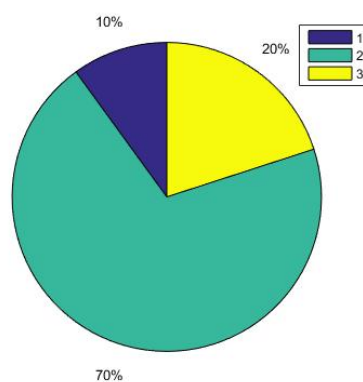
Fonte: o próprio autor.

Figura 31 – Quantidade de respostas por opções da questão: Q2 - I9



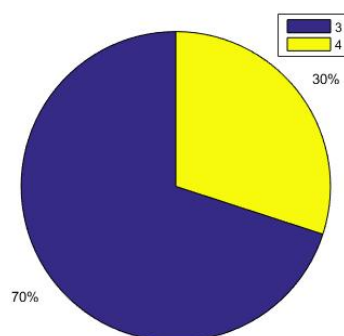
Fonte: o próprio autor.

Figura 32 – Quantidade de respostas por opções da questão: Q2 - I10



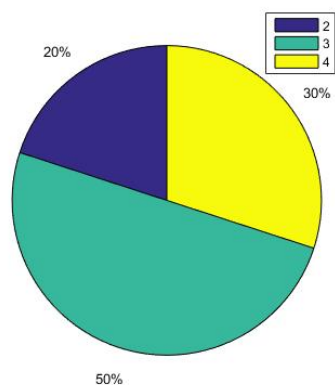
Fonte: o próprio autor.

Figura 33 – Quantidade de respostas por opções da questão: Q2 - I11



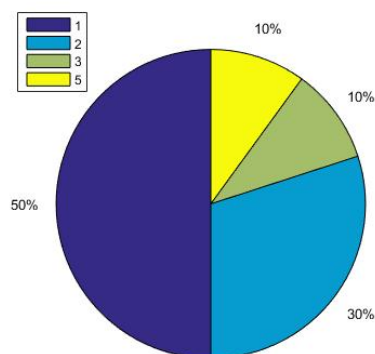
Fonte: o próprio autor.

Figura 34 – Quantidade de respostas por opções da questão: Q2 - I12



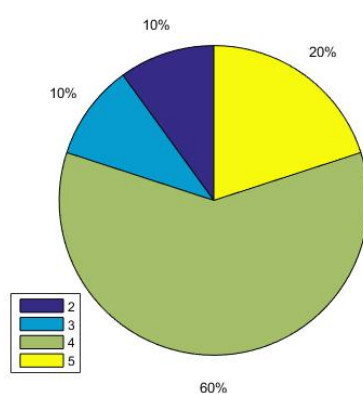
Fonte: o próprio autor.

Figura 35 – Quantidade de respostas por opções da questão: Q3 - I1



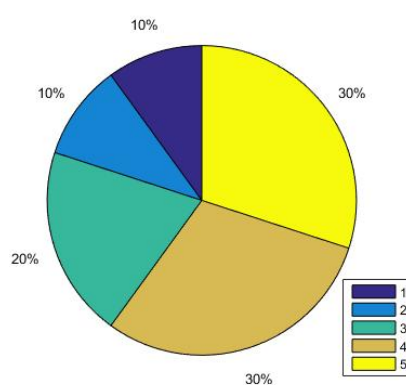
Fonte: o próprio autor.

Figura 36 – Quantidade de respostas por opções da questão: Q3 - I2



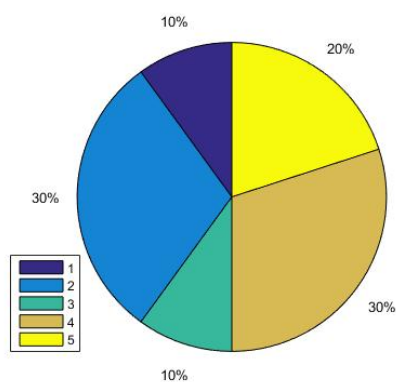
Fonte: o próprio autor.

Figura 37 – Quantidade de respostas por opções da questão: Q3 - I3



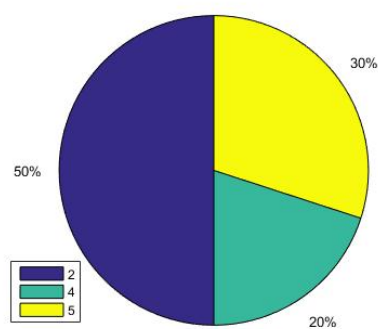
Fonte: o próprio autor.

Figura 38 – Quantidade de respostas por opções da questão: Q3 - I4



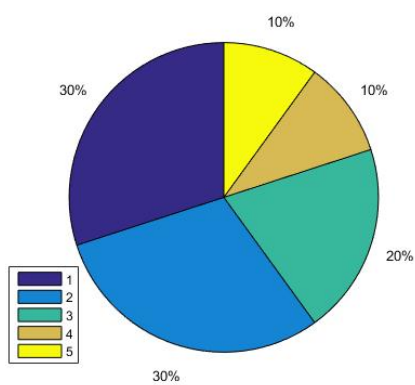
Fonte: o próprio autor.

Figura 39 – Quantidade de respostas por opções da questão: Q3 - I5



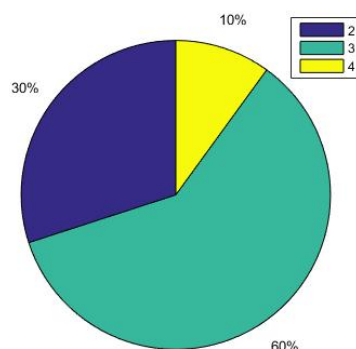
Fonte: o próprio autor.

Figura 40 – Quantidade de respostas por opções da questão: Q4 - I1



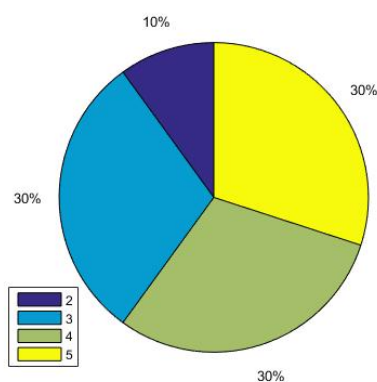
Fonte: o próprio autor.

Figura 41 – Quantidade de respostas por opções da questão: Q4 - I2



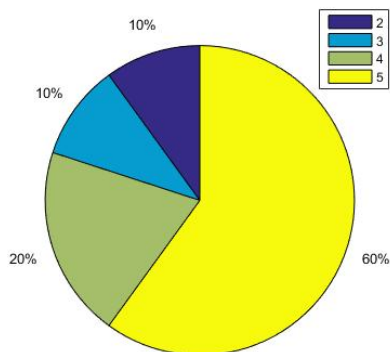
Fonte: o próprio autor.

Figura 42 – Quantidade de respostas por opções da questão: Q4 - I3



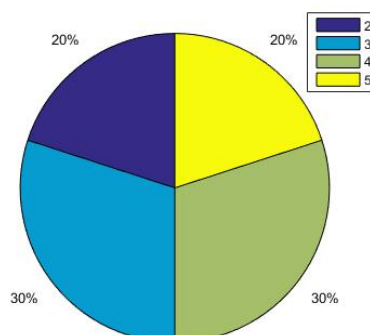
Fonte: o próprio autor.

Figura 43 – Quantidade de respostas por opções da questão: Q4 - I4



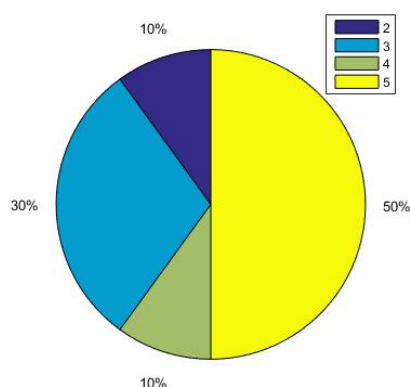
Fonte: o próprio autor.

Figura 44 – Quantidade de respostas por opções da questão: Q4 - I5



Fonte: o próprio autor.

Figura 45 – Quantidade de respostas por opções da questão: Q4 - I6



Fonte: o próprio autor.

Analisando essas respostas, foi possível perceber sintonia entre as dificuldades dos alunos da turma estudada e as dificuldades pelos trabalhos citados. Os itens I1 e I2 da questão Q1, estão relacionados a questão técnica da programação, e obtiveram as menores notas na média, o que leva a concluir que os alunos estão dominando tecnicamente a computação, ou seja, aprendendo a configurar computadores para desenvolvimento. Porém não estão dominando a lógica de programação da mesma maneira. Os itens I3 e I5 da questão Q1 são fortemente atrelados a capacidade lógica de programação, e foram os que obtiveram pior média nas respostas, alcançando a escala 3. Mas nenhum item chegou a ter média na escala 4 ou 5, o que demonstra que os alunos têm conseguido lidar com tais dificuldades ao longo do curso. As respostas a questão Q2 apenas confirmaram essas observações. Os itens I4, I6, I8, I9 e I11 estão fortemente associados as habilidades de programação, como construir programas completos e otimizados que previnem erros. Esses itens obtiveram as piores médias em suas respostas, mas essas também chegaram a escala maior que 3.

As duas últimas questões estão relacionadas a maneira como os alunos preferem de aprender: abordando mais teoria ou prática, fazendo muitos exercícios, estudando em turma ou sozinho. Foi constatado pelas respostas aos itens da questão Q3 que os alunos não têm preferência por teoria: a média de respostas no item I1 foi abaixo de 2. Nas médias dos demais itens fica claro que não possuem uma preferência exata por resolver exercícios de programação de maneira escrita ou no computador (itens I2 e I3), assim como estudar em turma ou sozinho (I4 e I5). O tamanho dos exercícios também parece não influenciar, pois exercícios práticos são mais trabalhos que listas teóricas, mas obtiveram a mesma média de preferência. As respostas a questão Q4 confirmou essas análises. O item I1 da questão Q4 aborda uma preferência teórica ao ensino, e foi o que obteve a pior média nas respostas. Pode se dizer que alunos não gostam de usar os livros e não sentem remorso por não utilizá-los para aprender programação. As médias aos itens I2 e I3 comprovam mais uma vez que gostam tanto de exercícios práticos como teóricos, e os itens I4 e I5 reforçam que preferem experimentar a programação do que apenas discutir conceitos.

