

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DE MINAS
GERAIS
CAMPUS SÃO JOÃO EVANGELISTA**

HATUS DANIEL ROCHA COSTA FEITOSA; HUDSON PIMENTA DOS SANTOS

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA APLICAÇÃO DA
TÉCNICA DE GAMIFICAÇÃO COMO APOIO A DISCIPLINA DE ALGORITMOS
E ESTRUTURAS DE DADOS I NO IFMG - CAMPUS SÃO JOÃO EVANGELISTA**

**SÃO JOÃO EVANGELISTA
2017**

HATUS DANIEL ROCHA COSTA FEITOSA; HUDSON PIMENTA DOS SANTOS

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA APLICAÇÃO DA
TÉCNICA DE GAMIFICAÇÃO COMO APOIO A DISCIPLINA DE ALGORITMOS
E ESTRUTURAS DE DADOS I NO IFMG - CAMPUS SÃO JOÃO EVANGELISTA**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Educação, Ciências e Tecnologia de Minas Gerais - Campus São João Evangelista como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof.^a Ma. Karina Dutra de Carvalho Lemos

Coorientador: Prof. Esp. Ítalo Magno Pereira

SÃO JOÃO EVANGELISTA
2017

FICHA CATALOGRÁFICA

F297d Feitosa, Hatus Daniel Rocha Costa; Santos, Hudson Pimenta dos.
2017

Desenvolvimento de uma plataforma para aplicação da técnica de gamificação como apoio a disciplina de Algoritmos e Estruturas de Dados I no IFMG - Campus São João Evangelista. / Hatus Daniel Rocha Costa Feitosa; Hudson Pimenta dos Santos. – 2017.
65f. ; il.

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista, 2017.

Orientadora: Ma. Karina Dutra de Carvalho Lemos
Coorientador: Esp. Ítalo Magno Pereira.

1. Gamificação. 2. Dificuldades. 3. Motivação. 4. Algoritmos.
I. Feitosa, Hatus Daniel Rocha Costa. II. Santos, Hudson Pimenta dos.
III. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista. IIII. Título.

CDD 794.81526

Elaborada pela Biblioteca Professor Pedro Valério

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais
Campus São João Evangelista

Bibliotecária Responsável: Rejane Valéria Santos – CRB-6/2907

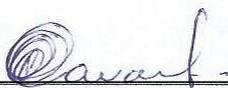
HATUS DANIEL ROCHA COSTA FEITOSA; HUDSON PIMENTA DOS SANTOS

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA APLICAÇÃO DA
TÉCNICA DE GAMIFICAÇÃO COMO APOIO A DISCIPLINA DE ALGORITMOS
E ESTRUTURAS DE DADOS I NO IFMG - CAMPUS SÃO JOÃO EVANGELISTA**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais - Campus São João Evangelista como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em 8 / 12 / 2017

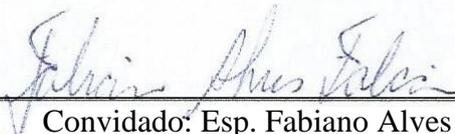
BANCA EXAMINADORA



Orientadora: Prof^a. Ma. Karina Dutra de Carvalho Lemos
Instituto Federal de Minas Gerais - Campus São João Evangelista



Coorientador: Prof. Esp. Ítalo Magno Pereira
Instituto Federal de Minas Gerais - Campus São João Evangelista



Convidado: Esp. Fabiano Alves Falcão
Instituto Federal de Minas Gerais - Campus São João Evangelista

RESUMO

Este trabalho tem como tema a técnica de gamificação e sua aplicação na disciplina de Algoritmos e Estruturas de Dados I (AED I) do curso de Bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais - Campus São João Evangelista (IFMG - SJE). Seu principal objetivo foi o desenvolvimento de uma ferramenta para facilitar a aplicação da gamificação com intuito de estimular e engajar os estudantes no aprendizado de algoritmos. Como metodologia foram aplicados dois questionários, o primeiro visando compreender as dificuldades presentes no aprendizado e o segundo, analisar as possíveis contribuições da gamificação neste contexto. Estes tiveram como amostra três turmas do próprio curso. Como resultados foram constatadas dificuldades no aprendizado de algoritmos, e com a aplicação da técnica pode-se averiguar um aumento na motivação dos estudantes ocasionada pelo desafio e pela competição gerada com o uso da plataforma.

Palavras chaves: Gamificação. Dificuldades. Motivação. Algoritmos.

ABSTRACT

This study has as theme the gamification technique and its application in the disciplines of Algorithms and Data Structure I of the Bachelor of Information Systems course of the Federal Institute of Minas Gerais - São João Evangelista Campus (IFMG – SJE). Its main objective was the development of a tool to ease the application of gamification to stimulate and engage the students in the learning of algorithms. As a methodology, two questionnaires were applied, the first looking to comprehend the difficulties that are present in the learning and the second, to analyze the possible contributions of gamification in this context. These had as sample, three classes of the course itself. As results, difficulties were detected in the learning of algorithms, and with the application of the technique it is possible to verify an increase in the student's motivation caused by the challenge and the competition generated with the use of the platform.

Keywords: Gamification. Difficulties. Motivation. Algorithms.

LISTA DE FIGURAS

Figura 1 - Desenvolvimento incremental	20
Figura 2 - Organização MVC	21
Figura 3 - Diagrama de caso de uso	33
Figura 4 - Diagrama de atividade do administrador	34
Figura 5 - Diagrama de atividade do usuário	34

LISTA DE GRÁFICOS

Gráfico 1 - Complexidade da disciplina	36
Gráfico 2 - Dificuldades no aprendizado.....	37
Gráfico 3 - Grau de entendimento	37
Gráfico 4 - Frequência de busca por ajuda	38
Gráfico 5 - Frequência de busca por materiais didáticos.....	38
Gráfico 6 - Esforço em aprender	39
Gráfico 7 - Aumento do interesse pela disciplina	39
Gráfico 8 - Estímulo por parte do professor	39
Gráfico 9 - Complexidade da disciplina	41
Gráfico 10 - Contribuição da gamificação	41
Gráfico 11 - Gamificação na programação.....	42
Gráfico 12 - Interação com a plataforma.....	42
Gráfico 13 - Nível das missões.....	43
Gráfico 14 - Apresentação do <i>feedback</i>	43
Gráfico 15 - <i>Feedback</i>	44

LISTA DE SIGLAS

AED I - Algoritmos e Estruturas de Dados I

AED III - Algoritmos e Estruturas de Dados III

CRUD - *Create, Read, Update, Delete*

CSS - *Cascading Style Sheets*

DGBL - *Digital Game-Based Learning*

HTML - *HyperText Markup Language*

IFMG - SJE - Instituto Federal de Minas Gerais - Campus São João Evangelista

MVC - *Model-View-Controller*

PHP - *Preprocessed Hypertext Pages*

UFMT - Universidade Federal do Mato Grosso

UFRN - Universidade Federal do Rio Grande do Norte

UML - *Unified Modeling Language*

SUMÁRIO

1	INTRODUÇÃO	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	MOTIVAÇÃO NA APRENDIZAGEM	12
2.2	ALGORITMOS.....	13
2.2.1	Dificuldades na aprendizagem	14
2.3	GAMIFICAÇÃO.....	15
2.3.1	Gamificação na educação	17
2.4	PROCESSOS PARA APLICAÇÕES WEB	18
2.4.1	Desenvolvimento incremental	19
2.4.2	Reúso de software	20
2.4.3	MVC	21
2.4.4	Linguagens para o desenvolvimento	22
2.4.5	Levantamento de requisitos	23
2.4.6	Modelagem da aplicação	24
2.4.7	Aplicações e sistemas para o desenvolvimento	24
2.5	DESIGN DE INTERAÇÃO.....	25
2.6	TRABALHOS CORRELATOS.....	26
3	METODOLOGIA	28
3.1	NATUREZA DA PESQUISA	28
3.2	POPULAÇÃO E AMOSTRA.....	28
3.3	INSTRUMENTOS UTILIZADOS	29
3.4	MÉTODOS E PROCEDIMENTOS.....	30
3.4.1	Requisitos funcionais	31
3.4.2	Requisitos não funcionais	32
3.4.3	Diagramas UML	33
3.5	TRATAMENTO DOS DADOS.....	35
4	RESULTADOS E DISCUSSÕES	36
4.1	RESULTADOS E DISCUSSÕES ACERCA DO PRIMEIRO QUESTIONÁRIO	36
4.2	RESULTADOS E DISCUSSÕES ACERCA DO SEGUNDO QUESTIONÁRIO	40
5	CONCLUSÕES	45
	REFERÊNCIAS	46
	APÊNDICE A - PRIMEIRO QUESTIONÁRIO	53
	APÊNDICE B - SEGUNDO QUESTIONÁRIO	56

1 INTRODUÇÃO

No contexto atual, em que os jogos digitais têm-se tornado uma das formas mais utilizadas de entretenimento, observa-se um constante desenvolvimento deste mercado. Fato evidenciado pela PWC (2016) que aponta o Brasil como 17º maior mercado de jogos do mundo em 2020, com faturamento aproximado de US\$ 0,8 bilhões e crescimento de 11,3% ao ano.

Considerando este crescimento, muito se tem questionado sobre os benefícios dos jogos na sociedade contemporânea. No entanto, este assunto já vem sendo tratado há bastante tempo, como observado no artigo de Almeida (1981) citado por Giaretta *et al.* (1998), no qual são apontados como colaboradores no desenvolvimento físico, intelectual, social e didático. Para Bomfoco e Azevedo (2012), os jogos são poderosas ferramentas para potencializar o aprendizado em diversas áreas do conhecimento.

Observando esses benefícios, em 2002, Nick Pelling criou o termo *gamification* que por sua grande divulgação tem sido aportuguesado para gamificação. Esta expressão se remete a um fenômeno muito discutido atualmente, que de acordo com Fardo (2013), tem-se difundido pela popularização e popularidade dos jogos e suas capacidades de motivar a ação e resolver problemas.

Schlemmer (2014) define gamificação como a aplicação de conceitos e mecânicas de jogos em outros ambientes, com objetivo de solucionar problemas e engajar indivíduos. Neste sentido, Alves e Maciel (2014) completam que a técnica não se trata de um jogo, mas sim da utilização de abstrações e metáforas desta cultura em outros meios.

Nesse contexto, o ambiente educacional tem apresentando grande destaque, como observado por Borges *et al.* (2013) que constatam em sua pesquisa quarenta e oito estudos sobre gamificação na educação, sendo quarenta e seis por cento destes voltados ao ensino superior.

O IFMG - SJE apesar de ser uma instituição conhecida por seus vários anos de especialização em áreas agrárias, desde 2010, oferece em seu *roll* o curso de Bacharelado em Sistemas de Informação. Este apresenta em sua grade algumas disciplinas relacionadas a algoritmos, que normalmente, segundo a pesquisa de Barcelos, Tarouco e Berch (2009), apresentam altos índices de reprovação.

Sendo assim, o presente projeto propôs a criação de uma plataforma de apoio para aplicação da técnica de gamificação na disciplina de AED I no IFMG - SJE. Visto que, a alta

taxa de reprovação ocasionada pela complexidade da disciplina foi o que motivou esta proposta, foi tida como questão norteadora a possível contribuição da técnica para a redução deste índice.

O principal objetivo deste projeto foi o desenvolvimento de uma ferramenta para facilitar a aplicação da gamificação como uma forma de estímulo e engajamento dos estudantes no aprendizado. Diante do exposto, os objetivos específicos foram: a) apontar as principais dificuldades na aprendizagem; b) desenvolver uma plataforma para apoiar o uso da técnica e; c) apresentar os benefícios viabilizados pela gamificação.

Este trabalho foi estruturado em cinco capítulos, sendo este destinado à apresentação do tema, sua justificativa e seus objetivos. No segundo, foi abordada a fundamentação teórica, na qual foram descritos conceitos e visões de especialistas para o embasamento da pesquisa. No capítulo três constou a metodologia, especificando os métodos, procedimentos e ferramentas empregados no trabalho. O capítulo quatro apresentou os resultados obtidos com a pesquisa, assim como a discussão acerca da mesma. E no capítulo cinco foram dadas as conclusões acerca da plataforma desenvolvida e das contribuições da técnica de gamificação.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará os tópicos que nortearão o desenvolver do trabalho, baseando-se em autores que abordam em seus estudos os conceitos sobre motivação, aprendizagem de algoritmos, gamificação e sua aplicação no ambiente educacional. Além de tratar das tecnologias a serem utilizadas neste projeto.

2.1 MOTIVAÇÃO NA APRENDIZAGEM

Para Castoldi e Polinarski (2009), a motivação deve estar sempre presente no meio educacional e cabe ao docente facilitar a construção e desenvolvimento desta noção no aprendiz. Segundo Lourenço e Paiva (2010), o interesse pelos aspectos motivacionais na aprendizagem é relativamente recente, já que as teorias mais antigas retratavam a mesma apenas como uma pré-condição.

Zenorini e Santos (2010) salientam que a motivação no âmbito acadêmico tem sido estudada sobre diversos ângulos, o que resulta em muitas teorias e abordagens acerca deste tema. Entretanto, conforme citado por Leal, Miranda e Carmo (2013), a Teoria da Autodeterminação tem sido muito retratada nos estudos contemporâneos, tendo seus principais conceitos levantados em análises sobre a aprendizagem.

Esta teoria foi elaborada em 1981 por Richard M. Ryan e Edward L. Deci, e de acordo com Guimarães e Boruchovitch (2004), “foi proposta com o objetivo de compreender os componentes da motivação intrínseca e extrínseca e os fatores relacionados com a sua promoção”.

Para Fadel *et al.* (2014), as motivações intrínsecas são originadas no próprio indivíduo, o que possibilita que o mesmo realize uma tarefa por vontade própria, desde que esta desperte interesse, envolvimento e prazer. Por outro lado, as motivações extrínsecas são oriundas do ambiente externo e tem como razão alguma recompensa requerida pelo sujeito (VIANNA *et al.*, 2013 *apud* FADEL *et al.*, 2014).

Do ponto de vista de Leal, Miranda e Carmo (2013), além de compreender estas duas vertentes, é interessante observar o impacto cultural, uma vez que pesquisas apontam este fenômeno como uma grande influência nos aspectos motivacionais. Neste sentido, Lourenço e

Paiva (2010) enfatizam que na maioria dos casos, a aprendizagem ocorre no meio social e temporal, tendo a sua formação alterada pelos fatores intrínsecos e extrínsecos.

Para Ribeiro (2001), a palavra motivação vem sendo muito utilizada recentemente por educadores para justificar os fracassos dos estudantes no processo de aprendizagem. Em concordância, Rodrigues *et al.* (2014) observam a falta desta como fator crítico na aprendizagem de disciplinas que envolvem algoritmos.

2.2 ALGORITMOS

De acordo com Medina e Ferdig (2006), o termo algoritmo tem sua origem datada do século XVII, oriundo do nome do matemático iraniano Abu Abdullah Mohammad Ibn Musa al-Khawarizmi. Esta expressão, no entanto, possui inúmeras definições.

Na visão de Ziviani (2011, p. 1), algoritmo é “uma sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema”. Já Forbellone e Eberspächer (2005) definem como uma série de passos que visam atingir um objetivo bem definido. Outro ponto de vista é mostrado por Fiscina e Borges (2013) que caracterizam o termo como instruções e especificações passadas para máquina, que são executadas em um menor tempo possível e podem ser repetidas diversas vezes.

Independentemente destas definições, é indiscutível o fato de que as disciplinas que envolvem algoritmos são consideradas a base do ensino nos cursos que abrangem a programação. O próprio curso de Bacharelado em Sistemas de Informação, presente no IFMG - SJE, contém várias disciplinas de algoritmos, como Introdução a Programação, AEDS I, II e III, e Linguagem de Programação I e II.