Pode se perceber um certo tipo de perfil na turma:

- Conforme mais abstrato e lógico o tema de estudo da programação, sentem mais dificuldades
- São tecnicamente capazes de configurar ambientes para desenvolver
- São diretamente motivados por fazer exercícios e visualizar o que acontece nos programas

O desvio padrão para algumas respostas ultrapassou uma unidade. Isso pode ser explicado, pois a turma estudada possui alunos de diversos períodos, que não compartilharam os mesmos professores e metodologias de ensino, ao longo do progresso no curso. Essa explicação foi aceita por causa do que se ouviu dos alunos na apresentação da plataforma que será explicada a seguir.

5.2 Apresentação da plataforma aos alunos

A apresentação da plataforma foi feita no mesmo dia de aplicação do questionário de dificuldades aos alunos. Durou cerca de uma hora e quarenta minutos. Foi realizada pelo aplicador do questionário (o próprio autor deste trabalho), em uma sala com um projetor ou *datashow*. A apresentação começou abordando o conceito de *gamification*, depois explicou-se que uma estratégia de *gamification* foi implementada em uma plataforma de ensino de programação que substituiria o que eles estavam usando até o momento, por fim cada um dos elementos de *gamification* implementados foram exibidos e explicados. Antes de apresentar cada um dos elementos, foi perguntado aos alunos se alguns dos problemas de ensino percebidos pelos trabalhos citados eram condizentes com a realidade da turma. Isso

foi feito pois, foram esses problemas que inspiraram a implementação de tais elementos, então se esperava uma validação por parte dos alunos para tais escolhas, ou seja, se poderiam auxiliar na solução de tais problemas sobre a experiência deles. Quando o motivo de implementação do elemento não estava necessariamente relacionado a um desses problemas, era explicado qual a motivação de ter implementado o mesmo.

Primeiro foi perguntado aos alunos, se durante o curso, perceberam que investiram mais tempo aprendendo sintaxe de linguagens de programação e configurações de ambiente de desenvolvimento do que aprendendo a lógica de programação. A resposta dividiu os alunos, pois parte deles haviam feito as disciplinas introdutórias de programação com um professor que ensinou pseudocódigo e o restante fez as mesmas disciplinas programando em JAVA. Os alunos que haviam aprendido com pseudocódigo não tiveram que decorar particularidades de nenhuma linguagem, nem tiveram preocupação com demais configurações e relataram que isso ajudou na aprendizagem, o que colabora para o que [Lister \(2000\)](#) havia recomendado. Alguns dos alunos que aprenderam inicialmente a programar com JAVA relataram que sentiram dificuldades até se familiarizem com a linguagem. Foi então comunicado aos alunos que a plataforma que foi desenvolvida, para substituir a que usavam, permitia a eles programarem em qualquer linguagem, até mesmo português - que é um tipo de pseudocódigo, e todos aprovaram a ideia.

Em segundo, foi perguntado se alunos tinham dificuldades para achar materiais para estudarem fora da aula. Eles disseram que sim, e que nem sempre conseguem estudar pelo material disponibilizado pelo professor. Foi exposto a eles que plataforma desenvolvida têm seus próprios materiais, com textos, vídeos e exemplos de fontes em várias linguagens, organizados por cursos e tópicos. Os alunos aprovaram a implementação.

Em terceiro, foi perguntado aos alunos se eles tinham dificuldade com temas mais abstratos de programação e se tinham dificuldades em mensurar o quanto sabiam sobre determinado tema, o quanto confiavam na sua capacidade de programar. Responderam que não sabem mensurar com precisão quais habilidades desenvolveram e quanto sabem sobre determinado tema. Foi mostrado para eles que a plataforma pontua pontos de habilidades em programação que são conseguidos conforme o aluno que a utiliza resolve exercícios. O sistema de correção de tais exercícios iria se manter conforme a plataforma anterior, ou seja, um sistema automático, que submete a solução do aluno a testes e compara o resultado desses testes a uma resposta referência. Os alunos aprovaram a ideia, e sugeriram, que utilizando essa pontuação de habilidades, a plataforma classifica se o aluno, em seu nível de experiência em programação, como por exemplo: iniciante, regular, experiente e habilidoso.

Em quarto, foi perguntado se eles estudavam mais sozinhos ou em grupos. Mais da metade disse que comparava suas soluções com um amigo da turma. Foi explicado a eles que existia um fórum no site, para que pudessem criar comentários nos tópicos e exercícios, e também podiam adicionar amigos para conversar por *chat*. Eles também aprovaram a

ideia, e sugeriram que o *chat* tivesse a funcionalidade de conversa em grupos.

Em quinto, foi perguntado se conseguiam se divertir e entreter resolvendo os exercícios, ou se não conseguiam resolver tudo por falta de motivação. Responderam que nem sempre estão motivados, e normalmente a motivação é conseguir a média da disciplina. Foi exibido então para eles o quadro de medalhas da plataforma, foi explicado que algumas medalhas dependem do *rank*, outras das habilidades que aluno conseguiu, outras de quantidade de exercícios resolvidos e suas dificuldades. Eles aprovaram a implementação, e ficou evidente que esse foi o elemento de mais gostaram da plataforma. Isso era esperado, pois na revisão de Hamari, Koivisto e Sarsa (2014) e Majuri, Koivisto e Hamari (2018), esse elemento era o mais escolhido em experimentos com *gamification*.

Em sexto, foi perguntado se gostariam de saber seu progresso em relação a turma, antes das avaliações e notas serem lançadas pelo professor. Responderam que sim, mas que seria difícil realizar tal constatação. Foi explicado então que cada aluno na plataforma escolheria um nome fantasia e uma foto (avatar) para ser sua identidade secreta. Baseado no progresso do aluno, uma tabela de ranqueamento seria montada pela plataforma, exibindo apenas o nome fantasia do aluno e sua foto. Eles gostaram da ideia, desde que a identidade do aluno não pudesse ser descoberta pelos outros alunos; somente o professor e o próprio aluno poderiam saber. Eles também perguntaram se o critério de pontuação era o mesmo da plataforma anterior, pois tinham duas críticas ao mesmo:

- Crítica 1: na plataforma anterior, a pontuação do aluno era penalizada se sua solução não fosse validada em todos os testes. O que os inibia tentar. E os testes não eram visíveis aos alunos, então não podiam saber o que estavam errando. E achavam isso injusto, pois estavam em processo de aprender.
- Crítica 2: a plataforma atribuía pontos extras ao aluno que enviasse a solução primeiro. Eles também achavam injusto, pois nem todos tinham tempo livre, só podiam resolver os exercícios em determinado momento da semana.

Foi explicado que o *rank* utilizava apenas a pontuação e dificuldade do exercícios para gerar as pontuações, isso também foi aprovado por eles. Sugeriram, que a plataforma calculasse pontuações extras para soluções mais otimizadas. Foi constatado que a plataforma utilizada anteriormente na turma não era projetada para ambiente de ensino, mas para maratonas de programação, por isso a pontuação era penalizada em caso de erros, e acrescida se resolvida rapidamente. Os alunos também sugeriram um avatar com mais possibilidades de customização, que pudessem ser desbloqueadas a partir da pontuação do aluno. Confirmaram que isso prenderia a atenção deles.

Em sétimo, foi exibido a tela de *feedback* da plataforma, e explicado que a cada exercício que o aluno escolhe para resolver, ele pode ver a dificuldade do mesmo, e acompanhar a correção, caso o aluno submete uma solução para o exercício. A plataforma agrupa todos os exercícios que o aluno resolve, associados a seus devidos tópicos, e por

uma interface, o aluno pode ver se o exercício está aguardando para ser corrigido, se é acertou, ou se errou. Em uma outra interface mais detalhada, é exibido a solução do aluno, a situação da correção, e os testes a que foram usados para validar a solução. Se a solução do aluno não passar em algum teste, é exibido para ele a diferença entre as saídas do teste; o que a solução dele respondeu e o que era a resposta esperada. Em caso de um eventual erro de sintaxe na solução, este também seria exibido ao aluno, com detalhes (tipo de erro e em qual linha do fonte solução ele se localizava).

Por fim, foi comentado com os alunos, outras possibilidades de elementos que não foram implementadas na plataforma. Por exemplo, foi citado a possibilidade do aluno trocar pontos por dicas para resolver os exercícios, porém acharam melhor as dicas serem acessíveis baseadas na quantidade de exercícios que o aluno já resolveu. Também foi mencionado a possibilidade da pontuação expirar, ou ser descontada com erros em soluções. Mas não aprovaram, ficou bem claro que querem ter margem para poder errar sem serem penalizados por isso, não querem perder pontos. Todos os elementos da maneira que foram implementados agradaram aos alunos e estes confirmaram que influenciaram no processo de aprendizagem a programação.

5.3 Questionário sobre dificuldades no aprendizado de programação

O questionário para usabilidade educacional da plataforma acabou não se tornando crucial como planejado, pois na apresentação da plataforma os alunos interagiram de maneira que já deram as respostas dos itens abordados nesses questionário. Mesmo assim, o mesmo foi disponibilizado pela internet, para quem quisesse responder. Também não foi requerido do aluno nenhuma identificação. Dois alunos responderam o questionário. A seguir é apresentado o questionário, a média de resposta dos alunos e também o desvio padrão.

Questionário para avaliação do Mundo do Código quanto a experiência de usuário e aprendizagem

Orientação: Responda cada item com um número inteiro de 1 a 5 (1 é a nota mínima e 5 a máxima)

Itens sobre a usabilidade geral:

Questões emocionais

- I1 As tarefas do Mundo do Código são motivadoras para aprender mais sobre programação? Média de 5, desvio padrão de 0

- I2 As tarefas dentro do Mundo do Código são divertidas? Média de 3.5, desvio padrão de 0.70711
- I3 O Mundo do Código incentiva a participação? Média de 5, desvio padrão de 0
- I4 Aprender programação pelo mundo do código é empolgante? Média de 4, desvio padrão de 0

Engajamento

- I5 A tabela de rank, quadro de medalhas e pontos de habilidades te incentivaram? Média de 3, desvio padrão de 2.8284
- I6 Os materiais disponibilizados no Mundo do Código foram úteis? Média de 3.5, desvio padrão de 0.70711

Recurso/Apeação visual

- I7 O Mundo do Código incentiva a explorar/aprender sozinho? Média de 4.5, desvio padrão de 0.70711
- I8 As interfaces do Mundo do Código te agradaram? Média de 4, desvio padrão de 1.4142

Satisfação

- I9 Conseguiu se divertir enquanto aprendia? Média de 3.5, desvio padrão de 0.70711
- I10 Ficou motivado a aprender depois que usou o Mundo do Código? Média de 3.5, desvio padrão de 0.70711
- I11 O Mundo do Código te proporcionou sensação de conquista/progresso? Média de 4, desvio padrão de 0

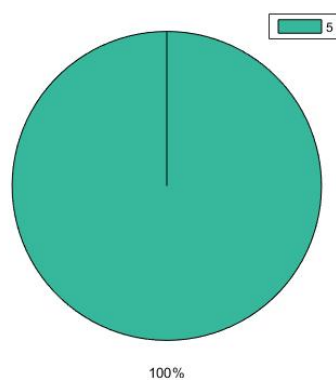
Itens sobre a usabilidade educacional:

Satisfação

- I12 Os exercícios do Mundo do Código são claros/objetivos? Média de 4.5, desvio padrão de 0.70711
- I13 Se sentiu confortável para errar e aprender com os próprios erros no Mundo do Código? Média de 4, desvio padrão de 1.4142
- I14 O Mundo do Código te ofereceu maneiras para tirar dúvidas e corrigir erros? Média de 4, desvio padrão de 0
- I15 As correções dos exercícios do Mundo do Código te agradaram? Média de 3, desvio padrão de 1.4142
- I16 As correções dos exercícios do Mundo do Código são claras? Média de 3.5, desvio padrão de 2.1213

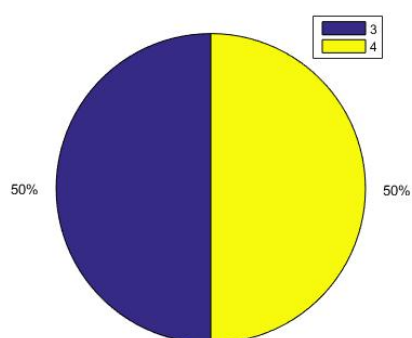
A seguir são apresentados gráficos com a quantidade de respostas por opção, para cada um dos item.

Figura 46 – Quantidade de respostas por opções do item: I1



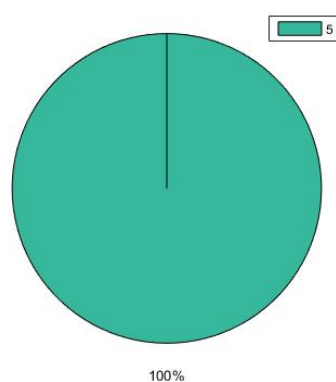
Fonte: o próprio autor.

Figura 47 – Quantidade de respostas por opções do item: I2



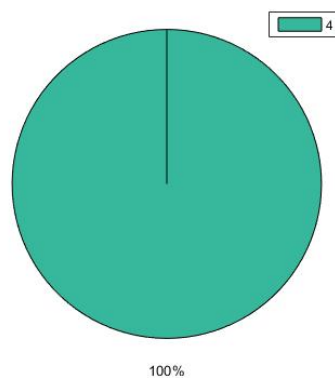
Fonte: o próprio autor.

Figura 48 – Quantidade de respostas por opções do item: I3



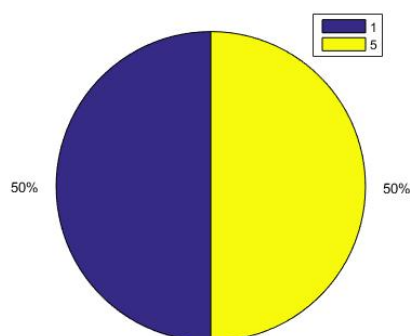
Fonte: o próprio autor.

Figura 49 – Quantidade de respostas por opções do item: I4



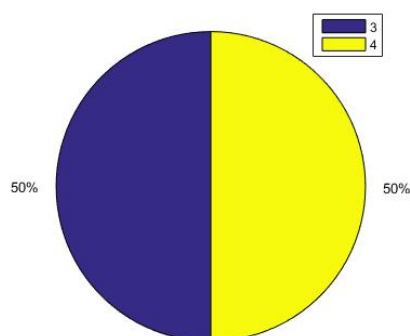
Fonte: o próprio autor.

Figura 50 – Quantidade de respostas por opções do item: I5



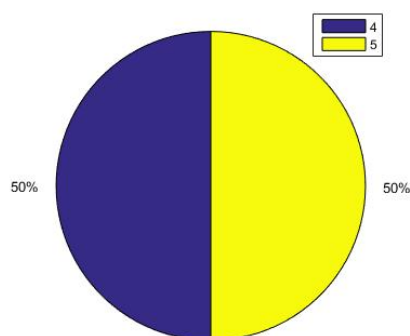
Fonte: o próprio autor.

Figura 51 – Quantidade de respostas por opções do item: I6



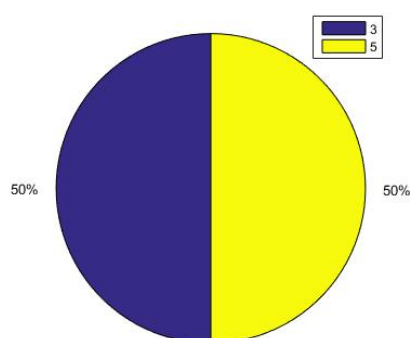
Fonte: o próprio autor.

Figura 52 – Quantidade de respostas por opções do item: I7



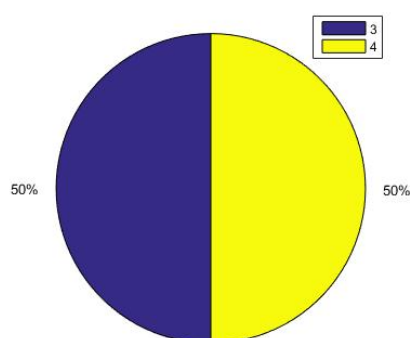
Fonte: o próprio autor.

Figura 53 – Quantidade de respostas por opções do item: I8



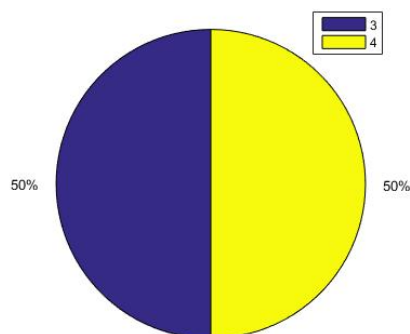
Fonte: o próprio autor.

Figura 54 – Quantidade de respostas por opções do item: I9



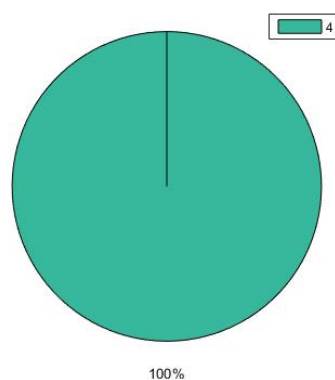
Fonte: o próprio autor.

Figura 55 – Quantidade de respostas por opções do item: I10



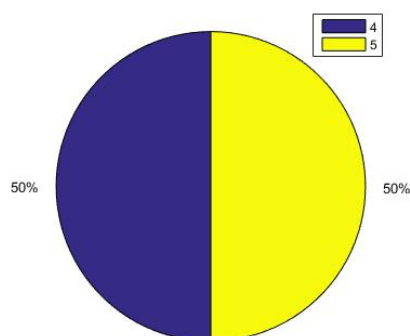
Fonte: o próprio autor.

Figura 56 – Quantidade de respostas por opções do item: I11



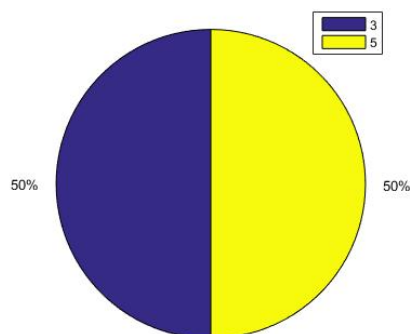
Fonte: o próprio autor.

Figura 57 – Quantidade de respostas por opções do item: I12



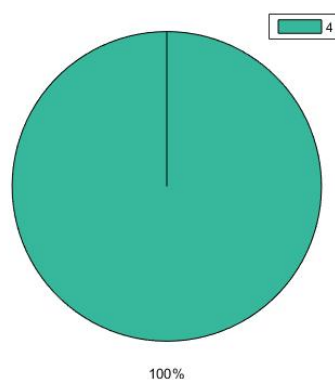
Fonte: o próprio autor.

Figura 58 – Quantidade de respostas por opções do item: I13



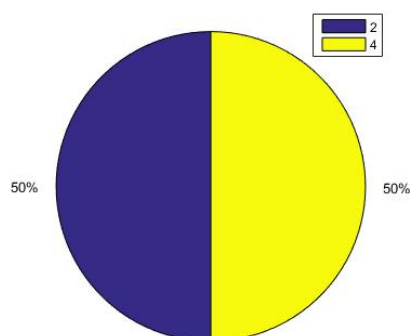
Fonte: o próprio autor.

Figura 59 – Quantidade de respostas por opções do item: I14



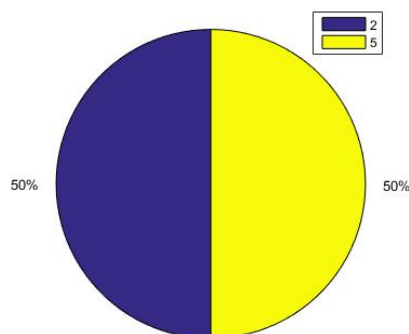
Fonte: o próprio autor.

Figura 60 – Quantidade de respostas por opções do item: I15



Fonte: o próprio autor.

Figura 61 – Quantidade de respostas por opções do item: I16



Fonte: o próprio autor.

Nenhuma média foi abaixo de 3, isso significa que a plataforma obteve uma boa usabilidade no geral. Alguns itens obtiveram o desvio maior do que 1, o que é uma grande oscilação. Um desses itens é o I5, que trata diretamente dos elementos de *gamification* que foram abordados. O que justifica isso, é o fato de alguns alunos ficarem desconfortáveis com a ideia de competição proporcionada pela plataforma. O item I8 também obteve um desvio acima de 1, esse item trata das interfaces do sistema. Durante a apresentação da plataforma, alguns alunos manifestaram crítica quando a facilidade de uso, o que está coerente com esse desvio. O item I13 confirma de certa maneira a competição sentida pelos alunos no item I5, pois exposto que não havia penalidades para erros na plataforma, o único fator que poderia causar medo de errar ao aluno, seria a competição. Por fim, os itens I15 e I16 também obtiveram desvios altos, e esses são cruciais, pois estão diretamente relacionados com a correção do exercício, ou seja, o elemento de *feedback*. Reconhece-se que este elemento precisa ser melhorado, justamente por sua importância.