Na visão de Rapkiewicz *et al.* (2006), tais disciplinas apresentam um conteúdo complexo e difícil de trabalhar, o que por sua vez, influencia e é influenciado pelas dificuldades de aprendizagem dos estudantes.

2.2.1 Dificuldades na aprendizagem

Barcelos, Tarouco e Berch (2009) consideram fundamental o estudo de algoritmos nos cursos de Computação, todavia a sua aprendizagem constitui um dos maiores empecilhos enfrentados pelos alunos. Este problema por sua vez, é refletido no desempenho educacional, já que segundo uma pesquisa feita por estes autores, as disciplinas que contêm algoritmos são as que apresentam maiores taxas de reprovação. Em concordância, Rapkiewicz *et al.* (2006) afirmam que os altos números de desistência também são ocasionados por esta questão. Tais índices ainda se mantêm elevados quando o contexto observado é o próprio IFMG - SJE, visto que os estudantes apresentam dificuldades neste conteúdo.

Normalmente, as disciplinas de algoritmos abordam “os princípios da lógica de programação, com o objetivo de desenvolver a capacidade de análise e resolução de problemas dos alunos através da descrição dos mesmos na forma de algoritmos” (RAPKIEWICZ *et al.*, 2006). Uma vez que a aprendizagem deste conteúdo depende da análise e do raciocínio, Rodrigue *et al.* (2014) salientam que o baixo rendimento se deve, em partes, pela resistência no desenvolvimento do raciocínio lógico, já que os alunos geralmente estão acostumados a decorar o conteúdo.

Esta adversidade por sua vez, pode ser resolvida através da dedicação e esforço do estudante, visto que uma das formas mais eficazes de se aprender algoritmos é através da resolução de muitos exercícios (FISCINA; BORGES, 2013). Mendes (2002) citado Rapkiewicz *et al.* (2006) completa que “a necessidade da prática em técnicas de resolução de problemas e de um bom nível de conhecimentos são fatores que dificultam a aprendizagem por parte dos alunos”.

Falckembach e Araujo (2006) enfatizam que esta dificuldade também decorre da forma de ensino aplicada dentro da sala de aula, já que a metodologia é a mesma para todos. Para estes autores é muito difícil para um professor levar em consideração as necessidades e o nível de conhecimento de cada um, de modo que possa proporcionar individualmente um ensino de qualidade. Piva, Freitas e Paula (2009) completam que para melhorar este cenário é importante observar o perfil dos alunos que estão ingressando no ensino superior, para poder traçar uma estratégia interdisciplinar mais efetiva para o desenvolvimento e retenção destes estudantes.

Do ponto de vista de Borges (2000) citado por Rapkiewicz *et al.* (2006), o modo tradicional de ensino não consegue atrair de forma eficiente o interesse dos alunos, uma vez

que estes não reconhecem a importância de alguns conteúdos em sua formação acadêmica. Devido a isso, os estudantes se veem desmotivados, o que de certa forma, acarreta o baixo rendimento. Analisando estas questões, Rapkiewicz *et al.* (2006) inferem que sob uma ótica pedagógica que trata a aprendizagem como um processo de construção de conhecimento, é possível identificar e intervir, através de estratégias adequadas, nos problemas deste modo de ensino.

Neste contexto, se torna relevante que os professores busquem outras formas ou metodologias que favoreçam o aspecto motivacional. Apesar de que motivar os estudantes ainda seja um desafio, os mecanismos de gamificação surgem como uma possível solução para este problema (AGUIAR, 2015).

2.3 GAMIFICAÇÃO

Segundo Fardo (2013), o termo *gamification* criado por Nick Pelling em 2002, é um fenômeno emergente e que tem-se difundido pela popularização e popularidade dos jogos. Devido a essa difusão, a expressão tem sido aportuguesada para gamificação. No entanto, de acordo com Mastrocola (2012), esta conotação é errônea, já que a tradução que expressaria melhor o sentido da palavra seria ludificação.

Em contraposição, Fardo (2013) afirma que como o vocábulo se remete mais especificamente aos *games*, seria mais adequado o termo gamificação, visto que a palavra ludificação abrangeria um contexto muito além dos jogos. Entretanto, a expressão gamificação se trata de um entrave da língua. Então, para manter seu sentido original, esta precisaria preservar a pronúncia em inglês, em outras palavras, ser lida como “gueimificação” (FARDO, 2013).

Apesar desta divergência acerca da semântica da palavra, a gamificação tem como objetivo analisar os elementos presentes no design dos jogos, que os tornam divertidos, e adaptá-los a situações fora deste ambiente (SCHLEMMER, 2014).

A gamificação pressupõe a utilização de elementos tradicionalmente encontrados nos *games*, como narrativa, sistema de *feedback*, sistema de recompensas, conflito, cooperação, competição, objetivos e regras claras, níveis, tentativa e erro, diversão, interação, interatividade, entre outros, em outras atividades que não são diretamente associadas aos *games*, com a finalidade de tentar obter o mesmo grau de envolvimento e motivação que normalmente encontramos nos jogadores quando em interação com bons *games* (FARDO, 2013).

Fadel *et al.* (2014) mencionam quatro características importantes do contexto dos *games* e conseqüentemente da técnica de gamificação, que possibilitam a imersão e engajamento das pessoas. São estas:

- **Desafio:** tratado como elemento propulsor da motivação e engajamento, devido a sua finalidade de estabelecer objetivos a serem alcançados, incentivando o uso de estratégias que incitam as funções cognitivas e subjetivas.
- **Feedback:** componente fundamental para subsidiar o desenvolvimento do usuário, informando o caminho a ser traçado para alcançar seus objetivos.
- **Premiação:** elemento relevante na motivação, por se tratar de uma recompensa dada pelo cumprimento de uma determinada tarefa. Além disso, pode ser considerada uma forma de *feedback*, já que demonstra o resultado alcançado pelo trabalho do indivíduo.
- **Práticas colaborativas:** elemento significativo na consolidação de uma inteligência coletiva, que está presente no âmbito dos jogos e no social, pelo fato de que a todo o momento conhecimentos são gerados pela troca de experiências.

Ainda de acordo com Fadel *et al.* (2014, p. 15), a “gamificação tem como base a ação de se pensar como em um jogo, utilizando as sistemáticas e mecânicas do ato de jogar em outro contexto”. Esta técnica se trata da “construção de modelos, sistemas ou modo de produção com foco nos indivíduos, tendo como premissa a lógica dos *games*” (FADEL *et al.*, 2014, p. 77). Tudo isso, levando-se em consideração a motivação, o sentimento e a participação das pessoas envolvidas (CHOU, 2014 *apud* FADEL *et al.*, 2014).

Para Zichermann e Cunningham (2011), os mecanismos de gamificação possibilitam o alinhamento dos interesses dos criadores de um determinado artefato com os objetos que motivam os usuários, uma vez que ambientes que interagem com emoções e desejos são eficazes no engajamento.

Desta forma, observando os elementos que constituem o contexto motivacional, Zichermann e Cunningham (2011) identificam a motivação intrínseca e extrínseca, já abordadas anteriormente. Neste contexto, Fadel *et al.* (2014) ressaltam a importância de estimular ambas as formas de motivação nos ambientes gamificados, para que desta maneira se consiga um maior envolvimento por parte do usuário.

Considerando o ponto de vista de Alves e Maciel (2014), com o desenvolvimento das tecnologias na sociedade contemporânea, tem havido uma crescente criação de novos recursos e estratégias de engajamento, no intuito de dar suporte aos modelos educacionais. Nesse cenário, estes autores destacam o crescimento do uso da gamificação.

2.3.1 Gamificação na educação

A gamificação traz a possibilidade de conectar a escola ao universo dos jovens, ao mesmo tempo em que se objetiva a aprendizagem. Mas ao invés de se apoiar nos métodos tradicionais como notas, utilizam-se elementos da mecânica dos jogos para promover experiências que envolvam emocionalmente e cognitivamente os estudantes (FADEL *et al.*, 2014).

Nesse aspecto, a gamificação:

[...] tem sido apropriada pela área da educação, possibilitando a construção de situações de ensino e de aprendizagem capazes de engajar os sujeitos, de forma prazerosa, na definição e resolução de problemas, contribuindo para repensar o contexto educacional formal (SCHLEMMER, 2014).

Apesar de se tratar de uma técnica recente, Fadel *et al.* (2014) mencionam o uso de atividades gamificadas na educação tradicional, como por exemplo, o ato de recompensar um aluno com uma estrelinha ou ir elevando a dificuldade de um exercício para desafia-los. Ainda que se trate de exemplos corriqueiros, é claramente observado o uso da técnica. Desta forma, a única mudança observada é com relação à sua compreensão e relevância no ambiente educacional (FADEL *et al.*, 2014).

Fardo (2013) aponta a educação como um ótimo campo para aplicação da gamificação, já que neste, se encontram indivíduos cheios de conhecimentos oriundos das interações com *games* e inseridos num ambiente que, na maioria das vezes, utiliza-se de métodos passivos de ensino-aprendizagem que têm se mostrado desmotivantes para este público.

Quadros (2012) salienta uma questão muito relevante, que é a diferença existente entre o ensino mediado por jogos e a gamificação. Alves e Maciel (2014) estabelecem que a aprendizagem baseada em jogos, do inglês *digital game-based learning* (DBGL), não deve ser confundida em nenhum aspecto com a gamificação, uma vez que esta não se trata da utilização de jogos na aprendizagem, mas apenas dos elementos e metáforas oriundos desta cultura.

Neste cenário, surge outro termo muito usado atualmente, que são os jogos sérios, do inglês *serious games*. Para Hagglund (2012) se tratam de jogos desenvolvidos com objetivos específicos, diferentes do entretenimento propriamente dito. Na maioria das vezes, são utilizados no meio educacional e para o treinamento de pessoas.

O ensino mediado por jogos parte da premissa de que o usuário tem controle total sobre um determinado objeto ou ser digital, e através deste controle, pode realizar tarefas e explorar diversos ambientes. Como exemplo, tem-se o jogo Warcraft, no qual o usuário pode escolher seu personagem, com o qual realizará missões, enquanto interage com pessoas do mundo inteiro e com isso desenvolve habilidades relacionadas à organização de grupos sociais, aprendizagem de línguas, dentre outras (QUADROS, 2012).

Em contraposição, a gamificação apenas compartilha dos elementos e do design dos jogos como desafios, uso de estratégia e recompensas, a fim de motivar a aprendizagem nos indivíduos. Como exemplo, tem-se o Livemocha, que se trata de um suporte de rede sociais voltado para o ensino e aprendizado de idiomas. Neste ambiente, os usuários realizam lições e ganham pontos, que elevam seu grau de representatividade e desbloqueiam atividades complementares na língua que estão aprendendo (QUADROS, 2012).

2.4 PROCESSOS PARA APLICAÇÕES WEB

Para Jacyntho (2008), o desenvolvimento de aplicações *web* apresenta grandes diferenças com relação ao desenvolvimento de softwares convencionais, uma vez que as duas formas possuem características técnicas e organizacionais distintas.

No entanto, Mendes *et al.* (2004) citado por Conte, Mendes e Travassos (2005) ressaltam a ausência de métodos padronizados para o desenvolvimento *web*, sendo utilizados na prática, alguns aspectos da Engenharia de Software tradicional. Neste contexto, Sommerville (2011) salienta que, apesar da Engenharia de Software adotar uma abordagem sistemática para o trabalho, no caso do desenvolvimento de sistemas *web*, é mais adequado uma abordagem criativa e menos formal, já que neste processo é requerida a combinação de habilidades de software e projeto.

Entretanto, na visão deste mesmo autor, existem quatro atividades fundamentais comuns a todos os processos de software. São estas:

- Especificação: nesta fase são definidas questões com relação à produção do sistema.
- Desenvolvimento: etapa na qual o software é projetado e programado.
- Validação: estágio no qual o sistema é verificado, com objetivo de garantir a qualidade.

- Evolução: momento em que o software passa por modificações para se adequar melhor ao mercado e ao cliente.

Pressman e Maxim (2016) salientam que, para reduzir as dificuldades presentes no desenvolvimento de softwares, são utilizados os modelos de processo. Estes têm como finalidade a estruturação e ordenação do processo de desenvolvimento. Atualmente, existem diversos modelos, como exemplo, pode-se citar o modelo em cascata e o incremental.

Dentro do contexto *web*, Sommerville (2011) considera ser mais adequado o uso de abordagens baseadas no desenvolvimento incremental e no reúso de softwares, já que é muito difícil definir antecipadamente todos os requisitos de um sistema *web* e construí-lo sem utilizar nenhum componente preexistente.

Na visão de Pressman e Maxim (2016), outro ponto importante a se destacar no ambiente *web* é a arquitetura da aplicação, visto que esta estabelece a infraestrutura necessária para o sistema atingir seus objetivos. Segundo Sommerville (2011), o conhecido padrão de arquitetura Modelo-Visão-Controlador, do inglês Model-View-Controller (MVC), é a base para o gerenciamento de interação de diversas aplicações *web*.

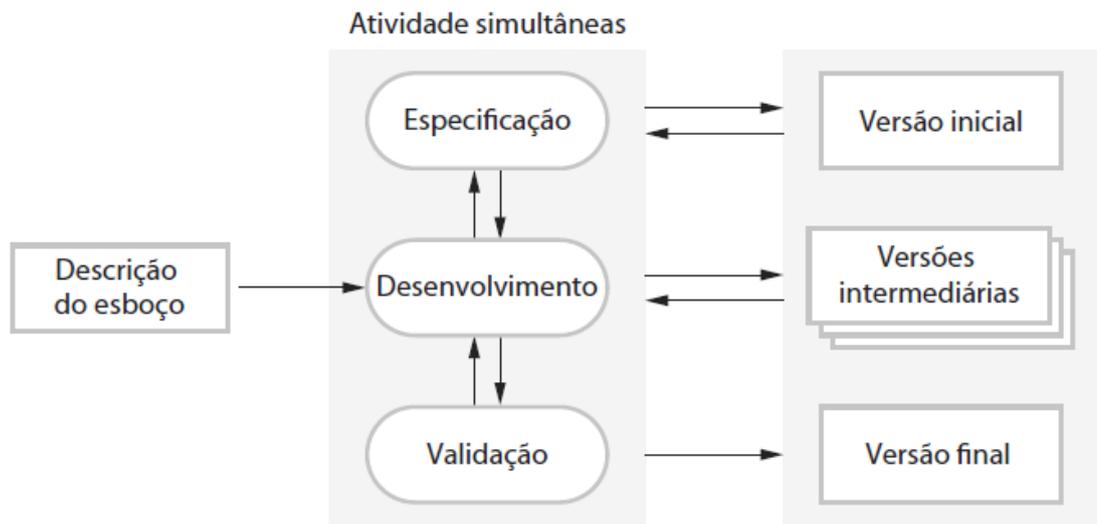
2.4.1 Desenvolvimento incremental

O desenvolvimento incremental é um modelo de processo que “[...] intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), de maneira que cada versão adiciona funcionalidade à anterior” (SOMMERVILLE, 2011).

Nas aplicações desenvolvidas de forma incremental, as primeiras versões contemplam apenas o produto essencial, dado que, possuem somente requisitos básicos. Os demais requisitos serão aplicados gradativamente de acordo com os resultados alcançados pela execução e avaliação das primeiras versões (PRESSMAN; MAXIM, 2016).

Atualmente, a abordagem incremental é a mais comum para o desenvolvimento de sistemas, uma vez que reflete a forma como normalmente os indivíduos resolvem seus problemas. Esta possui vários benefícios, como o menor custo para acomodar mudanças nos requisitos, a possibilidade de um maior *feedback* por parte do cliente e a viabilidade de uma rápida implementação e entrega do sistema (SOMMERVILLE, 2011). A estrutura deste modelo pode ser observada na Figura 1.