6 Conclusão

Através deste trabalho foi possível perceber que existem dificuldades relacionadas ao processo de aprendizagem de programação específicas da área, que são inerentes a muitos fatores considerados importantes pelo senso comum, como, por exemplo, a escolha da linguagem de programação ou uso de certas ferramentas ou *softwares*. Tais dificuldades são notadas por diversos autores de diferentes lugares na literatura.

Também foi possível perceber que existe um elo entre o aprendizado e a *gamification*, porém ressalvas precisam ser feitas para que tal relação seja saudável aos alunos. Entre elas, deve se tomar cuidado para não penalizar o aluno de maneira que o mesmo tenha medo de errar. Em demais jogos ou atividades, pressupõem-se que o jogador tem as habilidades necessárias para vencer o jogo, e para que tal jogador entenda que deve usar ao máximo sua capacidade, penalidades são usadas para motivar o mesmo a não errar. Porém no ensino o erro é importante, pois muitas vezes é este que constrói um conhecimento mais maduro, e diferentemente da maioria dos jogos, o aluno ainda não desenvolveu as habilidades para jogar seriamente, na verdade o objetivo do jogo no ensino é desenvolver essas habilidades, não apenas resolver muitos exercícios e alcançar o topo do *rank* com todas as medalhas disponibilizadas no Mundo do Código.

Outra ressalva, é que na maioria dos jogos têm-se regras rígidas, para que os jogadores fiquem delimitados a certos espaços ou possibilidades. Isso é para que a disputa no jogo fique mais justa. Mas no ensino, como já demonstrado, a disputa não é o fator primordial. Antes da mesma, o fator de exploração tem mais valor, pois diante de todo conteúdo que o aluno vai aprender, seria melhor que o mesmo determinasse seu ritmo de aprendizado. O que não é possível em sala de aula, mas que a *gamification* pode proporcionar, como foi feito na plataforma. Assim, o aluno cria mais confiança, fica mais motivado e pode descobrir suas habilidades únicas, pois não estará severamente preso ao cronograma do professor, ou a um roteiro de estudo do mesmo. Logo os elementos imersivos ganham mais valor que os competitivos, por exemplo, a narrativa e *feedback* possuem mais relevância que as pontuações e *ranks*.

Uma última ressalva, é que existe um impacto direto de cada elemento de *gamification* ao aprendizado, e cada dinâmica elaborada precisa tratar esses elementos de maneira individual para que seja possível perceber isso. No protótipo desenvolvido, as dinâmicas funcionavam em conjunto, e o aluno não podia escolher ou ver isoladamente o funcionamento da mesma. No fim dos testes com a turma, percebeu-se que seria melhor deixar ao aluno escolher quais as dinâmicas que ele quer ser inserido na plataforma. Por exemplo, deixar a participação no *rank* ser opcional. Isso se faz interessante pois um elemento pode

agir em contraste com outro, e na visão do aluno, todas as dinâmicas são uma atividade só. Logo se ele tiver uma experiência ruim causada por um elemento, todos os outros serão incluídos nessa experiência.

Assim percebeu-se que a *gamification* é uma estratégia factível de apoio ao ensino e possui boa recepção por parte dos alunos.

6.1 Trabalhos Futuros

Esse trabalho não teve como escopo preparar os materiais das disciplinas de maneira que melhor aproveitassem os elementos de *gamification*. Trabalhos complementares podem trabalhar tais materiais de maneira que os tópicos e exercícios de um curso, possuam várias versões de enunciados e materiais, buscando atingir todos os tipos de perfis de estudantes. Fortalecendo assim os elementos imersivos da *gamification*.

Quanto aos perfis de aprendizado e jogadores, também não explorado tais conceitos na literatura, pois não existe uma relação direta entre tais perfis, e nem existe uma relação dos mesmos com elementos de *gamification*. Portanto esses temas não foram levados em consideração durante a escolha dos elementos utilizados de *gamification* utilizados no protótipo, para contornar essa possível lacuna, um questionário que avalia cada um dos elementos de *gamification* utilizados pelos alunos foi elaborado e respondido pelos mesmos, como já apresentado. De qualquer maneira, o objetivo da proposta de *gamification* neste trabalho é incluir o máximo de perfis de aprendizado e jogadores, o estudo dos perfis teria mais aproveitamento caso a plataforma fosse destinada a um grupo seleto de indivíduos. Admite-se que esse fator pode ter impactado na experiência dos alunos com o protótipo, trabalhos futuros podem levar a questão dos perfis de maneira mais direta, desenvolvendo dinâmicas específicas para os tais. É necessário mais trabalhos que relacionem de maneira direta os perfis de jogadores, como os expostos por: [Bartle \(1996\)](#), [Nacke et al. \(2011\)](#) e [Yee \(2006\)](#), com os perfis de aprendizado, como os abordados por: [Kolb \(1984\)](#), [Gregorc \(1979\)](#), [Felder e Silverman \(1988\)](#), [Fleming \(2001\)](#) e [Dunn \(1978\)](#), para que outros trabalhos que realizem as atividades que envolvem *gamification*, possam escolher elementos de jogos mais precisamente, assim como projetar as devidas mecânicas, dinâmicas e estéticas.

Observou-se também, a necessidade de se desenvolver melhores classificações para os enunciados dos exercícios e suas respectivas dificuldades, assim como para os casos de teste. Isso é de grande valia para a construção do curso, e para os elementos de ranque e nivelamento de *gamification*; como já explicado, se as pontuações forem muito fáceis, ou muito difíceis, ou pouco interessantes, o jogador se desinteressará do aprendizado. É necessário determinar com clareza quais critérios influenciam na construção dessas pontuações para o caso específico de *gamification* para o aprendizado, por exemplo,

durante a elaboração destes elementos neste trabalho, as seguintes perguntas surgiram:

- A pontuação deve ser dinâmica ou estática? Ou seja, conforme o aluno resolve ou erra exercícios, a pontuação adquirida nos próximos exercícios deve mudar?
- Existem dificuldades apenas para os exercícios? Ou para os seus testes também? O enunciado do exercício também impõem alguma facilidade ou dificuldade? Se sim, isso merece ser pontuado?
- A pontuação deve ser fixa no tempo ou expirar após um intervalo de inatividade por parte do jogador?

É possível perceber uma gama de possibilidades, e todas influenciam fortemente a experiência do jogador. Esse trabalho não contemplou todas as possibilidades; como foi demonstrado, a pontuação é fixa no tempo, estática, e foram definidas dificuldades apenas para os exercícios e seus casos de teste, ou seja, uma classificação bem simples, embora teve como base exercícios de maratona de programação.

Este trabalho também pode ser ampliado, incluindo mais elementos e dinâmicas de *gamification* e testado em mais disciplinas ou mesmo outras áreas de ensino. É possível testar para outras áreas de ensino, pois o único elemento que precisaria ser reprojetoado é o *feedback* automatizado. E vale destacar que o protótipo foi testado em curto período de tempo: apenas um mês, apenas para o tópico de grafos da disciplina escolhida, testes com maiores durações e para diferentes tópicos também precisam ser feitos, para reforçar as conclusões deste trabalho.

Referências

- ANGELO, M. F.; BERTONI, F. C.; SANTOS, J. A. M.; LOULA, A. C. Análise da Aplicação do Método PBL no Ensino de Programação em Engenharia de Computação. PBL 2010 Congresso Internacional. São Paulo, fev. 2010. Nenhuma citação no texto.
- BARTLE, R. Hearts, clubs, diamonds, spades: players who suit MUDs. *Journal of Virtual Environments*, v.1, n.1, 1996. Citado na página 64.
- BENNEDSEN, J. **Teaching and Learning Introductory Approach - A Model-Based Approach**. 227 p. Tese (Doutorado), 2008. Citado na página 1.
- BENNEDSEN, J.; CASPERSEN, M. E.; Revealing the programming process. p. 186-190, 2005. Citado na página 1.
- BUSARELLO, Raul Inácio. Gamification: princípios e estratégias. Pimenta cultural, 2016. Citado 5 vezes nas páginas vii, 2, 4, 21 e 22.
- CAPORARELLO, L.; MAGNI, M; PENNAROLA, F. Learning and gamification: a possible relationship? 2017. Citado 2 vezes nas páginas vii e 26.
- CASTRO, Lúcia Mara Drumond. **A gamificação como método de aprendizagem para a geração Z - uma aplicação em fundamentos de programação I**. 2018. Monografia (Engenharia de computação) - Centro Federal de Educação Tecnológica de Minas Gerais, Timóteo, 2018. Citado na página 3.
- DEHNADI, S.; BORNAT, R. The camel has two humps (working title). p. 1-21, 2006. Citado 3 vezes nas páginas 1, 7 e 9.
- DETERS, J. I.; DA SILVA, J. M. C.; DE MIRANDA, E. M.; FERNANDES, A. M. R. O Desafio de Trabalhar com Alunos Repetentes na Disciplina de Algoritmos e Programação. Simpósio Brasileiro de Informática na Educação. 2008. Citado 3 vezes nas páginas 1, 9 e 10.
- DETERDING, S. et al. From game design elements to gamefulness: defining gamification. In: ACM. Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments. [S.l.], 2011. Citado 5 vezes nas páginas vii, 2, 4, 12 e 13.
- DOMÍNGUEZ, Adrián; NAVARRETE, Joseba Saenz; MARCOS, Luis; SANZ, Luis Fernández; PAGÉS, Carmen; HERRÁIZ, José-Javier Martínez. Gamifying learning experiences: Practical implications and outcomes, 2013. Citado na página 20.
- DUNN, R.; DUNN, K. Teaching students through their individual learning styles: a practical approach. Reston, VA: Reston Publishing Co., 1978. Citado na página 64.