Figura 1 - Desenvolvimento incremental



Fonte: Sommerville, 2011.

2.4.2 Reúso de software

A abordagem baseada no reúso de softwares tem como finalidade a integração de componentes já existente ao sistema, ao invés de desenvolvê-lo a partir do zero. O reúso de softwares ganhou maior notoriedade a partir dos anos 2000, quando o mercado começou a exigir menores custos de produção, manutenção, entregas mais rápidas e softwares com maior qualidade (SOMMERVILLE, 2011).

O reúso de softwares trouxe inúmeras vantagens, dentre as quais, destaca-se a redução de custos para o desenvolvimento, o aumento na confiabilidade dos produtos, a diminuição de riscos no processo e o desenvolvimento mais ágil (LANNA, 2009).

Para Sommerville (2011), o reúso de softwares pode ser dividido em três unidades, levando em consideração o seu tamanho. São estas:

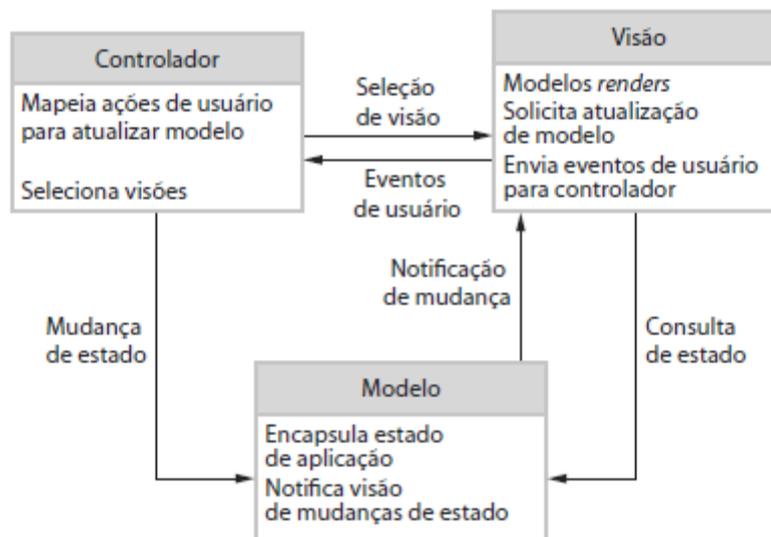
- Reúso de sistema de aplicação: quando sistemas de aplicação são utilizados como um todo por outros ou configurados de forma a se adaptar a um novo contexto.
- Reúso de componentes: ocorre normalmente com subsistemas ou objetos originários de um software que são reusados em novos sistemas.
- Reúso de objetos e funções: ocorre com componentes de softwares que implementam uma única função ou classe de objetos num software e são reutilizados em terceiros.

Ainda de acordo com Sommerville (2011), existem situações em que o alto custo para se modificar um componente específico, torna seu reuso inviável. Nestes casos, utiliza-se o reuso de conceito, que se trata da reutilização de uma ideia ou técnica.

2.4.3 MVC

De acordo com Sommerville (2011), a arquitetura MVC é estruturada em três componentes lógicos, conforme é observado na Figura 2.

Figura 2 - Organização MVC



Fonte: Sommerville, 2011.

Em conformidade, Pressman e Maxim (2016) descrevem os três componentes lógicos observados como:

- **Modelo:** contém todo o conteúdo e a lógica de processamento específicos da aplicação, inclusive os objetos de conteúdo, acesso a fontes de dados, informações externas e toda a funcionalidade de processamento específica para a aplicação.
- **Visão:** contempla todas as funções específicas da interface, além de possibilitar a apresentação dos conteúdos e da lógica de processamento, inclusive os objetos de conteúdo, acesso a fontes de dados, informações externas e toda a funcionalidade de processamento exigida pelo usuário.

- Controlador: responsável por gerenciar o acesso ao modelo e à visão, além de coordenar o fluxo de dados entre estes.

O MVC apresenta algumas vantagens, como a possibilidade de alterar os dados de forma independente da representação, a apresentação de um mesmo dado de formas diferentes e a capacidade de generalizar para todas as representações uma mudança feita em apenas uma delas (SOMMERVILLE, 2011).

2.4.4 Linguagens para o desenvolvimento

Atualmente, para o desenvolvimento de aplicações *web* são utilizados vários tipos de linguagens, dentre estas, as linguagens de marcação e programação. A Linguagem de Marcação de Hipertextos, do inglês *HyperText Markup Language* (HTML), é uma linguagem usada para a criação de páginas *web*, considerada a mais popular do seu contexto (PORTAL EDUCAÇÃO, 2013). De acordo com o W3SCHOOLS (2017a), descreve a estrutura das páginas usando elementos denominados *tags*.

Geralmente, para melhorar o design das páginas HTML são utilizadas as Folhas de Estilo em Cascata, do inglês *Cascading Style Sheets* (CSS), que são mecanismos para adicionar estilos às páginas. O CSS descreve como os elementos HTML serão exibidos na tela, de forma a economizar trabalho do programador (W3SCHOOLS, 2017b).

No entanto, somente o HTML e CSS não são capazes de possibilitar a dinamicidade requerida nas plataformas atuais, então surge o *JavaScript*. Esta linguagem de programação é definida pelo W3SCHOOLS (2017c) como uma linguagem para HTML e *web* de fácil aprendizagem para o usuário.

O *JavaScript* fornece às páginas web a possibilidade de programação, transformação e processamento de dados enviados e recebidos, interagindo com a marcação e exibição de conteúdo da linguagem HTML e com a estilização desse conteúdo proporcionada pelo CSS nessas páginas (SILVA, 2015).

Normalmente, para tornar o *JavaScript* mais fácil e prático de usar, utiliza-se a biblioteca *jQuery*, que simplifica muitas tarefas desta linguagem (W3SCHOOLS, 2017e). A jQuery Foundation (2017) afirma que esta é uma biblioteca rápida, pequena e rica em recursos, além do fato de combinar versatilidade e extensibilidade.

Num ambiente *web*, existem dois tipos de linguagem, as *server-side* (lado do servidor) e as *client-side* (lado do cliente). As linguagens *server-side* tem suas operações processadas diretamente no servidor, enquanto as *client-side* podem ser processadas localmente (UOL HOST, 2016). Diferente das linguagens citadas anteriormente, que são *client-side*, as linguagens mencionadas a seguir são *server-side*.

As Páginas de Hipertexto Pré-processadas, do inglês *Preprocessed Hypertext Pages* (PHP), são consideradas uma linguagem de *script* de servidor e uma poderosa ferramenta para tornar as páginas *web* em dinâmicas e interativas (W3SCHOOLS, 2017d). O PHP tem sua aparência sintática e natureza dinâmica similares ao *JavaScript*, porém permite um fácil acesso aos dados de formulários HTML (SEBESTA, 2011). De acordo com o índice de linguagens mais usadas apresentado pelo TIOBE (2017), estas duas linguagens ocupam o nono e sétimo lugares, respectivamente, no *ranking* do mês de maio.

Além do PHP, existem várias outras linguagens de programação. Como exemplo, pode-se citar o Java, que foi a linguagem de programação mais utilizada no mundo no mês de maio (TIOBE, 2017). Esta linguagem foi projetada com base na linguagem C++, porém apresenta uma maior simplicidade e confiabilidade (SEBESTA, 2011).

2.4.5 Levantamento de requisitos

Geralmente, no desenvolvimento de aplicações tem-se como uma das principais etapas o levantamento de requisitos. Na visão de Pressman e Maxim (2016), nesta fase são identificados os problemas, são propostas soluções, que são analisadas e são especificadas em um conjunto de requisitos. Os requisitos correspondem às descrições de como o sistema deve funcionar, suas funções e restrições.

De acordo com Sommerville (2011), os requisitos de software são frequentemente divididos em requisitos funcionais e não funcionais. Os funcionais são descrições dos serviços que o sistema deve conter e seu comportamento em determinadas situações. Já os não funcionais são as restrições das funções e serviços oferecidos pelo sistema (SOMMERVILLE, 2011).

2.4.6 Modelagem da aplicação

Normalmente, para a modelagem das aplicações são criados diagramas utilizando a Linguagem de Modelagem Unificada, do inglês *Unified Modeling Language* (UML), definida por Larman (2007), como a linguagem padrão para diagramação. Com a UML é possível criar vários tipos de diagramas que auxiliam no processo de desenvolvimento. Dentre estes, pode-se citar, o diagrama de atividade e o de caso de uso.

Para Sommerville (2011), os diagramas de caso de uso representam todas as possíveis interações do sistema. Segundo Pressman e Maxim (2016), estes diagramas são importantes para compreender os clientes, uma vez que, quando se trata de obter sucesso, buscar a satisfação do cliente está em primeiro lugar.

Os diagramas de atividade são utilizados para especificar as atividades que constituem um sistema e o fluxo de controle de uma atividade para a outra (SOMMERVILLE, 2011). Na visão de Larman (2007), estes diagramas “são úteis para modelagem de processos de negócios, fluxos de trabalho, fluxos de dados e algoritmos complexos”.

2.4.7 Aplicações e sistemas para o desenvolvimento

No desenvolvimento de sistemas, é necessário o uso de algumas linguagens, como já citado anteriormente. Quando se trata do desenvolvimento *web*, mais especificamente na linguagem PHP, tem-se muitos *frameworks*, dentre os quais, pode-se citar o Laravel e o CodeIgniter. Estes foram apontados pelo DuneBook (2017) como os melhores *frameworks* PHP para se aprender em 2017. O Laravel se destaca por ser o mais recente, oferecer um robusto conjunto de ferramentas e uma arquitetura que incorpora recursos de outros *frameworks*. No entanto, o CodeIgniter que também é baseado no MVC, como o próprio Laravel, apresenta um kit de ferramentas mais simples e elegante que permite a criação de aplicações *web* completas (DUNEBOOK, 2017).

Normalmente, os desenvolvedores também utilizam softwares para otimizar o desenvolvimento, os chamados editores de código, dentre os quais destaca-se o Atom. Este é um editor multi-plataforma e *open source*, que oferece uma interface organizada e simples para o usuário (LAB KING HOST, 2017).

Com relação à modelagem das aplicações, geralmente utiliza-se a UML, como já mencionado anteriormente. No entanto, existem algumas ferramentas que auxiliam neste processo, como exemplo, tem-se o StarUML. Este é um modelador de softwares UML que suporta totalmente onze tipos de diagramas diferentes, além de permitir o uso das mais famosas linguagens de programação (STARUML, 2016).

Atualmente, com a popularidade dos *tablets* e *smarphones*, tem surgido uma nova perspectiva de desenvolvimento *web* que traz consigo novos desafios. Então, para resolver estas novas questões surge o *Web Design Responsivo*, que visa a criação de páginas *web* adaptáveis em qualquer dispositivo, seja este grande ou pequeno (ZEMEL, 2015). Para facilitar a aplicação deste conceito, utiliza-se muito o Bootstrap, que é uma estrutura popular de HTML, CSS e JavaScript para o desenvolvimento de sites responsivos. Trata-se de um *framework* gratuito baseado nas linguagens padrões de desenvolvimento *web*, que facilita e agiliza este processo (W3SCHOOLS, 2017f).

Outra questão relevante para o desenvolvimento é a criação de um ambiente *web*. Para isso, existem algumas ferramentas que simplificam este processo, como exemplo, tem-se o WampServer. Segundo o Techtudo (2013), se trata um software que instala um ambiente de desenvolvimento *web*, possibilitando a criação de aplicações com Apache2, PHP e banco de dados MySQL. Trata-se de um programa intuitivo de usar, que permite a comunicação sem dificuldades com servidor, além de possuir em seu pacote o PhpMyAdmin (TECHTUDO, 2013).

O PhpMyAdmin, por sua vez, é uma ferramenta livre escrita em PHP, destinada a lidar com a administração do MySQL na *web*. O PhpMyAdmin suporta uma gama de operações em MySQL e MariaDB, além de possuir uma interface *web* bem interativa (PHPMYADMIN, 2017).

2.5 DESIGN DE INTERAÇÃO

Na sociedade contemporânea, em que novas tecnologias surgem a todo o momento, se torna fundamental que se busque o desenvolvimento de novos métodos e abordagens para o design no contexto da interação (PORTUGAL, 2010).

Enquanto a engenharia de software se preocupa com as questões relacionadas à realização do projeto, como planejamento, aspectos estruturais e custo, o design de interação

tem como foco a criação de experiências agradáveis para o usuário final (PREECE; ROGERS; SHARP, 2013).

Neste sentido, o conhecimento pode ser construído para os outros, assim como se constrói informações para terceiros a partir de dados. Considerando tal afirmação, o Design de Interação tem a capacidade de criar experiências no usuário, a fim de transmitir um determinado conhecimento (PORTUGAL, 2010).

Na visão de Preece, Rogers e Sharp (2013, p. 8), o design de interação visa “projetar produtos interativos para apoiar o modo como as pessoas interagem em seus cotidianos, seja em casa ou no trabalho”. Essencialmente destina-se ao desenvolvimento de produtos que sejam fáceis, eficientes e agradáveis da perspectiva do usuário, considerando a criação de experiências que melhorem e ampliem a forma como os indivíduos trabalham, se comunicam e interagem (PREECE; ROGERS; SHARP, 2013).

2.6 TRABALHOS CORRELATOS

Como mencionado por Fardo (2013), a gamificação se apresenta como um fenômeno emergente e com potencial para ser aplicada em diversos ambientes, sendo a educação um ótimo campo para sua aplicação. Nessas condições, foram encontrados alguns trabalhos relacionados à aplicação da gamificação no âmbito educacional.

Um destes é o da Prof^a. Ma. Karen da Silva Figueiredo, apresentado no XI Simpósio Brasileiro de Sistemas de Informação, Goiânia/GO, ano de 2015, cujo nome é “Proposta de Gamificação de Disciplinas em um Curso de Sistemas de Informação”. Este trabalho apresenta uma proposta de gamificação em disciplinas do curso de Sistemas de Informação com objetivo de engajar e motivar os alunos. Apresentando o design inicial do jogo para as disciplinas de Algoritmos III e Análise e Projeto de Sistemas I do curso de Sistemas de Informação da Universidade Federal de Mato Grosso (UFMT).

Tem-se também o projeto do Prof. Dr. André M. C. Campos, juntamente com o Me. Renato Quarentei Gardiman e o Prof. Dr. Charles Andryê Galvão Madeira, apresentado no 23º Workshop sobre Educação em Computação, Recife/PE, ano de 2015, cujo nome é “Uma ferramenta gamificada de apoio à disciplina introdutória de programação”. Este projeto apresenta a experiência destes autores no desenvolvimento e aplicação de uma ferramenta denominada *Kodesh (Koding Shell)*, para auxílio de estudantes nas atividades de

programação. A ferramenta em questão foi desenvolvida objetivando, além da questão motivacional dos alunos, também como forma de apoio metodológico no processo de ensino-aprendizagem. Uma vez que no ano de 2013, se instaurava um novo curso de Bacharelado em Tecnologia da Informação na Universidade Federal do Rio Grande do Norte (UFRN) que teria várias turmas em paralelo.

Ambos os trabalhos apresentados se assemelham com o presente projeto, pois se tratam do uso de ferramentas gamificadas como elemento motivador em disciplinas de algoritmos. No entanto, este trabalho se difere dos demais por ser aplicado em outro contexto, no caso, a disciplina de AED I do curso de Bacharelado em Sistemas de Informação do IFMG - SJE. Outro ponto a se destacar, com relação ao trabalho de Campos, Gardiman e Madeira (2015), é o fato do IFMG - SJE não utilizar ferramentas de correção de códigos automáticas e ter no presente projeto, uma primeira tentativa de desenvolver uma plataforma gamificada voltada ao apoio da disciplina em questão.