- FAIELLA, Filomena; RICCIARDI, Maria. Gamification of learning: a review of issues and research, 2015. Citado na página [25](#).
- FASSBINDER, A. G. de O.; PAULA, L. C de; ARAÚJO, J. C. D.; Experiências no estímulo à prática de Programação através do desenvolvimento de atividades extracurriculares relacionadas com as competições de conhecimentos. XX Workshop sobre Educação em Informática. 2012. Nenhuma citação no texto.
- FELDER, R. M.; SILVERMAN, L. K. Learning styles and teaching styles in engineering education. International Journal of Engineering Education, Ontario, v. 78, n. 7, p. 674–681, 1988. Citado 2 vezes nas páginas [2](#) e [64](#).
- FLEMING, N. D. Teaching and learning styles: VARK strategies. Christchurch, New Zealand: N. D. Fleming, 2001. Citado na página [64](#).
- GIRAFFA, L. M. M.; MORA, M. d. C. Evasão na disciplina de algoritmo e programação: Um estudo a partir dos fatores intervenientes na perspectiva do aluno. 2013. Nenhuma citação no texto.
- GOMES, A.; HENRIQUES, J.; MENDES, A. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. V. 1, p. 93-103, 2008. Citado 2 vezes nas páginas [1](#) e [8](#).
- GREGORC, A. F. Learning/teaching styles: their nature and effects. NASSP Monograph, 1979. Citado na página [64](#).
- HAMARI, J.; KOIVISTO, J.; SARSA H. Does Gamification Work? A Literature Review of Empirical Studies on Gamification, 2014. Citado 9 vezes nas páginas [xiii](#), [2](#), [23](#), [54](#), [82](#), [83](#), [84](#), [85](#) e [86](#).
- HARPUR, P. A.; DE VILLIERS, M. R. MUUX-E, a framework of criteria for evaluating the usability, user experience and educational features of m-learning environments. 2015. Citado na página [28](#).
- HUIZINGA, Johan. Homo Ludens, 2000. Citado 2 vezes nas páginas [11](#) e [12](#).
- HUNICKE, R.; LEBLANC, M.; ZUBEK, R. MDA: A formal approach to game design and game research. 2004. Citado 4 vezes nas páginas [vii](#), [13](#), [14](#) e [26](#).
- KELLEY, David. The Art of Reasoning, 1988. Citado na página [11](#).
- KOLB, D. A. Experimental learning: experience as the source of learning and development. New Jersey: Prentice-Hall, Englewood Cliffs, 1984. Citado na página [64](#).
- LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M. A study of the difficulties of novice programmers. v. 37, p. 14, 2005. Citado 3 vezes nas páginas [1](#), [10](#) e [31](#).

- LAZZARO, N. Why we play games: Four keys to more emotion without story. 2004. Citado na página 2.
- LISTER, R. On blooming first year programming, and its blooming assessment. In: **Proceedings of the Australasian conference on Computing education - ACSE**. [S.l.]: ACM Press, 2000. p. 158-162. Citado 5 vezes nas páginas vii, 1, 6, 7 e 53.
- MARIN, M. A. M.; KLIEMANN, M. P. Reflexões sobre o uso da tecnologia computacional na educação. 2014. Nenhuma citação no texto.
- MCGONIGAL, J. Reality is broken: Why games make us better and how they can change the world. [S.l.]: Penguin, 2011. Citado na página 2.
- MAJURI, J.; KOIVISTO, J.; HAMARI, J. Gamification of education and learning: A review of empirical literature, 2018. Citado 3 vezes nas páginas 2, 26 e 54.
- MARCONI, M. D. A.; LAKATOS, E. M.; Metodologia do Trabalho Científico. 7a. ed. São Paulo: Atlas, 2012. 225 p. Citado na página 29.
- NACKE, L. E.; BATEMAN, C.; MANDRYK, R. L. BrainHex: preliminary results from a neurological gamer typology survey. In Proceedings of the international conference on Entertainment Computing, LNCS 6972, 288–293, 2011. Citado na página 64.
- PRENSKY, M. The motivation of gameplay: The real twenty-first century learning revolution. On the horizon, MCB UP Ltd, v. 10, n. 1, p. 5-11, 2002. Nenhuma citação no texto.
- ROCHA, P. S.; FERREIRA, B.; MONTEIRO, D.; NUNES, D. S. C.; GÓES, H. C. N.; Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. Cinted-UFRGS, v. 8, n. 3, p.1-11, Porto Alegre, dez/2010. Citado na página 2.
- SCHOLZTZ, Brenda; RAGA, Larissa; BAXTER, Gavin. Design and Evaluation of a “Gamified” System for Improving Career Knowledge in Computing Sciences. 2016. Citado 2 vezes nas páginas 27 e 39.
- SIGNORI, G.; GUIMARÃES, J. C. F. de. Gamificação como método de ensino inovador. International Journal on Active Learning, v. 1, n. 1, p. 66-77, 2016. Citado 2 vezes nas páginas 5 e 6.
- SILVA, Nícollas Andrade. **Dificuldades de aprendizagem em disciplinas iniciais de programação de computadores no curso de engenharia de computação do CEFET-MG - campus Timóteo**. 2018. Monografia (Engenharia de computação) - Centro Federal de Educação Tecnológica de Minas Gerais, Timóteo, 2018. Citado 3 vezes nas páginas 2, 3 e 8.

- Sindicato das Entidades Mantenedoras de Estabelecimentos de Ensino Superior no Estado de São Paulo (Brasil). Índice de evasão de alunos é maior na área de tecnologia da informação. Disponível em: <<http://g1.globo.com/sp/sao-carlosregiao/noticia/2012/09/indice-de-evasao-de-alunos-e-maior-na-area-de-tecnologiadainformacao.html>>. Citado na página 1.
- WAZLAWICK, R.S. (2009).; Metodologia de pesquisa para Ciência da Computação, 184p. Editora Campus/Elsevier. ISBN: 9788535235227. Citado na página 29.
- YEE, N. Motivations for play in online games. *Cyberpsychology & Behavior: The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society*, 9(6), 772-775, 2006. Citado na página 64.
- ZICHERMANN, G.; CUNNINGHAM, C. Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media, 2011. (O'Reilly Series). ISBN 9781449397678. Disponível em: <https://books.google.com.br/books?id=Hw9X1miVMMwC> Citado 8 vezes nas páginas vii, 3, 14, 15, 16, 17, 18 e 19.

Apêndices

APÊNDICE A – Interfaces do Mundo do Código

Este apêndice apresentará imagens das interfaces implementados no Mundo do Código. As informações representadas nas imagem não são coerentes.

A.1 Rank

Figura 62 – Tabela de rank na plataforma.

Ludenos

CURSOS FEEDACK HABILIDADES MEDALHAS **RANK**

Mrs. Aluno ▾

Rank



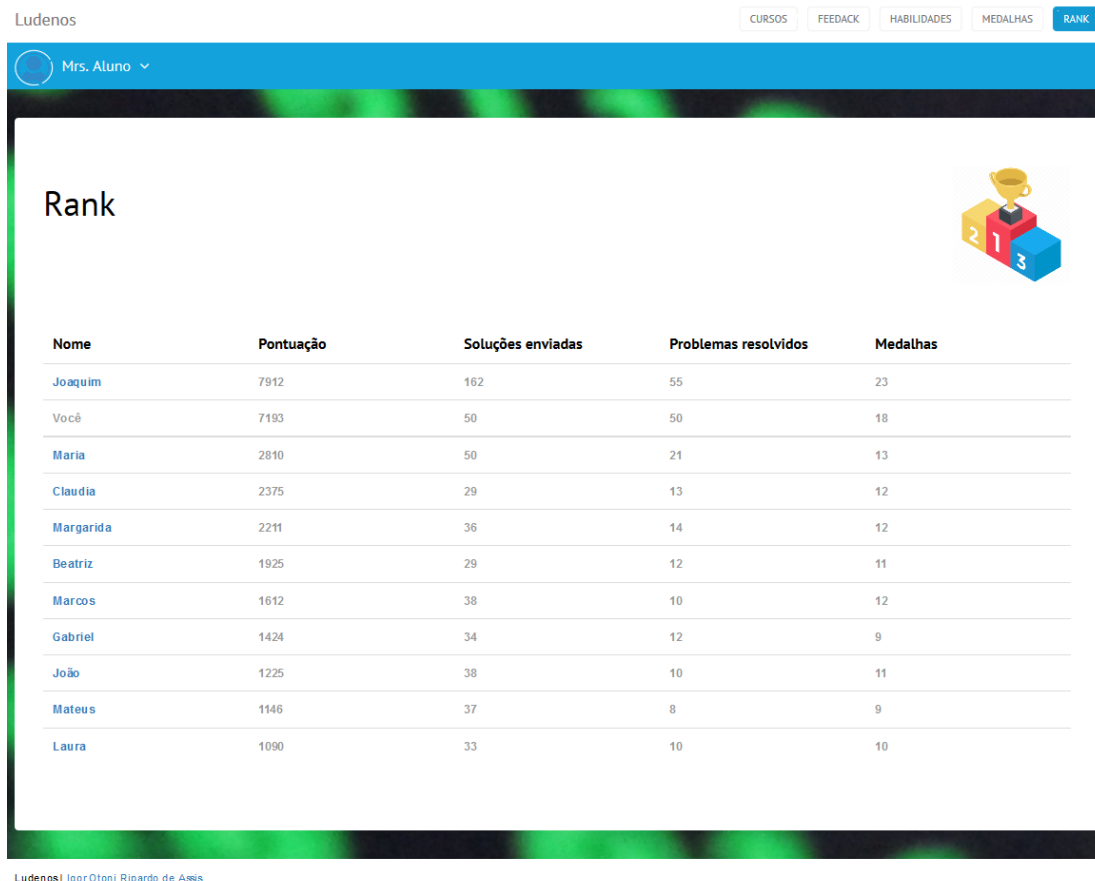
Nome	Pontuação	Soluções enviadas	Problemas resolvidos	Medalhas
Joaquim	7912	162	55	23
Você	7193	50	50	18
Maria	2810	50	21	13
Claudia	2375	29	13	12
Margarida	2211	36	14	12
Beatriz	1925	29	12	11
Marcos	1612	38	10	12
Gabriel	1424	34	12	9
João	1225	38	10	11
Mateus	1146	37	8	9
Laura	1090	33	10	10

Ludenos | Igor Ottoni Ripardo de Assis

Fonte: o próprio autor.

A.2 Pontuação de habilidades

Figura 63 – Pontuação de habilidades em programação na plataforma.



Fonte: o próprio autor.

A.3 Medalhas

Figura 64 – Medalhas na plataforma.