No contexto de ensino-aprendizagem, Quadros (2012) aponta o Livemocha e o Busuu como bons exemplos de sistemas gamificados, pois são suportes de redes sociais, voltados ao ensino e aprendizagem de idiomas. Numa perspectiva mais atual, tem-se o Duolingo, que segundo Aguiar (2015), é um ambiente on-line que possui elementos gamificados, e também voltado ao ensino-aprendizagem de línguas.

3 METODOLOGIA

Este capítulo apresentará os aspectos da pesquisa, bem como seu caráter, natureza, população e amostra, além de expor os instrumentos, métodos e procedimentos utilizados no decorrer do estudo.

3.1 NATUREZA DA PESQUISA

A metodologia aplicada no presente projeto possui caráter descritivo, que na visão de Gil (2008) busca descrever os principais aspectos de uma população ou fenômeno. A mesma apresentou o método indutivo, que parte da premissa que “a generalização não deve ser buscada aprioristicamente, mas constatada a partir da observação de casos concretos suficientemente confirmadores dessa realidade” (GIL, 2008, p. 10).

A pesquisa teve uma abordagem qualitativa, que de acordo com Gerhardt e Silveira (2009) busca compreender as características de um grupo ou organização que não podem ser quantificadas. E quanto à natureza, se tratou de uma pesquisa aplicada, “que objetiva gerar conhecimento para aplicação prática, dirigida à solução de um problema específico” (GERHARDT; SILVEIRA, 2009, p. 35).

3.2 POPULAÇÃO E AMOSTRA

A população desta pesquisa foi formada pelos professores das disciplinas relacionadas a algoritmos do curso de Bacharelado em Sistemas de Informação do IFMG - SJE e seus respectivos alunos, enquanto a amostra, contou com 49 estudantes do curso. Para a aplicação do primeiro questionário, apresentado no Apêndice A, foram utilizados os alunos da turma de Algoritmos e Estruturas de Dados III (AED III) e os demais estudantes da turma extra de AED I, que auxiliaram no levantamento das principais dificuldades da disciplina, uma vez que já haviam cursado a mesma. Já o segundo, mostrado no Apêndice B, foi aplicado aos

estudantes da turma regular de AED I, com objetivo de averiguar as possíveis contribuições da gamificação.

3.3 INSTRUMENTOS UTILIZADOS

Inicialmente, os esforços foram direcionados ao estudo do fenômeno de gamificação, seu aspecto motivacional e uso na educação, além da análise das dificuldades na aprendizagem de algoritmos. Esta parte constituiu a fundamentação teórica, que foi construída por meio da revisão de artigos e livros de especialistas da área.

Para o desenvolvimento da plataforma foram utilizadas algumas das linguagens padrões para o desenvolvimento *web*, como o próprio HTML, CSS e JavaScript. Com relação à responsividade foi aplicado o Bootstrap, por ser um *framework* popular, e para a edição do código, o software Atom. A criação dos diagramas ficou a cargo do StarUML, por ser um software gratuito e de fácil compreensão.

Além das linguagens citadas anteriormente, foi usado o PHP, já que um dos envolvidos no projeto possuía um maior conhecimento acerca desta linguagem, e também por ser comumente utilizada no ambiente *web*. Para agilizar o desenvolvimento foi empregue também o CodeIgniter, um *framework* PHP simples e bem completo.

Já com relação à criação do ambiente para o desenvolvimento da aplicação, esta ficou por conta do WampServer e do PhpMyAdmin, considerando o banco de dados MySQL. Tais ferramentas foram escolhidas pelo fato de criarem ambientes *web* semelhantes ao do HostGator, que foi a empresa de hospedagem de sites escolhida para disponibilizar a plataforma. Para localizar a mesma na Internet de forma mais simples, foi necessário a criação de um domínio, realizado através da empresa de registros de nomes para sites, Super Domínios.

Para elaboração dos questionários foi utilizado o conhecimento adquirido com a fundamentação teórica e com a revisão dos questionários de Ribeiro (2005), Teixeira (2015) e Universidade Federal de Juiz de Fora (2017).

3.4 MÉTODOS E PROCEDIMENTOS

O projeto foi constituído pela construção da fundamentação teórica, da etapa de planejamento, criação e testes da plataforma, assim como aplicação da mesma. Paralelamente a estes procedimentos foram elaborados os questionários.

Com aquisição de conhecimento proveniente da construção da fundamentação teórica, iniciou-se a fase de planejamento. Nesta etapa foram levantados os requisitos e pensadas questões relativas ao design da plataforma. Determinou-se também que a mesma seria denominada Gami, em decorrência do fenômeno de gamificação, e construídos dois dos principais diagramas UML, expostos na seção 3.4.3.

Com os requisitos e diagramas gerados, teve-se início o desenvolvimento da plataforma, para o qual se utilizou das linguagens e ferramentas citadas no tópico anterior. Concomitante a mesma, foi elaborado o primeiro questionário, dedicado à descoberta das principais dificuldades presentes no aprendizado de algoritmos na disciplina de AED I.

Encerrando o desenvolvimento, foram feitos testes relativos à segurança, integridade e funcionalidade da plataforma. A mesma foi registrada com o domínio “gamif.top”, através da Super Domínios e hospedada no servidor da HostGator.

Antes de ser disponibilizada para acesso dos estudantes, foi realizada uma visita na turma regular de AED I, para apresentação da plataforma e seus mecanismos, além da definição da premiação para os primeiros colocados do *ranking*. Neste período, também foram observados os resultados do primeiro questionário e foi elaborado o segundo.

Para aplicar a gamificação, a plataforma Gami foi construída para que os administradores pudessem gerenciar os usuários, missões e *feedback*, tudo isso por meio de uma página de controle exclusiva destes. Já os estudantes tinham acesso somente às missões, seus *feedbacks* e o *ranking*.

As missões foram desenvolvidas com base em desafios da Olimpíada Brasileira de Informática (2012, 2013, 2015) da modalidade de programação, nível júnior e nível 1, além da Maratona de Programação da SBC (2012, 2013) realizada pela Sociedade Brasileira de Computação.

A plataforma ficou disponível durante duas semanas. Neste período, foram postadas oito missões, apresentadas no Apêndice C, sendo desconsiderados finais de semana e feriados. Diariamente, os administradores corrigiam as respostas e postavam um *feedback* em parabenização ou com sugestões para correção.

Após a implantação e o encerramento do tempo destinado ao acesso do público, os envolvidos na pesquisa responderam a um questionário sobre a plataforma e o uso da gamificação.

3.4.1 Requisitos funcionais

A plataforma Gami foi desenvolvida baseada nos seguintes requisitos funcionais:

RF 01 - *Create, Read, Update, Delete* (CRUD) do usuário

O sistema permite o cadastro, edição e remoção de usuários, no entanto, essa função é restrita ao administrador.

RF 02 - Realizar login

O sistema possibilita que os usuários façam login em suas respectivas contas para que possam ter acesso às missões e demais funcionalidades.

RF 03 - CRUD da missão

A plataforma permite o cadastro, edição e remoção de missões, entretanto apenas o administrador tem permissão para realizar tais funções.

RF 04 - Iniciar missão

A plataforma possibilita que o usuário receba as missões, porém para ter acesso às mesmas é necessário iniciá-las. Com esta ação, inicia-se também a contagem do tempo que será utilizado para contabilizar a pontuação do usuário nesta missão.

RF 05 - Visualizar missão

Após o usuário iniciar a missão, a aplicação permite que o mesmo visualize as informações desta.

RF 06 - Desistir missão

Depois de iniciar uma missão, o sistema possibilita que o usuário desista da mesma. No entanto, caso o usuário realize esta função, o mesmo não receberá nenhuma pontuação.

RF 07 - Enviar resposta

Após iniciar uma missão, é aberta ao usuário a possibilidade de carregar um arquivo no formato .txt, .c ou .cpp, que será enviado como resposta para missão.

RF 08 - CRUD do *feedback*

A plataforma possibilita que o administrador cadastre, edite e remova *feedbacks*, no entanto, só o mesmo possui tal permissão.

RF 09 - Visualizar *feedback*

Após o cadastro de um novo *feedback* por parte do administrador, a plataforma possibilita ao respectivo usuário a visualização do mesmo.

RF 10 - Enviar correção

Depois de enviar uma primeira resposta para uma missão e receber o *feedback* que a mesma apresenta algum erro, é aberto ao usuário a possibilidade de enviar um novo arquivo no formato .txt, .c ou .cpp, como forma de correção a resposta anterior. Esta correção também receberá um *feedback*, no entanto a pontuação será contabilizada com o tempo da primeira resposta.

RF 11 - Visualizar *ranking*

O sistema permite que os usuários visualize suas respectivas posições no *ranking*, que será construído com as pontuações das missões.

RF 12 - Calcular pontuação

O sistema calculará a pontuação de acordo com o intervalo de tempo contado a partir do momento em que o usuário inicia a missão, sendo no máximo 1000 pontos e o no mínimo 500 pontos.

3.4.2 Requisitos não funcionais

A plataforma também foi baseada nos seguintes requisitos não funcionais:

RNF 01 - O sistema apresentará uma interface objetiva, simples e intuitiva, facilitando a acessibilidade.

RNF 02 - O sistema estará disponível durante duas semanas.

RNF 03 - Apenas o administrador terá acesso direto ao banco de dados

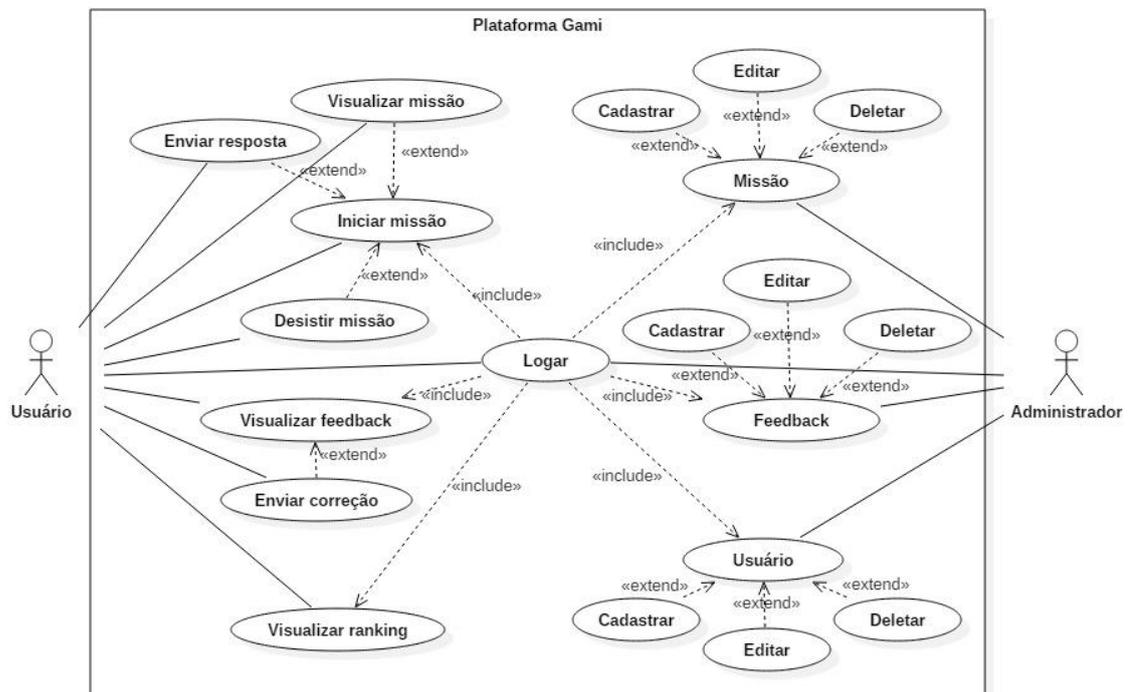
RNF 04 - O sistema aceitará apenas arquivos de resposta nas seguintes extensões: .txt, .c e .cpp.

3.4.3 Diagramas UML

Para uma melhor compreensão da plataforma foram desenvolvidos dois diagramas UML, no caso, o diagrama de caso de uso e o de atividade.

A Figura 3 mostra o diagrama de caso de uso, no qual é possível visualizar todas as interações dos usuários e administradores no sistema.

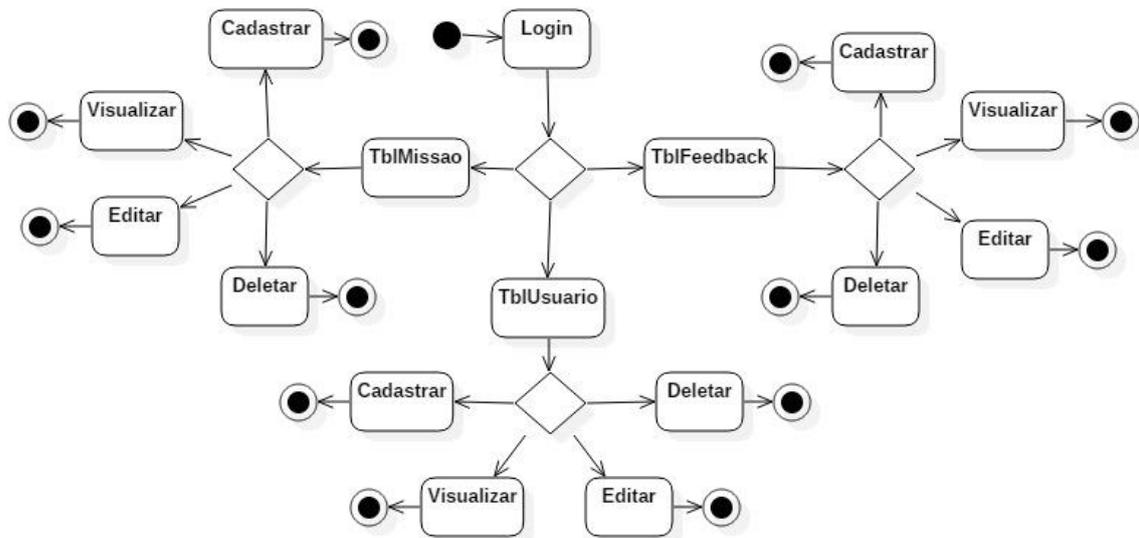
Figura 3 - Diagrama de caso de uso



Fonte: Elaborada pelos autores

A Figura 4 apresenta o diagrama de atividades pela visão do administrador, com este é possível identificar todos os procedimentos que os administradores podem executar.

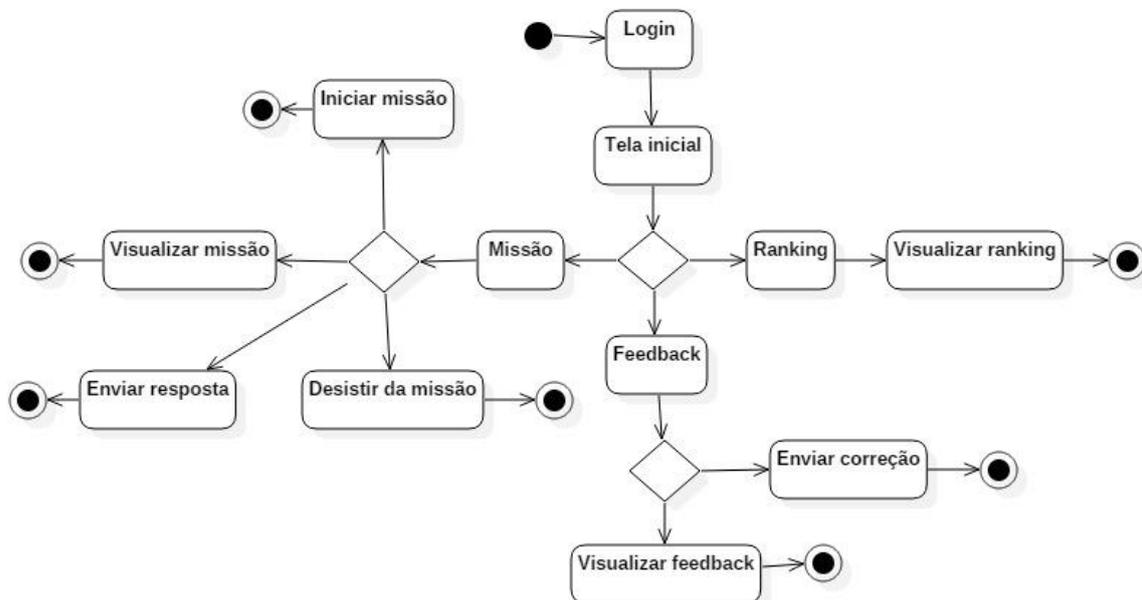
Figura 4 - Diagrama de atividade do administrador



Fonte: Elaborada pelos autores

A Figura 5 mostra o diagrama de atividades pela perspectiva do usuário, possibilitando a análise das ações do usuário na plataforma.