Ludenos

CURSOS FEEDBACK HABILIDADES **MEDALHAS** RANK

Mrs. Aluno ▾

Envios\Soluções: 50\50

Pontuação: 7193

Dificuldade	Quantidade de exercicios resolvidos
Desafio	13
Difícil	16
Fácil	14
Mediano	7

Medalhas



Que comecem os jogos!
Resolveu o primeiro problema!



Nada mal para um iniciante!
Resolveu 10 problemas!



Me sinto em casa!
Resolveu 25 problemas!



A parada é séria!
Resolveu 50 problemas!



Testador!
Resolveu 1 problema de cada dificuldade!



Veterano!
Resolveu 5 problemas de cada dificuldade!



Bem vindo ao Ludenos!
Enviou sua primeira solução!



Motivado!
Enviou 10 soluções!



Engajado!
Enviou 25 soluções!



Nerd!
Enviou 50 soluções!



Estagiário!
Desenvolveu a primeira habilidade!



Júnior!
Desenvolveu 2 habilidades!

Fonte: o próprio autor.

A.4 Feedback

Figura 65 – Feedback na plataforma.

Ludenos

CURSOS FEEDBACK HABILIDADES MEDALHAS RANK

Mrs. Aluno ▾

Feedback

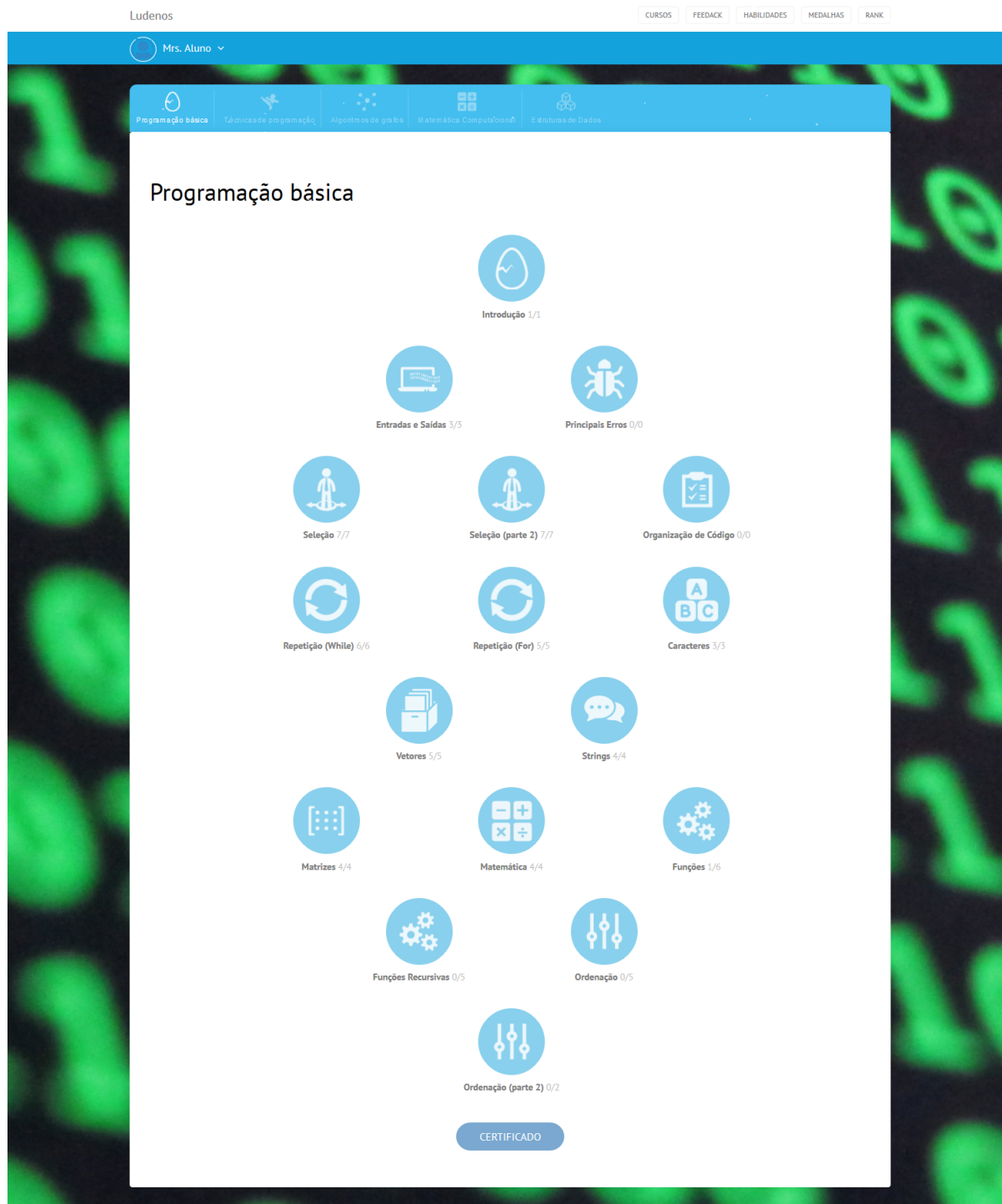


Exercício	Tópico	Dificuldade	Situação
Olá Ludenos!	Introdução	Desafio	Resolvido
Soma	Entradas e Saídas	Fácil	Resolvido
Divisão	Entradas e Saídas	Fácil	Resolvido
Tomadas	Entradas e Saídas	Mediano	Resolvido
Meio aluno	Seleção	Fácil	Resolvido
Maio área	Seleção	Fácil	Resolvido
Filper	Seleção	Difícil	Resolvido
Medalhas	Seleção	Desafio	Resolvido
Fuga com helicóptero	Seleção	Fácil	Resolvido
Andando no tempo	Seleção	Fácil	Resolvido
Carnaval	Seleção	Mediano	Resolvido
Aprovado ou Reprovado	Seleção (parte 2)	Fácil	Resolvido
Medalhas Olímpicas	Seleção (parte 2)	Fácil	Resolvido
Tira-teima	Seleção (parte 2)	Difícil	Resolvido
Campeonato	Seleção (parte 2)	Difícil	Resolvido
Zerinho ou Um	Seleção (parte 2)	Fácil	Resolvido
Gangorra	Seleção (parte 2)	Desafio	Resolvido
Cobra Coral	Seleção (parte 2)	Desafio	Resolvido
Todos os Divisores	Repetição (While)	Difícil	Resolvido

Fonte: o próprio autor.

A.5 Estrutura de um curso

Figura 66 – Estrutura de um curso na plataforma.



Fonte: o próprio autor.

A.6 Tópico de um curso

Figura 67 – Estrutura de um tópico na plataforma.



Ludenos

CURSOS FEEDBACK HABILIDADES MEDALHAS RANK

Mrs. Aluno 1/1

Introdução

Assistir mais tarde Compartilhar

Introdução

Programação Básica

A primeira pergunta que as pessoas fazem quando decidem começar a estudar informática é: o que preciso para fazer um programa? É completamente possível escrever e compilar um arquivo em qualquer linguagem usando o bloco de nota e um terminal do computador, mas é bem melhor usar ferramentas que facilitem esse processo. O ideal para quem começa é usar um programa que junta **IDE** e compilador. IDE (Integrated Development Environment) é onde vamos escrever o programa, ela é feita para facilitar a leitura, correção e organização do código, bem como facilitar o **debug** (busca por erros após a criação do programa). Escrito o código, o compilador vai transformar os comandos que estão escritos usando alguma linguagem de programação em algo que o computador entenda e então criar o programa.

Neste curso, a linguagem utilizada será o **C++**. A IDE é de escolha do leitor, mas nós recomendamos, e iremos utilizar durante os exemplos, o **CodeBlocks**, que já vem com compilador. Se quiser baixá-la, clique [aqui](#). Outra IDE muito utilizada é o Dev-C++, clique [aqui](#) caso você deseje utilizá-la. Independente da escolha, a instalação é feita de forma bem simples (normalmente pressionando "Next, Next, Next...").

Agora que você já tem o que precisa, vamos começar.

Para começarmos a escrever nosso primeiro programa devemos ir no menu "File" do CodeBlocks e em seguida "New" e "Empty file".

Salve seu arquivo com o nome que desejar e o computador criará um programa com nome semelhante. **Se você está usando o CodeBlocks, não esqueça de escrever ".cpp" ao fim do nome do arquivo**, para defini-lo como do tipo C++.

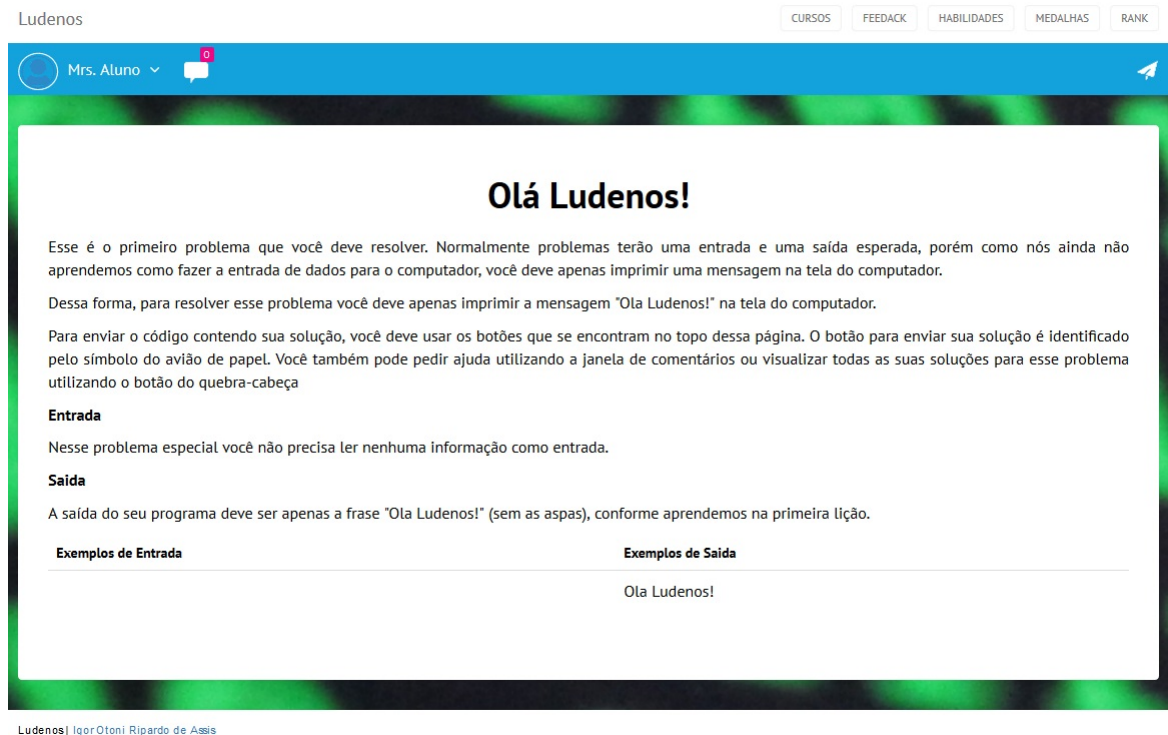
O primeiro programa que todos faremos é simplesmente imprimir algo na tela. Para começar, assim como começaríamos qualquer outro, vamos declarar as bibliotecas que usaremos. As bibliotecas são espécies de dicionários com alguns comandos. Assim, quando usamos um comando no programa, é preciso que a biblioteca em que ela se encontra esteja declarada no começo do código para que o computador saiba o que significa aquele comando.

Se o nome de uma biblioteca é *name*, para declará-la escrevemos **#include <name>**. É importante saber que o computador executa o que o mandamos fazer na ordem em que os comandos são escritos, por isso é importante declararmos as bibliotecas que usaremos no começo do código, pois não podemos usar uma função (um tipo de comando) antes de dizer para o computador a biblioteca dela. Para imprimir alguma frase na tela, usaremos uma função chamada **cout**, que está na biblioteca **iostream**.

Fonte: o próprio autor.

A.7 Exercício de um tópico

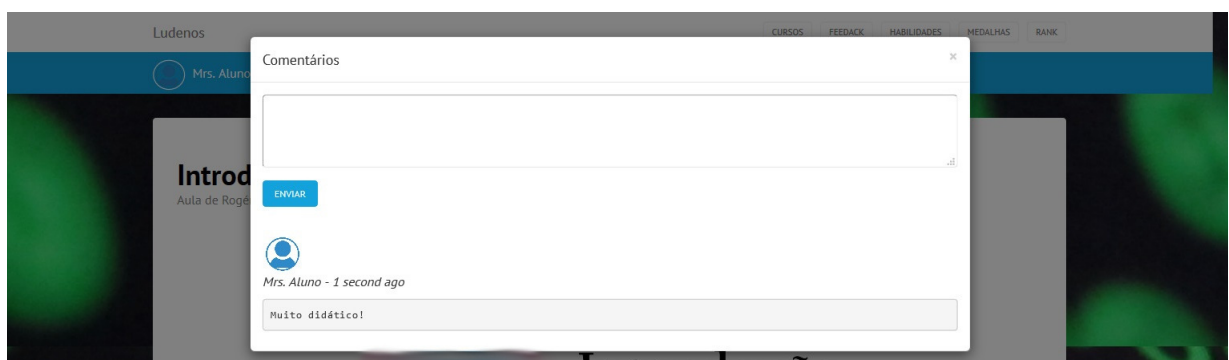
Figura 68 – Estrutura de um exercício na plataforma.



Fonte: o próprio autor.

A.8 Comentário de um tópico

Figura 69 – Comentário de tópico (ou exercício) na plataforma.



Fonte: o próprio autor.

A.9 Solução de um exercício

Figura 70 – Solução de um exercício na plataforma.

Ludenos CURSOS FEEDBACK HABILIDADES MEDALHAS RANK

Mrs. Aluno

Soma

ARQUIVO COM FONTE DA SOLUÇÃO:

No file selected.

Fonte:

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {
    int x = *argv[1] - '0'; int y = *argv[2] - '0';
    cout<<x+y<<"\n";
}
```

Resultado da execução:

Teste 1

Entradas

1 0

Saída resultante

1

Saída esperada

1

Correto

Teste 2

Entradas

-1 -1

Saída resultante

-6

Saída esperada

-2

Errado

Teste 3

Entradas

5 5

Saída resultante

10

Saída esperada

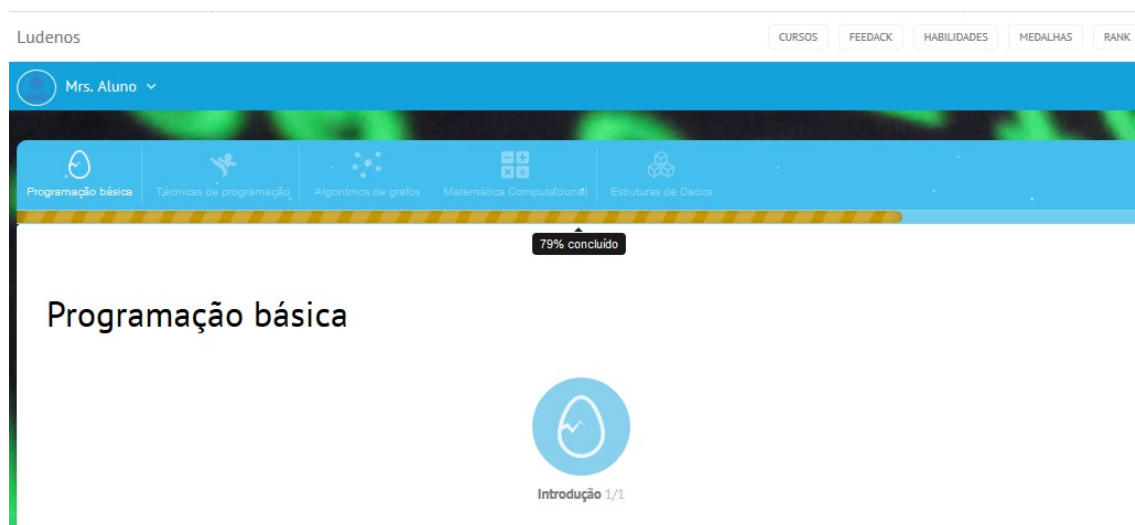
10

Correto

Fonte: o próprio autor.

A.10 Barra de progresso do curso

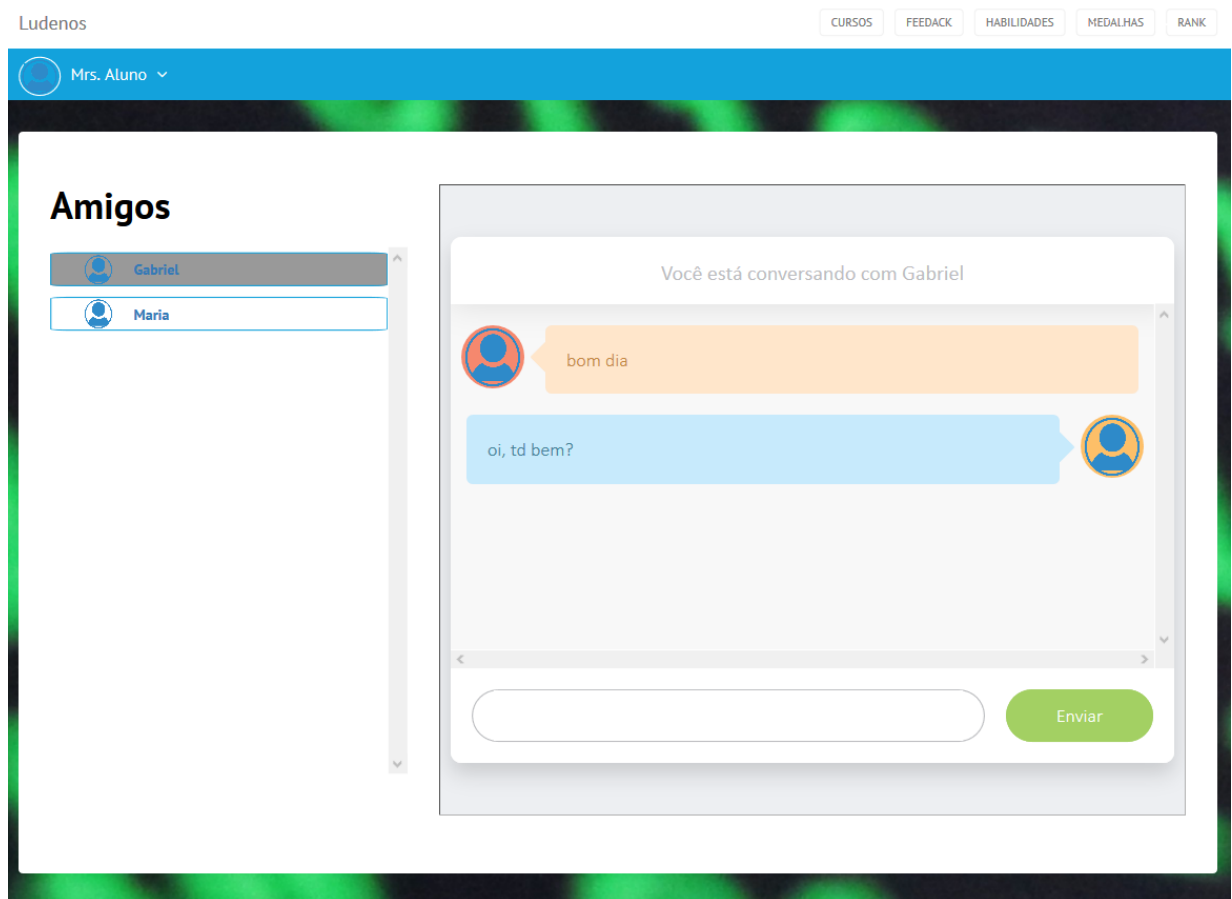
Figura 71 – Barra de progresso na plataforma.



Fonte: o próprio autor.

A.11 Chat entre amigos

Figura 72 – Chat entre amigos na plataforma.



Ludenos | Igor Ottoni Ripardo de Assis

Fonte: o próprio autor.

Anexos

ANEXO A – Anexos da revisão bibliográfica de Hamari, Koivisto e Sarsa (2014)

Nesta seção são apresentados anexos da revisão bibliográfica de Hamari, Koivisto e Sarsa (2014).

A.0.1 Referências dos trabalhos estudados

1. A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Steering user behavior with badges", In Proceedings of the 22nd international conference on World Wide Web, May 13-17, 2013, Rio de Janeiro, Brazil, pp. 95-106. (ACM)
2. C. Cheong, F. Cheong, and J. Filippou, "Quick Quiz: A Gamified Approach for Enhancing Learning", In Proceedings of Pacific Asia Conference on Information Systems, June 18-22, 2013, Jeju Island, Korea. (AISel)
3. P. Denny, "The Effect of Virtual Achievements on Student Engagement", In Proceedings of CHI 2013: Changing Perspectives, April 27-May 2, 2013, Paris, France, pp. 763-772. (ACM)
4. A. Domínguez, J. Saenz-de-Navarrete, L. de-Marcos, L. Fernández-Sanz, C. Pagés, and J.-J. Martínez-Herráiz, "Gamifying learning experiences: Practical implications and outcomes", Computers & Education, 63, 2013, pp. 380-392. (ScienceDirect)
5. T. Dong, M. Dontcheva, D. Joseph, K. Karahalios, M.W. Newman, and M.S. Ackerman, "Discovery-based games for learning software", In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, May 5-10, 2012, Austin, Texas, USA, ACM, pp. 2083-2086. (ACM)
6. T. Downes-Le Guin, R. Baker, J. Mechling, and E. Ruylea, "Myths and realities of respondent engagement in online surveys", International Journal of Market Research, 54(5), 2012, 1-21. (ScienceDirect)
7. C. Eickhoff, C.G. Harris, A.P. de Vries, and P. Srinivasan, "Quality through flow and immersion: gamifying crowdsourced relevance assessments", In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, August 12-16, 2012, Portland, Oregon, USA, ACM, pp. 871-880. (ACM)
8. R. Farzan, and P. Brusilovsky, "Encouraging user participation in a course recommender system: An impact on user behavior", Computers in Human Behavior, 27(1), 2011, pp. 276-284. (Scopus)

9. R. Farzan, J.M. DiMicco, D.R. Millen, B. Brownholtz, W. Geyer, and C. Dugan, "When the experiment is over: Deploying an incentive system to all the users", In Symposium on Persuasive Technology, 2008. (Through references)
10. R. Farzan, J.M. DiMicco, D.R. Millen, B. Brownholtz, W. Geyer, and C. Dugan, "Results from deploying a participation incentive mechanism within the enterprise", In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, April 5-10, 2008, Florence, Italy, ACM, pp. 563-572. (ACM)
11. Z. Fitz-Walter, D. Tjondronegoro, and P. Wyeth, "Orientation passport: using gamification to engage university students", In Proceedings of the 23rd Australian Computer-Human Interaction Conference, November 28 - December 2, 2011, Canberra, Australia, ACM, pp. 122-125. (ACM)
12. D.R. Flatla, C. Gutwin, L.E. Nacke, S. Bateman, and R.L. Mandryk, "Calibration games: making calibration tasks enjoyable by adding motivating game elements", In Proceedings of the 24th annual ACM symposium on User interface software and technology, October 16-19, 2011, Santa Barbara, CA, USA, ACM, pp. 403-412. (ACM)
13. S. Grant, and B. Betts, "Encouraging User Behaviour with Achievements: An Empirical Study", In Proceedings of the 10th Working Conference on Mining Software Repositories, May 18-19, 2013, San Francisco, CA, USA, pp. 65-68. (ACM)
14. A. Gustafsson, and M. Bång, "Evaluation of a pervasive game for domestic energy engagement among teenagers", In Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, December, 2008, Yokohama, Japan, ACM, pp. 232-239. (ACM)
15. L. Hakulinen, T. Auvinen, and A. Korhonen, "Empirical Study on the Effect of Achievement Badges in TRAKLA2 Online Learning Environment", In Proceedings of Learning and Teaching in Computing and Engineering (LaTiCE) conference, March 21-24, 2013, Macau, pp. 47-54. (Scopus)
16. S. Halan, B. Rossen, J. Cendan, and B. Lok, "High Score!-Motivation Strategies for User Participation in Virtual Human Development", In Proceedings of the international conference on Intelligent Virtual Agents, 2010, Springer Berlin/Heidelberg, pp. 482-488. (ACM)
17. J. Hamari, "Transforming Homo Economicus into Homo Ludens: A Field Experiment on Gamification in a Utilitarian Peer-To-Peer Trading Service", Electronic Commerce Research and Applications, 12(4), 2013, pp. 236-245. (Scopus)
18. J. Hamari, and J. Koivisto, "Social motivations to use gamification: an empirical study of gamifying exercise", In Proceedings of the European Conference on Information Systems, June 5-8, 2013, Utrecht, The Netherlands. (AISel)
19. J.H. Jung, C. Schneider, and J. Valacich, "Enhancing the motivational affordance of information systems: The effects of real-time performance feedback and goal setting in group collaboration environments", Management Science, 56(4), 2010, pp. 724-742.

(Scopus)

20. W. Li, T. Grossman, and G. Fitzmaurice, "GamiCAD: a gamified tutorial system for first time autocad users", In Proceedings of the 25th annual ACM symposium on User interface software and technology, October 7-10, 2012, Cambridge, Massachusetts, USA, ACM, pp. 103-112. (ACM)
21. M. Montola, T. Nummenmaa, A. Lucero, M. Boberg, and H. Korhonen, "Applying game achievement systems to enhance user experience in a photo sharing service", In Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era, September 30 - October 2, 2009, Tampere, Finland, ACM, pp. 94-97. (ACM)
22. A.-L. Smith, and L. Baker, "Getting a clue: creating student detectives and dragon slayers in your library", Reference Services Review, 39(4), 2011, pp. 628-642. (Scopus)
23. J. Thom, D. Millen, and J. DiMicco, "Removing gamification from an enterprise SNS", In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, February 11-15, 2012, Seattle, Washington, USA, ACM, pp. 1067-1070. (ACM)
24. M. Witt, C. Scheiner, and S. Robra-Bissantz, "Gamification of Online Idea Competitions: Insights from an Explorative Case", INFORMATIK 2011 - Informatik schafft Communities 41. Jahrestagung der Gesellschaft für Informatik, October 4-7, 2011, Berlin, Germany. (Google Scholar)

A.0.2 Tabelas com as classificações

Tabela 2 – Classificação dos trabalhos por Hamari, Koivisto e Sarsa (2014).

	Atividade ou serviço	Categoria motivacional	Elementos de jogos	Repostas psicológicas	Respostas comportamentais	Resultado	Tipo de estudo	Método de avaliação
1	Website de perguntas e respostas	Cognitivo	Badges (Medalhas)		Participação		Quantitativo	Análise estatística
2	Quiz	Cognitivo, social, emocional	Pontuação, feedback, ranqueamento, desafios	Diversão, engajamento	Aprendizado		Quantitativo	Questionário
3	Ferramenta de ensino online	Cognitivo	Medalhas	Diversão, conquista	Participação	Positivo parcial	Quantitativo	Questionário
4	Plataforma de aprendizado	Cognitivo, social, emocional	Medalhas, ranqueamento	Engajamento	Aprendizado	Positivo parcial	Quantitativo	Experimento
5	Adobe Photoshop	Cognitivo	Objetivos, desafios, feedback, níveis, enredo	Diversão, engajamento	Aprendizado		Qualitativo	Entrevistas
6	Survey	Cognitivo, emocional	Narrativa, avatars, recompensas	Satisfação, engajamento	Participação	Positivo parcial	Quantitativo	Questionário
7	Crowdsourced document relevance assessments	Cognitivo, social	Pontuação, ranqueamento e desafios	Engajamento	Qualidade e eficiência	Positivo	Quantitativo	Análises
8	Sistema de recomendação de curso baseado na comunidade	Cognitivo	Progresso		Aumento de produção	Positivo parcial	Quantitativo	Experimento
9	Sistema interno de rede social de uma companhia	Cognitivo, social	Pontuação, níveis, ranqueamento		Participação	Positivo parcial	Mixagem de métodos	Análises
10	Sistema interno de rede social de uma companhia	Cognitivo, social	Pontuação, níveis, ranqueamento		Participação	Positivo parcial	Mixagem de métodos	Entrevistas
11	Aplicação móvel informativa para alunos de universidades	Cognitivo, emocional	Conquistas, objetivos	Diversão	Exploração		Quantitativo	Questionário
12	Tarefas de calibração	Cognitivo	Desafios, progresso, enredo, níveis, recompensas	Diversão	Qualidade	Positivo parcial	Quantitativo	Questionário
13	Website de questionário	Cognitivo	Medalhas		Engajamento		Quantitativo	Análise estatística

Fonte: Adaptação do autor de (Hamari, Koivisto e Sarsa , 2014 , p. 8, 9, 10).

Tabela 3 – Continuação da classificação dos trabalhos por Hamari, Koivisto e Sarsa (2014).

	Atividade ou serviço	Categoria motivacional	Elementos de jogos	Repostas psicológicas	Respostas comportamentais	Resultado	Tipo de estudo	Método de avaliação
14	Economização de energia	Cognitivo, social	Enredo, desafios, níveis, ranqueamento, feedback	Motivação social	Mudança no consumo		Mixagem de métodos	Análise entrevistas
15	Aprendizado online	Cognitivo	Medalhas		Aprendizado		Quantitativo	Experimento
16	Pacientes virtuais para treinamento de tratamentos	Cognitivo, emocional	Ranqueamento, narrativa, desafios	Mudança na socialização	Participação	Positivo parcial	Mixagem de métodos	Análises
17	Comércio ponto a ponto	Cognitivo, social	Medalhas, objetivos	Socialização, competição	Qualidade	Positivo parcial	Quantitativo	Análise
18	Serviço de rede social	Social, emocional	Medalhas, ranqueamento, níveis	Socialização	Diversão	Positivo parcial	Quantitativo	Questionário
19	Ambiente de geração de ideia mediada por computador	Cognitivo	Objetivos, feedback		Performace	Positivo parcial	Quantitativo	Experimento
20	Utilização do AutoCAD	Cognitivo	Enredo, objetivos, feedback, desafios, recompensas	Engajamento, diversão	Performace	Positivo parcial	Mixagem de métodos	Análise, entrevistas
21	Serviço móvel de compartilhamento de fotos	Cognitivo	Conquistas, pontuação	Motivação			Qualitativo	Entrevistas
22	Serviço de biblioteca	Cognitivo	Enredo, recompensas		Aprendizado	Positivo parcial	Quantitativo	Questionário
23	Sistem interno de rede social de uma companhia	Cognitivo, social, emocional	Nenhuma		Participação	Positivo	Quantitativo	Análise
24	Competição de ideias online	Cognitivo, social	Pontuação, ranqueamento	Motivação, diversão			Quantitativo	Questionário

Fonte: Adaptação do autor de (Hamari, Koivisto e Sarsa , 2014, p. 8, 9, 10).