Figura 5 - Diagrama de atividade do usuário



Fonte: Elaborada pelos autores

3.5 TRATAMENTO DOS DADOS

Após a aplicação dos questionários, realizou-se o tratamento dos dados coletados. Para isso, estes foram tabulados e transformados em gráficos. As informações observadas foram analisadas a partir de uma abordagem quantitativa, uma vez que os responsáveis pelo trabalho buscavam compreender as principais características dos estudantes, no que diz respeito ao aprendizado de algoritmos e a importância do uso da gamificação neste. Também foi utilizado o método indutivo, visto que os autores desta pesquisa visaram constatar os benefícios da gamificação realizando uma análise da realidade do próprio IFMG - SJE.

Para auxiliar no tratamento dos dados foi utilizada a ferramenta Microsoft Excel 2010, que possibilitou a tabulação e construção dos gráficos, que viabilizaram uma melhor análise dos resultados descritos no Capítulo 4.

4 RESULTADOS E DISCUSSÕES

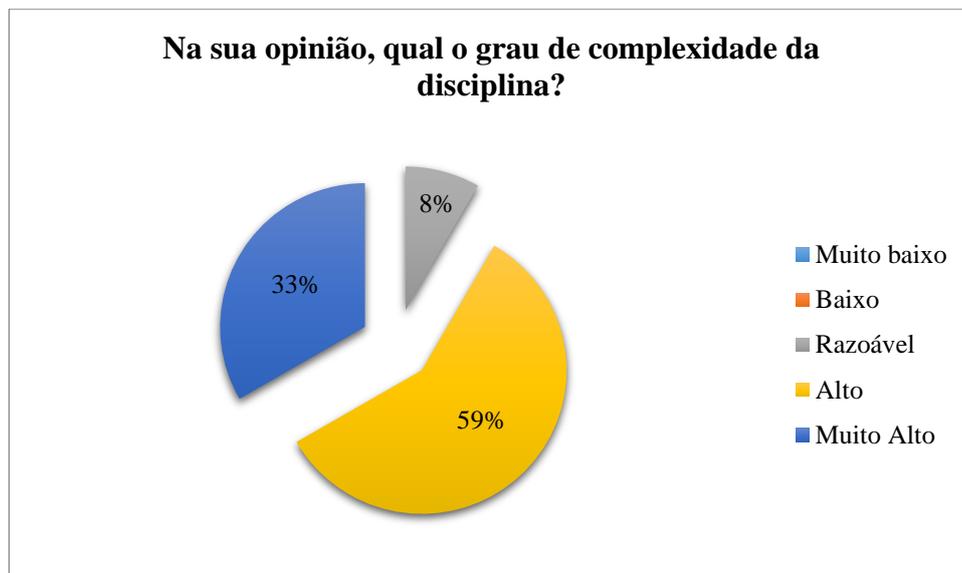
Neste capítulo serão apresentados os resultados obtidos com a aplicação dos questionários e serão discutidas questões relativas as dificuldades no aprendizado de algoritmos e o impacto da gamificação neste contexto.

4.1 RESULTADOS E DISCUSSÕES ACERCA DO PRIMEIRO QUESTIONÁRIO

Nesta seção serão apresentados e discutidos os resultados obtidos com a aplicação do primeiro questionário, que foi destinado à descoberta das principais dificuldades presentes na disciplina de AED I.

O Gráfico 1 mostra que a maior parte dos estudantes consideram que a disciplina possui complexidade alta, o que comprova a teoria dos autores Rapkiewicz *et al.* (2006) e Barcelos, Tarouco e Berch (2009), que apontam em seus estudos a grande dificuldade apresentada na aprendizagem de algoritmos.

Gráfico 1 - Complexidade da disciplina

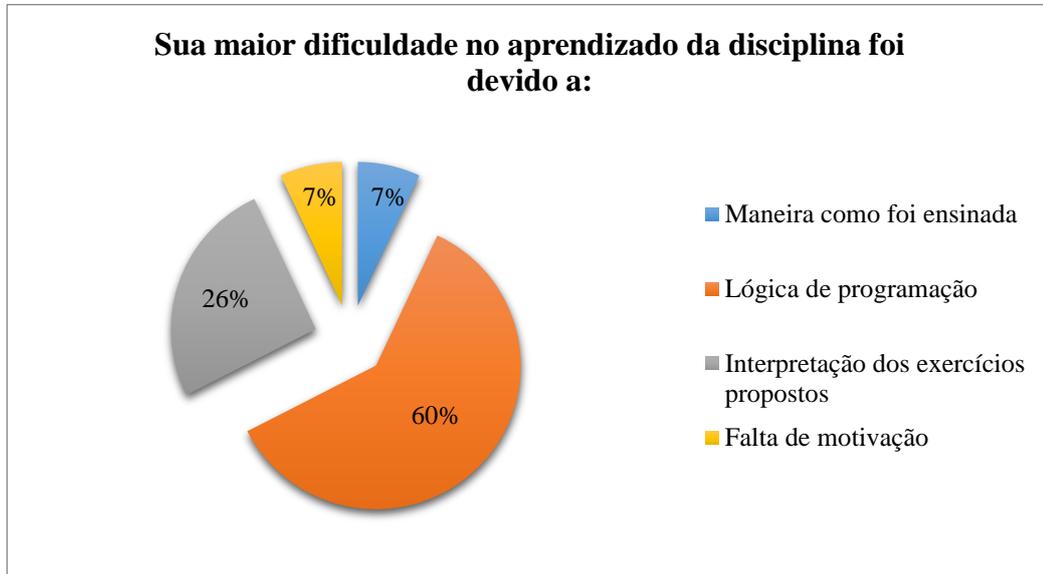


Fonte: Dados da pesquisa

Em conformidade com o trabalho de Rodrigue *et al.* (2014), que aponta a resistência dos estudantes no desenvolvimento do raciocínio lógico, o Gráfico 2 revela que 60% dos

alunos atribuem a sua maior dificuldade no aprendizado da disciplina à Lógica de Programação.

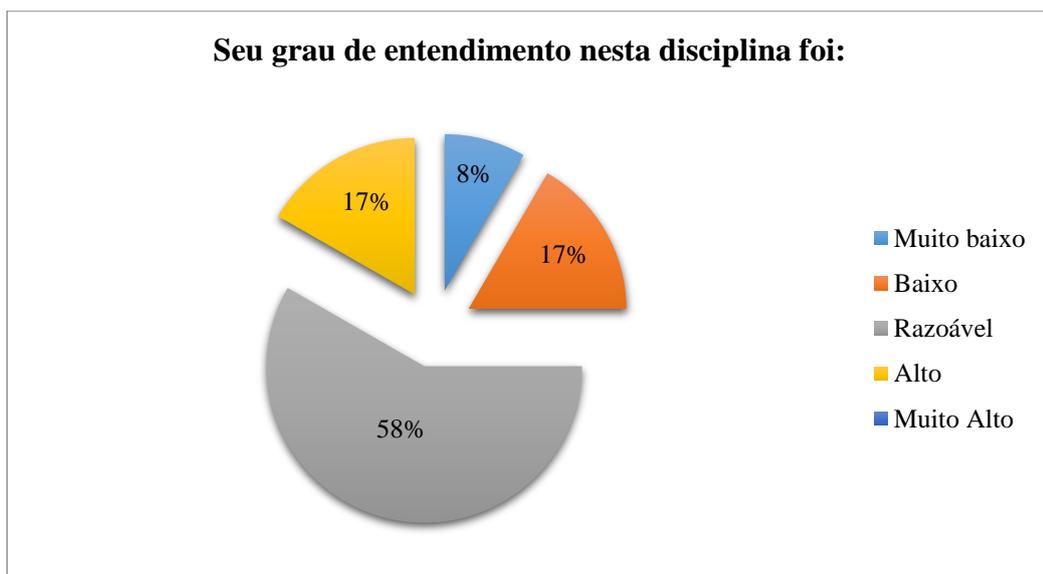
Gráfico 2 - Dificuldades no aprendizado



Fonte: Dados da pesquisa

O Gráfico 3 revela que poucos estudantes apresentam um alto entendimento da disciplina, o que de certa forma é reflexo da dificuldade no aprendizado e também da possível falta de motivação.

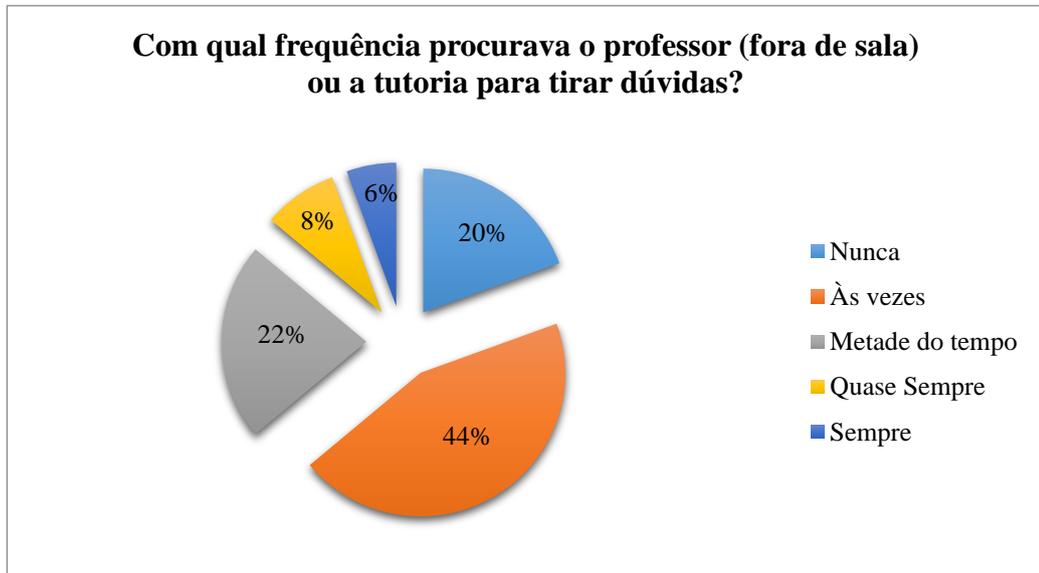
Gráfico 3 - Grau de entendimento



Fonte: Dados da pesquisa

O Gráfico 4 mostra a falta de interesse por parte dos estudantes, que apesar de apresentarem dificuldade na disciplina, não buscam com frequência auxílio com o professor ou tutor. Esta situação evidencia a falta de motivação citada por Rapkiewicz et al. (2006).

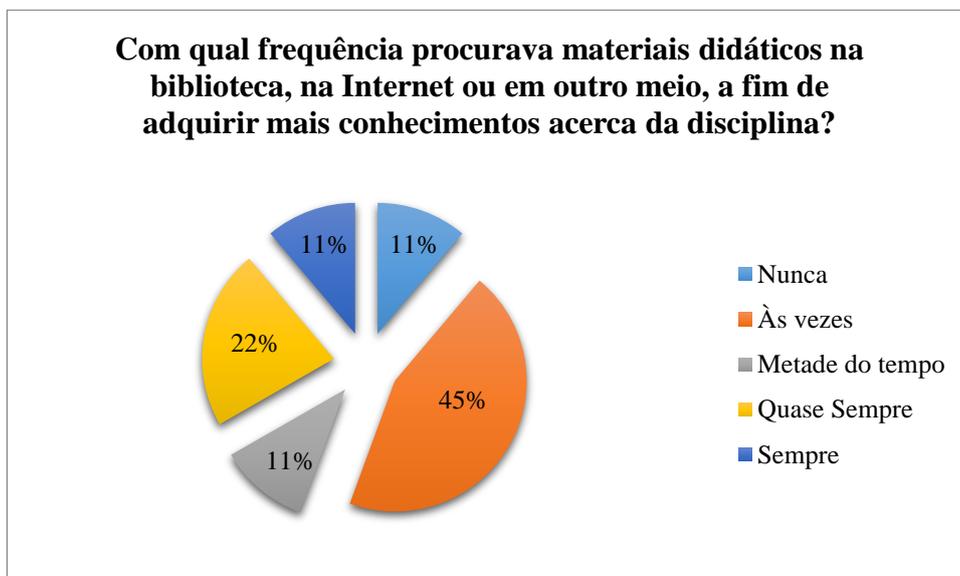
Gráfico 4 - Frequência de busca por ajuda



Fonte: Dados da pesquisa

O Gráfico 5 como o anterior, evidencia a falta de interesse de grande parte dos estudantes, que não buscam adquirir um maior conhecimento acerca da disciplina.

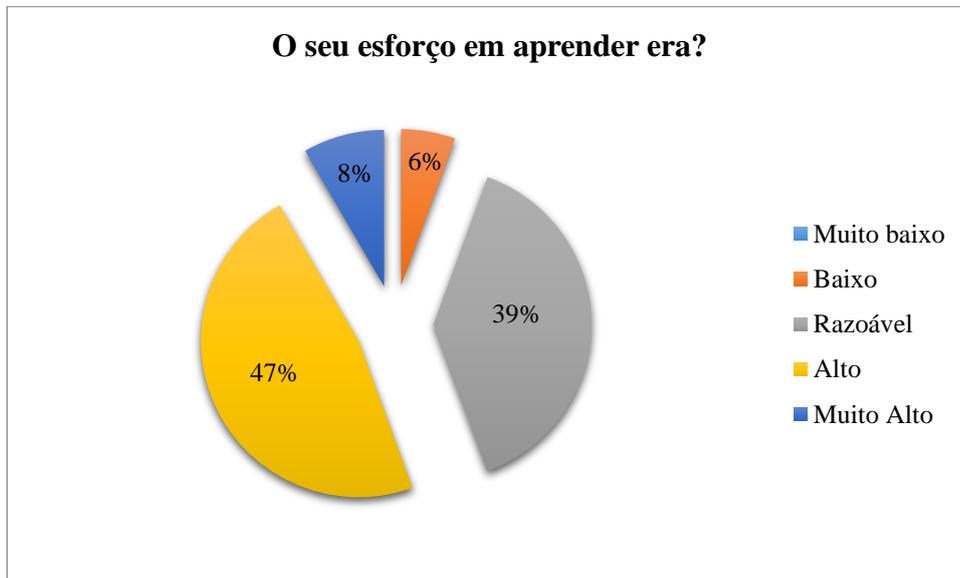
Gráfico 5 - Frequência de busca por materiais didáticos



Fonte: Dados da pesquisa

O Gráfico 6 mostra que uma parte considerável dos estudantes não demonstram muito esforço para compreender o conteúdo, o que de certa forma pode ser resultado da complexidade da disciplina ou da falta de motivação.

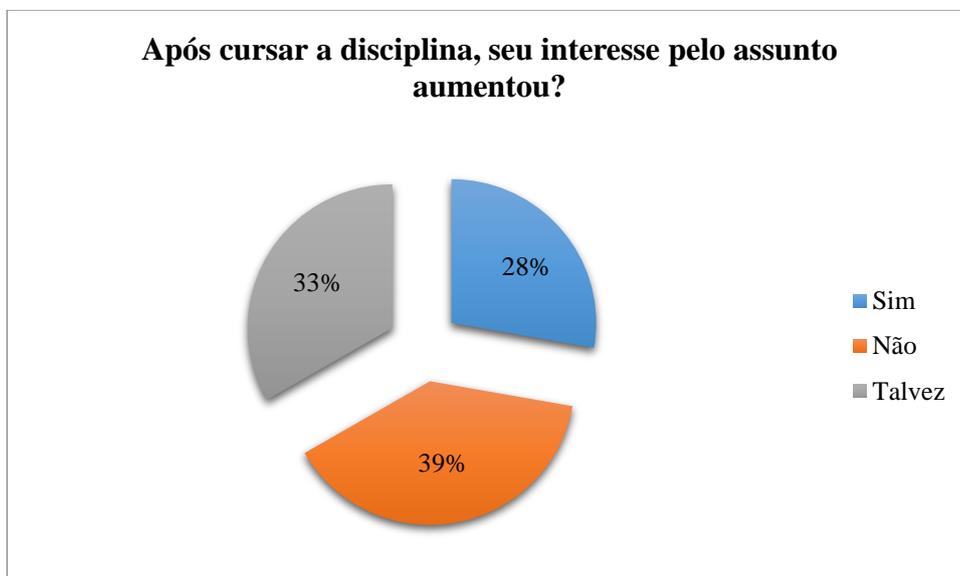
Gráfico 6 - Esforço em aprender



Fonte: Dados da pesquisa

O Gráfico 7 revela que a maior parte dos estudantes, ao fim da disciplina, não demonstraram um crescimento no interesse pela mesma, fato que pode ter sido motivado pela falta de interesse ou até mesmo pela dificuldade.

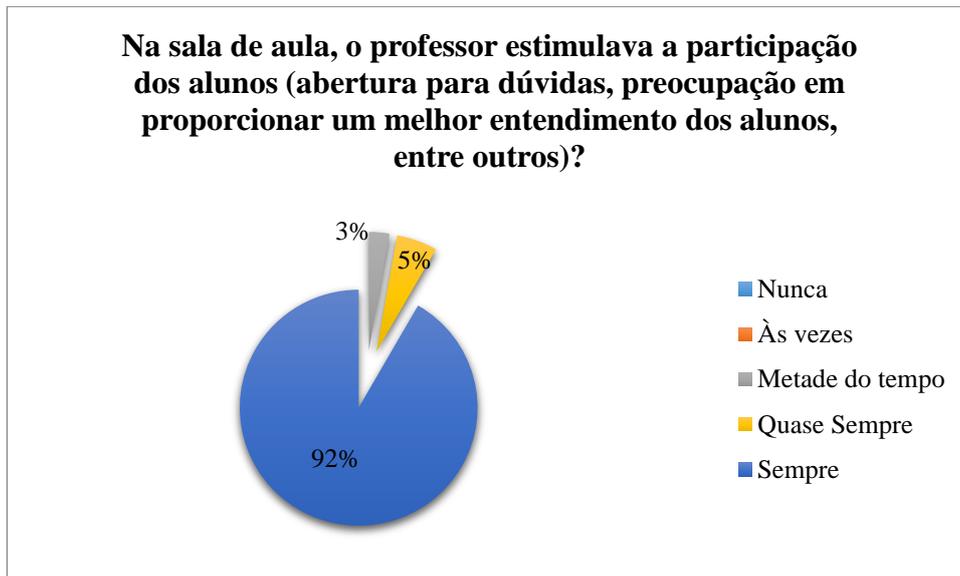
Gráfico 7 - Aumento do interesse pela disciplina



Fonte: Dados da pesquisa

O Gráfico 8 demonstra que no que diz respeito ao professor, este, na maioria das vezes, buscava estimular a participação dos estudantes, o que de certa forma gera motivação, que é apontado por Castoldi e Polinarski (2009) como essencial no meio educacional.

Gráfico 8 - Estímulo por parte do professor



Fonte: Dados da pesquisa

4.2 RESULTADOS E DISCUSSÕES ACERCA DO SEGUNDO QUESTIONÁRIO

Nesta seção serão apresentados e discutidos os resultados obtidos com a aplicação do segundo questionário, que visou revelar as possíveis contribuições que o uso da gamificação trouxe para os estudantes.

O Gráfico 9 revela que, semelhante às turmas que responderam o primeiro questionário, a maior parte dos estudantes da turma regular de AED I consideram a complexidade da disciplina alta, o que novamente confirma a teoria dos autores Rapkiewicz *et al.* (2006) e Barcelos, Tarouco e Berch (2009).

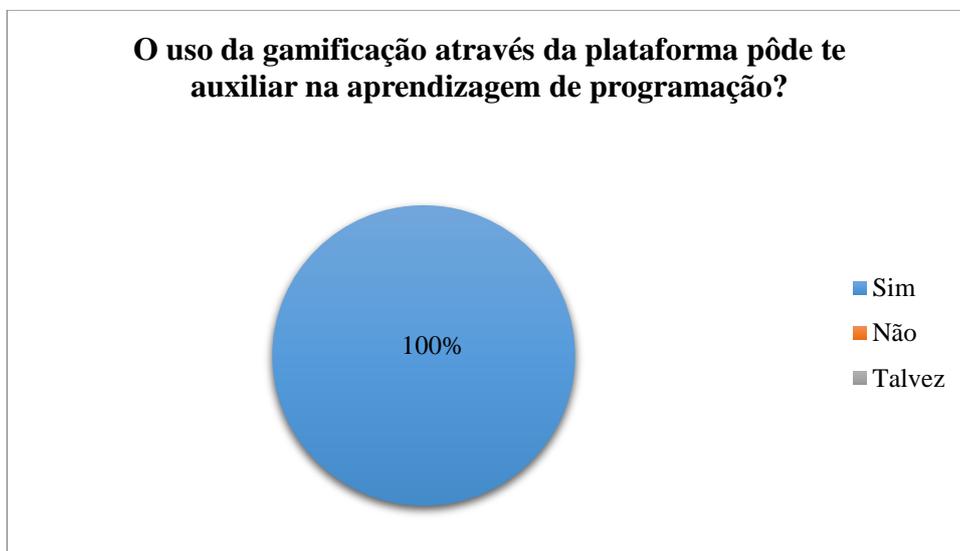
Gráfico 9 - Complexidade da disciplina



Fonte: Dados da pesquisa

O Gráfico 10 demonstra que a gamificação pode auxiliar os estudantes na aprendizagem de programação, o que comprova a ideia de Aguiar (2015), que aponta a gamificação como solução para os problemas de aprendizagem deste conteúdo.

Gráfico 10 - Contribuição da gamificação



Fonte: Dados da pesquisa

Semelhante ao gráfico anterior, o Gráfico 11 revela que a maioria dos estudantes concordam que a gamificação pode contribuir para o aprendizado das disciplinas de programação, o que novamente confirma a ideia Aguiar (2015).

Gráfico 11 - Gamificação na programação



Fonte: Dados da pesquisa

O Gráfico 12 mostra que os estudantes não apresentaram nenhuma dificuldade ao interagir com a plataforma, o que demonstra que a mesma foi desenvolvida de maneira clara e objetiva, simplificando assim, a interação com o usuário.

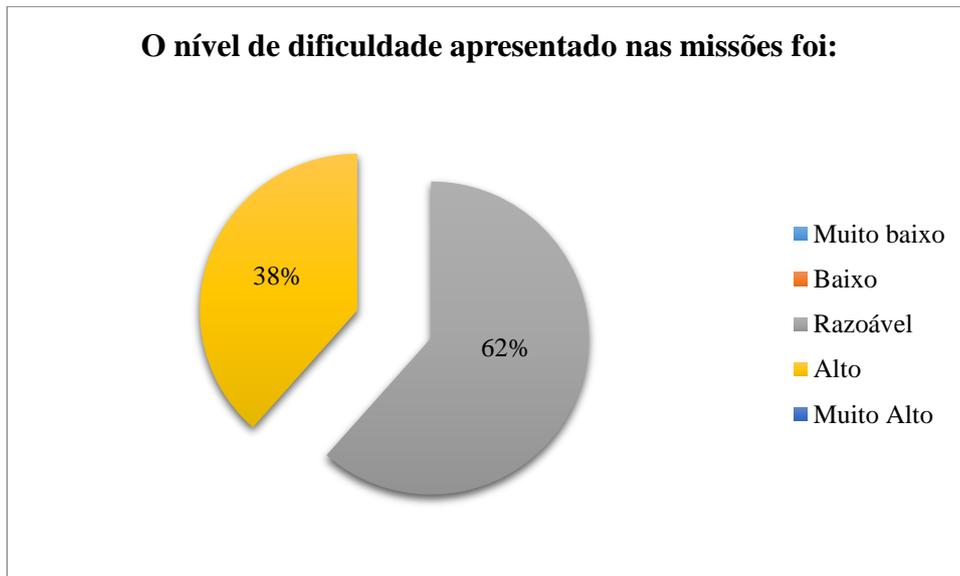
Gráfico 12 - Interação com a plataforma



Fonte: Dados da pesquisa

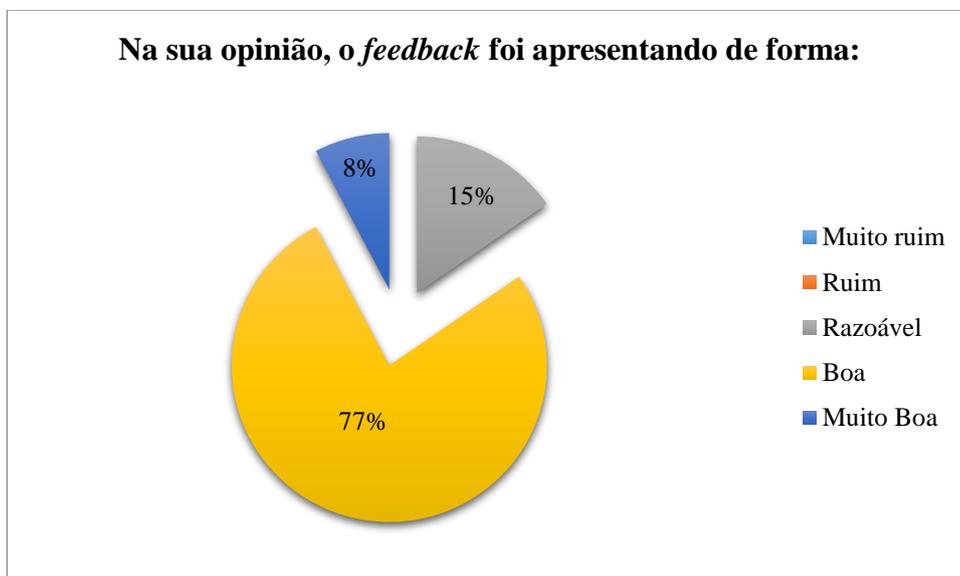
O Gráfico 13 evidencia que a maioria dos estudantes consideraram o nível das missões como razoável e alto, no entanto, de acordo com os dados observados na plataforma, quase todos alcançaram êxito na resolução das missões.

Gráfico 13 - Nível das missões



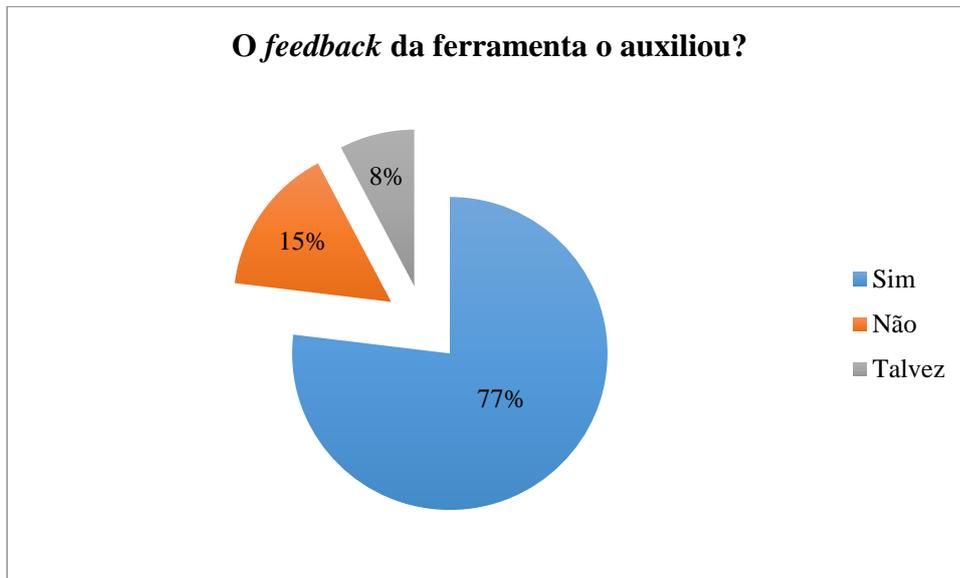
Fonte: Dados da pesquisa

O Gráfico 14 demonstra que grande parte dos estudantes consideraram boa a forma como o *feedback* foi apresentado, o que evidencia o sucesso na implementação das mecânicas de jogos na plataforma.

Gráfico 14 - Apresentação do *feedback*

Fonte: Dados da pesquisa

Já Gráfico 15 revela que a maioria dos estudantes foram auxiliados pelo uso do *feedback*, fato este que indica a importância do mesmo, conforme é citado por Fadel *et al.* (2014) que consideram esta mecânica como uma das quatro características fundamentais da gamificação.

Gráfico 15 - *Feedback*

Fonte: Dados da pesquisa

Com os resultados apresentados nesta seção e na anterior foi possível compreender as dificuldades presentes no aprendizado de algoritmos e as contribuições alcançadas pelo uso da gamificação. Os dados aqui contatados fundamentaram as conclusões descritas no Capítulo 5.

5 CONCLUSÕES

Após análise da literatura observou-se uma ampla difusão da gamificação no âmbito educacional, sendo aplicada em diversos níveis de ensino e sob diferentes perspectivas. Neste contexto, esta técnica demonstra grande versatilidade, podendo ser adaptada em diversas situações e em razão disso, constata-se seu sucesso no ambiente educacional.

Neste sentido, este trabalho teve como propósito analisar as contribuições da gamificação aplicada no ensino de programação. Para tal, foi desenvolvida uma plataforma para viabilizar a aplicação desta técnica. Apesar de ter sido implementada apenas com simples mecânicas de jogos, a plataforma, de acordo com a pesquisa, obteve sucesso ao cumprir seus objetivos.

No que diz respeito às dificuldades no aprendizado, os estudantes que constituíram a amostra demonstraram os mesmos problemas indagados pelos autores do referencial, uma vez que, apesar de apresentarem dificuldades de aprendizagem, os mesmos não se mostravam motivados ou interessados em procurar ajuda.

Com relação às contribuições da gamificação, pode-se observar que o *feedback*, o *ranking* e o desafio proporcionado pelas missões tiveram um impacto significativo nos estudantes, despertando neles um espírito competitivo que estimulou a busca pelo conhecimento.

Como citado anteriormente, a gamificação pode ser aplicada de diversas maneiras, o que possibilita que a mesma possa ser executada de uma forma mais abrangente que a apresentada neste trabalho, podendo ser utilizada como uma metodologia de ensino e não só como uma forma de estudo.

Portanto, conclui-se que esta técnica apresentou resultados relevantes dentro do contexto abordado, possibilitando, futuramente, sua aplicação em outros níveis de ensino ou cursos do próprio instituto. Para trabalhos futuros recomenda-se a aplicação da técnica utilizando metodologias e mecânicas diferentes, que poderão trazer maiores impactos no aprendizado.

REFERÊNCIAS

AGUIAR, Janderson Jason B. **Experiência baseada em Gamificação no Ensino sobre Herança em Programação Orientada a Objetos**, 2015. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/6327/4436>>. Acesso em: 02 abr. 2017.

ALVES, Fábio Pereira; MACIEL, Cristiano. **A Gamificação na Educação: Um Panorama do Fenômeno em Ambientes Virtuais de Aprendizagem**, 2014. Disponível em: <https://www.researchgate.net/profile/Fabio_Alves10/publication/269995356_A_gamificacao_na_educacao_um_panorama_do_fenomeno_em_ambientes_virtuais_de_aprendizagem/links/549daebb0cf2d6581ab64025.pdf>. Acesso em: 05 mar. 2017.

BARCELOS, Ricardo J. S.; TAROUCO, Liane; BERCH, Magda. **O uso de mobile learnig no ensino de algoritmos**, 2009. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/download/13573/14076>>. Acesso em: 16 mar. 2017.

BOMFOCO, Marco Antônio; AZEVEDO, Vitor de Abreu. **Os Jogos Eletrônicos e suas Contribuições para a Aprendizagem na Visão de J. P. Gee**, 2012. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/36411/23515>>. Acesso em: 04 mar. 2017.

BORGES, Simone de S. *et al.* **Gamificação Aplicada à Educação: Um Mapeamento Sistemático**, 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2501>>. Acesso em: 14 mar. 2017.

CAMPOS, André; GARDIMAN, Renato; MADEIRA, Charles. **Uma ferramenta gamificada de apoio à disciplina introdutória de programação**, 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/037.pdf>>. Acesso em: 16 abr. 2017.

CASTOLDI, Rafael; POLINARSKI, Celso Aparecido. **A Utilização de Recursos Didático-Pedagógicos na Motivação da Aprendizagem**, 2009. Disponível em: <<http://atividadeparaeducacaoespecial.com/wp-content/uploads/2014/09/recursos-didatico-pedag%C3%B3gicos.pdf>>. Acesso em: 25 mar. 2017.

CONTE, Tayana Uchôa; MENDES, Emilia; TRAVASSOS, Guilherme Horta. **Processos de Desenvolvimento para Aplicações Web: Uma Revisão Sistemática**, 2005. Disponível em: <https://www.researchgate.net/profile/Guilherme_Travassos/publication/228647730_Processos_de_Developolvimento_para_Aplicacoes_Web_Uma_Revisao_Sistematica/links/00b7d51475c7f69e0b000000.pdf>. Acesso em: 22 abr. 2017.

DUNEBOOK. **5 best PHP frameworks to learn in 2017**, 2017. Disponível em: <<https://www.dunebook.com/5-best-php-frameworks-learn-2017/>>. Acesso em: 15 maio 2017.

FADEL, Luciane Maria *et al.* (Org.). **Gamificação na educação**. São Paulo: Pimenta Cultural, 2014. 300p.

FALCKEMBACH, Gilse Antoninha M.; ARAUJO, Fabricio Viero. **Aprendizagem de Algoritmos: dificuldades na resolução de problemas**, 2006. Disponível em: <<http://periodicos.unesc.net/sulcomp/article/view/916/909>>. Acesso em: 11 abr. 2017.

FARDO, Marcelo Luis. **A Gamificação Aplicada em Ambientes de Aprendizagem**, 2013. Disponível em: <<http://seer.ufrgs.br/renote/article/view/41629/26409>>. Acesso em: 04 mar. 2017.

FIGUEIREDO, Karen da Silva. **Proposta de Gamificação de Disciplinas em um Curso de Sistemas de Informação**, 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2015/081.pdf>>. Acesso em: 16 abr. 2017.

FISCINA, Fabrizio Leandro F.; BORGES, Marcio Vieira. **Algoritmos: Licenciatura em Ciências da Computação**. Salvador: UNEB/GEAD, 2013.

FORBELLONE, André Luiz V.; EBERSPÄCHER, Henri Frederico. **Lógica de Programação: A construção de algoritmos e estruturas de dados**. 3. ed. São Paulo: Prentice Hall, 2005.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo (Org.). **Métodos de pesquisa**. Rio Grande do Sul: Editora da UFRGS, 2009.

GIARETTA, Letícia da Luz *et al.* **Camaleão: Ferramenta de Apoio a Confecção de Jogos Educativos Computadorizados**, 1998. Disponível em: <http://www.niee.ufrgs.br/eventos/RIBIE/1998/pdf/com_pos_dem/218.pdf>. Acesso em: 03 mar. 2017.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

GUIMARÃES, Sueli Édi R.; BORUCHOVITCH, Evely. **O Estilo Motivacional do Professor e a Motivação Intrínseca dos Estudantes: Uma Perspectiva da Teoria da Autodeterminação**, 2004. Disponível em: <<http://www.scielo.br/pdf/%0D/prc/v17n2/22466.pdf>>. Acesso em: 23 mar. 2017.

HAGGLUND, P. **Taking gamification to the next level: a detailed overview of the past, the present and a possible future of gamification**, 2012. Disponível em: <<http://www8.cs.umu.se/education/examina/Rapporter/PerMafrost.pdf>>. Acesso em: 22 abr. 2017.

JACYNTHO, Mark Douglas A. **Processos para Desenvolvimento de Aplicações Web**, 2008. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/09_23_jacyntho.pdf>. Acesso em: 04 maio 2017.

JQUERY FOUNDATION. **What is jQuery?** 2017. Disponível em: <<https://jquery.com>>. Acesso em: 15 maio 2017.

LAB KING HOST. **Editor de código gratuito: 8 opções para você escolher o seu**, 2017. Disponível em: <<https://www.kinghost.com.br/blog/2017/02/editor-de-codigo-gratuito-8-opcoes-para-voce-escolher-o-seu/>>. Acesso em: 18 maio 2017.

LANNA, André Luiz P. M. **Reúso de Processos de Software baseado na componentização de Processos e Conhecimento**, 2009. Disponível em: <http://www.biblioteca.pucminas.br/teses/EngEletrica_LannaAL_1.pdf>. Acesso em: 16 maio 2017.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007.

LEAL, Edvalda Araújo; MIRANDA, Gilberto José; CARMO, Carlos Roberto S. **Teoria da Autodeterminação: uma Análise da Motivação dos Estudantes do Curso de Ciências Contábeis**, 2013. Disponível em: <<http://www.periodicos.usp.br/rcf/article/view/78828/82895>>. Acesso em: 26 mar. 2017.

LOURENÇO, Abílio Alfonso; PAIVA, Maria Olímpia A. **A Motivação Escolar e o Processo de Aprendizagem**, 2010. Disponível em: <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1806-58212010000200012>. Acesso em: 23 mar. 2017.

MARATONA DE PROGRAMAÇÃO. **Caderno de Problemas: Fase 1**, 2012. Disponível em: <<http://maratona.ime.usp.br/hist/2012/primeira-fase/maratona.pdf>> Acesso em: 31 out. 2017.

MARATONA DE PROGRAMAÇÃO. **Caderno de Problemas:** Fase 1, 2013. Disponível em: <<http://maratona.ime.usp.br/hist/2013/primeira-fase/prim-fase2013/maratona.pdf>> Acesso em: 31 out. 2017.

MASTROCOLA, Vicente Martin. **Ludificador:** Um Guia de Referências para o Game Designer Brasileiro. São Paulo: Independente, 2012.

MEDINA, Marco; FERTIG, Cristina. **Algoritmos e Programação:** Teoria e Prática. São Paulo: Novatec Editora, 2006.

OLIMPÍADA BRASILEIRA DE INFORMÁTICA. **Caderno de Tarefas:** Modalidade Programação - Nível 1 - Fase 2, 2012. Disponível em: <http://olimpiada.ic.unicamp.br/passadas/pdf/provas/ProvaOBI2012_prog_f2n1.pdf> Acesso em: 29 out. 2017.

OLIMPÍADA BRASILEIRA DE INFORMÁTICA. **Caderno de Tarefas:** Modalidade Programação - Nível Júnior - Fase 1, 2013. Disponível em: <http://olimpiada.ic.unicamp.br/passadas/pdf/provas/ProvaOBI2013_prog_f1nj.pdf> Acesso em: 29 out. 2017.

OLIMPÍADA BRASILEIRA DE INFORMÁTICA. **Caderno de Tarefas:** Modalidade Programação - Nível Júnior - Fase 1, 2015. Disponível em: <http://olimpiada.ic.unicamp.br/extras/provas/ProvaOBI2015_prog_f1nj.pdf> Acesso em: 30 out. 2017.

OLIMPÍADA BRASILEIRA DE INFORMÁTICA. **Caderno de Tarefas:** Modalidade Programação - Nível Júnior - Fase 2, 2015. Disponível em: <http://olimpiada.ic.unicamp.br/extras/provas/ProvaOBI2015_prog_f2n1.pdf> Acesso em: 30 out. 2017.

PIVA JR., D.; FREITAS, R.L.; PAULA, L.Q. **A Importância da Leitura e da Abstração do Problema no processo de formação do raciocínio lógico-abstrato em alunos de Computação**, 2009. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2009/008.pdf>>. Acesso em: 12 abr. 2017.

PHPMYADMIN. **Bringing MySQL to the web**, 2017. Disponível em: <<https://www.phpmyadmin.net/>>. Acesso em: 04 maio 2017.

PORTAL EDUCAÇÃO. **Linguagem de marcação**, 2013. Disponível em: <<https://www.portaleducacao.com.br/conteudo/artigos/educacao/linguagem-de-marcacao/31639>>. Acesso em: 04 maio 2017.

PORTUGAL, Cristina. **Questões complexas do design da informação e de interação**, 2010. Disponível em:
<https://infodesign.emnuvens.com.br/public/journals/1/V7_n2_2010/ID_v7_n2_2010_01_06_Portugal.pdf?download=1&phpMyAdmin=H8DwcFLEmv4B1mx8YJNY1MFYs4e>.
Acesso em: 28 abr. 2017.

PREECE, J.; ROGERS, Y.; SHARP, H. **Design de interação: além da interação humano-computador**. 3. ed. Porto Alegre: Bookman, 2013.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: uma abordagem profissional**. 8. ed. São Paulo: Mc Graw Hill, 2016.

PWC. **17ª Pesquisa Global Entertainment & Media Outlook**, 2016. Disponível em:
<<http://www.pwc.com.br/pt/outlook-16/games/games-principais-insights.html>>. Acesso em:
03 mar. 2017.

QUADROS, Gerson Bruno F. **Gamificando os Processos de Ensino na Rede**, 2012. Disponível em:
<<http://ueadsl.textolivre.pro.br/2012.2/papers/upload/98.pdf>>. Acesso em: 01 abr. 2017.

RAPKIEWICZ, Clevi Elena *et al.* **Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais**, 2006. Disponível em:
<<http://www.seer.ufrgs.br/index.php/renote/article/view/14284/8203>>. Acesso em: 25 mar. 2017.

RIBEIRO, Filomena. **Motivação e Aprendizagem em Contexto Escolar**, 2001. Disponível em:
<http://www.cefopna.edu.pt/revista/revista_03/es_05_03_FR.htm>. Acesso em: 23 mar. 2017.

RIBEIRO, Maurílio Rizza. **Análise das Dificuldades Relacionadas ao Ensino de Física no Nível Médio**, 2005. Disponível em: <
http://www.infis.ufu.br/infis_sys/pdf/MAURILIO%20RIZZA%20RIBEIRO.pdf>. Acesso em: 18 ago. 2017.

RODRIGUES, Thiago Oliveira *et al.* **Projeto para Ensino de Linguagem de Programação Utilizando a Plataforma Raspberry Pi**, 2014. Disponível em:
<<http://www.abenge.org.br/cobenge-2014/Artigos/130353.pdf>>. Acesso em: 25 mar. 2017.

SCHLEMMER, Eliana. **Gamificação em Espaços de Convivência Híbridos e Multimodais: Design e cognição em discussão**, 2014. Disponível em:
<<https://www.revistas.uneb.br/index.php/faeaba/article/view/1029/709>>. Acesso em: 04 mar. 2017.

SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. 9. ed. Porto Alegre: Bookman, 2011.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?**, 2015. Disponível em: <<https://canaltech.com.br/materia/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. Acesso em: 04 maio 2017.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STARUML. **StarUML 2**: Um sofisticado modelador de software, 2016. Disponível em: <<http://staruml.io/>>. Acesso em: 18 maio 2017.

TECHTUDO. **Com WampServer tenha um servidor web completo em seu computador**, 2013. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/wampserver.html>>. Acesso em: 15 maio 2017.

TEIXEIRA. **Principais Dificuldades de Aprendizagem em Matemática no Ensino Fundamental**, 2015. Disponível em: <www.dmcejp.unir.br/menus_arquivos/1787_principais_dificuldades_de_aprendizagem_em_matematica_no_ensino_fundamental.pdf>. Acesso em: 18 ago. 2017.

TIOBE. **TIOBE Index for April 2017**, 2017. Disponível em: <<https://www.tiobe.com/tiobe-index>>. Acesso em: 08 abr. 2017.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA. **Formulário de Avaliação de Disciplina e Professor para Impressão**, 2017. Disponível em: <http://www.ufjf.br/fisdiurno/files/2017/05/Formulario_de_Avalia%C3%A7%C3%A3o_de_Disciplina.pdf>. Acesso em: 18 ago. 2017.

UOL HOST. **O que é PHP?**, 2016. Disponível em: <<http://www.uolhost.uol.com.br/academia/noticias/tecnologia/2016/04/15/o-que-e-php.html#rml>>. Acesso em: 04 maio 2017.

W3SCHOOLS. **Bootstrap Get Started**, 2017f. Disponível em: <https://www.w3schools.com/bootstrap/bootstrap_get_started.asp>. Acesso em: 17 maio 2017.

_____. **CSS Tutorial**, 2017b. Disponível em: <https://www.w3schools.com/css/css_intro.asp>. Acesso em: 07 abr. 2017.

_____. **HTML5 Tutorial**, 2017a. Disponível em:
<https://www.w3schools.com/html/html_intro.asp>. Acesso em: 07 abr. 2017.

_____. **jQuery Introduction**, 2017e. Disponível em:
<https://www.w3schools.com/jquery/jquery_intro.asp>. Acesso em: 15 maio 2017.

_____. **JS Tutorial**, 2017c. Disponível em: <<https://www.w3schools.com/js/default.asp>>.
Acesso em: 08 abr. 2017.

_____. **PHP 5 Tutorial**, 2017d. Disponível em:
<<https://www.w3schools.com/php/default.asp>>. Acesso em: 22 abr. 2017.

ZEMEL, Tércio. **Web Design Responsivo: Páginas adaptáveis para todos os dispositivos**. São Paulo: Casa do Código, 2015.

ZENORINI, Rita da Penha C.; SANTOS, Acácia Aparecida A. **Escala de Metas de Realização como Medida da Motivação para Aprendizagem**, 2010. Disponível em:
<<http://www.psicorip.org/Resumos/PerP/RIP/RIP041a5/RIP04430.pdf>>. Acesso em: 25 mar. 2017.

ZICHERMANN, Gabe; CUNNINGHAM, Christopher. **Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps**. Sebastopol, CA: O'Reilly Media, Inc. 2011.

ZIVIANI, Nivio. **Projeto de Algoritmos: com implementação em PASCAL e C**. 3. ed. São Paulo: Cengage Learning, 2011.

APÊNDICE A - PRIMEIRO QUESTIONÁRIO

QUESTIONÁRIO DE AVALIAÇÃO DE ALGORITMOS E ESTRUTURA DE DADOS

I

Nome (opcional): _____

Sobre a disciplina

- 1) O conteúdo era apresentado com objetivos claros?

<input type="checkbox"/> Nunca	<input type="checkbox"/> Quase Sempre
<input type="checkbox"/> Às vezes	<input type="checkbox"/> Sempre
<input type="checkbox"/> Metade do tempo	

- 2) O material didático fornecido ou citado era:

<input type="checkbox"/> Péssimo	<input type="checkbox"/> Bom
<input type="checkbox"/> Ruim	<input type="checkbox"/> Excelente
<input type="checkbox"/> Regular	

- 3) Na sua opinião, qual o grau de complexidade da disciplina?

<input type="checkbox"/> Muito baixo	<input type="checkbox"/> Alto
<input type="checkbox"/> Baixo	<input type="checkbox"/> Muito Alto
<input type="checkbox"/> Razoável	

- 4) Na sua opinião, esta disciplina precisa de mais tempo semanal (mais créditos)?

<input type="checkbox"/> Sim	<input type="checkbox"/> Não	<input type="checkbox"/> Talvez
------------------------------	------------------------------	---------------------------------

- 5) Sua maior dificuldade no aprendizado da disciplina foi devido a:
 - Maneira como foi ensinada
 - Lógica de programação
 - Interpretação dos exercícios propostos
 - Falta de motivação

Sobre você

- 1) Como era sua frequência nas aulas?

<input type="checkbox"/> Muito baixa	<input type="checkbox"/> Alta
<input type="checkbox"/> Baixa	<input type="checkbox"/> Muito Alta
<input type="checkbox"/> Razoável	

- 2) Com qual frequência procurava o professor (fora de sala) ou a tutoria para tirar dúvidas?
- Nunca Quase Sempre
 Às vezes Sempre
 Metade do tempo
- 3) Com qual frequência procurava materiais didáticos na biblioteca, na Internet ou em outro meio, a fim de adquirir mais conhecimentos acerca da disciplina?
- Nunca Quase Sempre
 Às vezes Sempre
 Metade do tempo
- 4) Seu grau de entendimento nesta disciplina foi:
- Muito baixo Alto
 Baixo Muito Alto
 Razoável
- 5) O seu esforço em aprender era?
- Muito baixo Alto
 Baixo Muito Alto
 Razoável
- 6) Na sua opinião, a metodologia de ensino apresentada era satisfatória?
- Sim Não Razoável
- 7) Após cursar a disciplina, seu interesse pelo assunto aumentou?
- Sim Não Talvez
- 8) Você adquiriu conhecimentos novos com esta disciplina?
- Sim Não Talvez

Sobre o Professor

- 1) O grau de domínio do professor sobre o conteúdo era:
- Muito baixo Alto
 Baixo Muito Alto
 Razoável
- 2) As aulas preparadas pelo professor eram:
- Péssimas Boas
 Ruins Excelentes
 Regulares

- 3) A explicação do conteúdo era:
- Péssima Boa
- Ruim Excelente
- Regular
- 4) As avaliações produzidas pelo professor eram compatíveis com o conteúdo apresentado em sala?
- Nunca Quase Sempre
- Às vezes Sempre
- Metade do tempo
- 5) Na sala de aula, o professor estimulava a participação dos alunos (abertura para dúvidas, preocupação em proporcionar um melhor entendimento dos alunos, entre outros)?
- Nunca Quase Sempre
- Às vezes Sempre
- Metade do tempo
- 6) O grau de disponibilidade do professor fora de sala de aula para os alunos (horário de atendimento, resposta a e-mail, entre outros) era:
- Baixo Razoável Alto
- 7) Você faria outra disciplina com o mesmo professor ou o indicaria para outro aluno?
- Sim Não Talvez

APÊNDICE B - SEGUNDO QUESTIONÁRIO

QUESTIONÁRIO DE AVALIAÇÃO DO APRENDIZADO, DA PLATAFORMA GAMI E DA TÉCNICA DE GAMIFICAÇÃO

Nome (opcional): _____

Sobre o aprendizado

- 1) Na sua opinião, qual o grau de complexidade das disciplinas de programação?

<input type="checkbox"/> Muito baixo	<input type="checkbox"/> Alto
<input type="checkbox"/> Baixo	<input type="checkbox"/> Muito Alto
<input type="checkbox"/> Razoável	
- 2) Sua maior dificuldade no aprendizado de programação é devido a:
 - Maneira como é ensinada
 - Lógica de programação
 - Interpretação dos exercícios propostos
 - Falta de motivação

Sobre a plataforma Gami

- 3) Você teve dificuldade para usar a plataforma?

<input type="checkbox"/> Sim	<input type="checkbox"/> Não
------------------------------	------------------------------
- 4) O nível de dificuldade apresentado nas missões foi:

<input type="checkbox"/> Muito baixo	<input type="checkbox"/> Alto
<input type="checkbox"/> Baixo	<input type="checkbox"/> Muito Alto
<input type="checkbox"/> Razoável	
- 5) Na sua opinião, o feedback foi apresentando de forma:

<input type="checkbox"/> Muito ruim	<input type="checkbox"/> Boa
<input type="checkbox"/> Ruim	<input type="checkbox"/> Muito boa
<input type="checkbox"/> Razoável	
- 6) O feedback da ferramenta o auxiliou?

<input type="checkbox"/> Sim	<input type="checkbox"/> Não	<input type="checkbox"/> Às vezes
------------------------------	------------------------------	-----------------------------------
- 7) De alguma forma a plataforma aumentou seu empenho na disciplina de AED I?

<input type="checkbox"/> Sim	<input type="checkbox"/> Não	<input type="checkbox"/> Talvez
------------------------------	------------------------------	---------------------------------

Sobre a gamificação

8) O uso da gamificação através da plataforma pode te auxiliar na aprendizagem de programação?

Sim

Não

Talvez

9) Na sua opinião, quanto a gamificação pode ajudar no aprendizado das disciplinas de programação?

Nada

Razoável

Pouco

Muito

APÊNDICE C - MISSÕES

Missão #1

Dois ou Um

O jogo Dois ou Um (ou Zerinho ou Um) é frequentemente utilizado para escolher um ganhador entre três ou mais jogadores. Para quem não conhece, o jogo funciona da seguinte maneira: cada jogador escolhe entre dois ou um, depois de um comando que normalmente é quando todos gritam “Dois ou um”, todos mostram o valor escolhido. O dois é representado por dois dedos levantados, normalmente o indicador e médio, e o um por um único dedo, normalmente o indicador.

O vencedor é o que mostra o valor diferente de todos os outros, se isso não acontecer, não há vencedor na partida.

Pedro, Lucas e João jogam Dois ou Um o tempo todo e decidiram pedir para que você crie um programa que vai ajudá-los.

Entrada

A entrada é composta de uma única linha, que contém três inteiros P, L e J, indicando, respectivamente, os valores escolhidos por Pedro, Lucas e João.

Saída

Seu programa deve gerar uma única linha com o nome do vencedor. Caso não haja vencedor deve ser mostrado um ‘*’ (asterisco).

Exemplo

Entrada 2 2 1	Saída João
Entrada 1 1 1	Saída *
Entrada 2 1 1	Saída Pedro

Missão #2

Equação de Segundo Grau

Marcos está cursando Engenharia Florestal e está tendo a disciplina de Fundamentos da Matemática. Apesar de ser bom em matemática, equação? de segundo grau nunca foi o seu forte. Então para ajudar, te pediu que criasse uma calculadora.

Obs: sua calculadora não precisa calcular as raízes quando o delta não tiver uma raiz exata.

Entrada

A entrada é constituída de uma única linha contendo três inteiros A, B e C.

Saída

Caso haja raízes, na saída deve aparecer as duas raízes, mesmo sendo iguais. Caso não haja deve aparecer o caractere asterisco (*).

Exemplo

Entrada 2 8 -24	Saída 2 -6
Entrada 1 2 -1	Saída *
Entrada -2 20 -50	Saída 5 5
Entrada 2 -1 0	Saída 0 0.5

Missão #3

Salvadores

Um recente terremoto na província de Tangshan na China afetou muitos edifícios e foi detectado que o principal dique de contenção sofreu grandes danos na parte subterrânea da sua estrutura. Caso não seja consertado rapidamente pode ocasionar um desmoronamento.

O conserto só pode ser realizado por mergulhadores, mas como é muito profundo é uma operação de risco. No entanto, mesmo assim alguns cidadãos vendo o risco para seus familiares e amigos decidiram se voluntariar.

Como é tradicional em missões de risco, cada voluntário recebeu uma placa com um número para identificá-lo. Ao terminar, cada mergulhador devolve a placa de identificação.

A operação foi um sucesso apesar de que alguns voluntários não voltaram. Você foi contratado para criar um programa que ajude na identificação desses que não voltaram.

Entrada

A entrada é composta por duas linhas. A primeira é composta por dois inteiros N e R , indicando, respectivamente, o número de voluntários que mergulharam, e o número de voluntários que retornaram do mergulho. A segunda contém R inteiros, indicando os voluntários que retornaram do mergulho. As placas são identificadas com um inteiro que está entre 1 e N .

Saída

Seu programa deve produzir uma linha contendo os números dos voluntários que não retornaram do mergulho. Caso todos tenham retornado a saída deve ser um único caractere ‘*’ (asterisco).

Exemplo

Entrada 5 3 3 1 5	Saída 2 4
Entrada 6 6 6 1 3 2 5 4	Saída *

Missão #4

Tecnologia no futebol

Atualmente com o aumento da tecnologia, a mesma tem sido aplicada em vários contextos. O futebol, uma das paixões do brasileiro, não ficou fora dessa. Sendo assim, um árbitro pensou em usar as várias câmeras que normalmente gravam os jogos, em conjunto com um programa, para auxiliar na marcação de impedimentos. As câmeras iriam registrar as posições dos jogadores e o programa definiria se foi ou não impedimento.

A regra de impedimento funciona da seguinte maneira: considerando três jogadores, L, R e D, sendo, respectivamente, o jogador que lança a bola, o que recebe e o último jogador defensor, será impedimento somente se o jogador R estiver no seu campo de ataque, mais a frente do jogador L e mais avançado que o jogador D. Se o jogador R estiver no seu campo de defesa ou na linha do meio campo, ele não estará em impedimento.

O árbitro que teve esta ideia, não possui conhecimento em programação, por isso pediu a você que fizesse este programa. Lembrando que, considerando o campo com 100 m e dadas as posições dos jogadores L, R e D, no momento exato do lançamento, haverá impedimento se e somente se a seguinte condição for verdadeira:

$$(R > 50) \text{ e } (L < R) \text{ e } (R > D)$$

Entrada

A entrada deve ser composta por uma linha, contendo os três inteiros L, R e D, sendo estes as posições dos respectivos jogadores.

Saída

O programa deve imprimir uma única linha, contendo um caractere, que deve ser “S” caso haja impedimento, ou “N” caso contrário.

Exemplos

Entrada 35 60 75	Saída N
Entrada 55 68 67	Saída S

Missão #5

Jogo de Cartas

Carlos e João estão jogando um novo jogo de cartas. Neste jogo, inicialmente, é colocado um número par de cartas sobre uma mesa, cada uma contendo números inteiros dispostos em uma sequência qualquer. Carlos começa o jogo pegando uma das cartas das pontas. Então, João também pega uma das cartas das pontas e novamente Carlos pega uma carta das pontas. E assim por diante, até João pegar a última carta. Carlos, o primeiro a jogar, tem como objetivo somar o maior número de pontos contabilizados com os valores das cartas escolhidas. João, o segundo jogador, deve atrapalhar Carlos e fazer com que ele consiga o menor número de pontos possível. Resumindo, ambos querem realizar seu objetivo no jogo, Carlos querendo somar mais pontos e João tentando atrapalha-lo. Você tem como missão, criar um programa que ajude Carlos a analisar uma determinada sequência de cartas e indicar o número máximo de pontos que ele pode conseguir nessa partida.

Entrada

A entrada deve ser composta por duas linhas, a primeira contendo um inteiro N, que indica o número de cartas sobre a mesa. E a segunda contendo os N inteiros, que descrevem os números da sequência de cartas.

Saída

Como resultado o programa deve imprimir uma única linha, contendo um inteiro, indicando o maior número de pontos que Carlos conseguirá na partida.

Exemplo

Entrada 4 0 -3 5 10	Saída 10
Entrada 4 47 50 -3 7	Saída 54
Entrada 4 10 20 30 40	Saída 60

Missão #6

Estudo de programação

Normalmente, Carlos e mais três amigos se reúnem para estudar. Cada fim de semana, eles se juntam na casa de um deles para estudar uma disciplina. Na próxima vez, eles estudarão programação na casa de Carlos. Sabendo que sua casa é pequena e moram muitos familiares, para ficarem a vontade, deverão se restringir somente ao quarto de Carlos. No entanto, tem-se um problema, pois o quarto de Carlos só possui uma única tomada. Mas, por outro lado, todos possuem uma régua de tomadas. Pensando nesta situação, Carlos desafiou seus amigos a criarem um programa que, dado o número de régua e o de tomadas em cada uma, determinasse quantas tomadas poderiam ser disponibilizadas no quarto dele.

Entrada

A entrada consiste de uma linha com quatro inteiros positivos T1, T2, T3, T4, indicando o número de tomadas de cada uma das quatro régua.

Saída

O programa deve imprimir uma linha contendo um único número inteiro, indicando o número máximo de notebooks que podem ser conectados num mesmo instante.

Exemplos

Entrada 2 4 3 2	Saída 8
Entrada 6 6 6 6	Saída 21
Entrada 5 4 3 6	Saída 15
Entrada 5 3 4 7	Saída 16

Missão #7

Fórmula 14

No GP da Austrália, vai ser realizada a sensacional final do mundial de fórmula 14. Normalmente, nestes GPs existem técnicos que analisam as corridas e fornecem a todo momento dados da mesma, como velocidade máxima, número de ultrapassagens, possíveis vencedores, dentre outras. Você, como um novo membro destes técnicos, tem a missão de construir um programa que determine qual dos dois carros que lideram a corrida será o possível vencedor do mundial, considerando as distâncias até a linha de chegada e que mantenham a mesma velocidade.

Entrada

A entrada deste programa deve ser composta por duas linhas, cada uma descrevendo um dos carros que lidera a corrida. A descrição dos carros consiste de três inteiros N, D e V indicando, respectivamente, o número do carro, a sua distância à linha de chegada em metros, e a sua velocidade, em quilômetros por hora. Os números dos carros são distintos.

Saída

Como saída, o programa deve imprimir uma linha contendo um número inteiro X, indicando o número do possível vencedor.

Exemplos

Entrada	Saída
45 900 40 17 300 20	17

Entrada	Saída
1 1000 100 2 1000 99	1

Entrada	Saída
5 350 30 6 400 30	6

Missão #8

Campeonato Italiano

O campeonato italiano se encontra nas últimas rodadas, e tem como possíveis ganhadores Milan e Juventus. Como em qualquer outro campeonato, cada vitória conta três pontos e cada empate, um ponto. Fica melhor colocado, o time com mais pontos. Em caso de empate no número de pontos, fica o time que tiver maior saldo de gols. Se o número de pontos e o saldo de gols forem os mesmos, então os dois times estão empatados.

Considerando os números de vitórias e empates, e os saldos de gols dos dois times, você tem como objetivo determinar qual deles está melhor colocado no campeonato, ou se eles estão empatados.

Entrada

A entrada é descrita em uma única linha, que contém seis inteiros, separados por um espaço em branco: MV, ME, MS, JV, JE, JS, que são, respectivamente, o número de vitórias, de empates e o saldo de gols do Milan, e o número de vitórias, de empates e o saldo de gols da Juventus.

Saída

O programa deve imprimir uma única linha contendo a letra “M”, se o Milan estiver melhor classificado que a Juventus, ou a letra “J”, se a Juventus estiver melhor classificada. Caso estejam empatados a linha deve conter apenas o caractere “*” (asterisco).

Exemplos

Entrada 10 5 18 11 1 18	Saída M
Entrada 10 5 18 11 2 18	Saída *
Entrada 10 5 18 11 2 20	Saída J