



Programa Interdisciplinar de Pós-Graduação em

# **Computação Aplicada**

**Mestrado Acadêmico**

Sandro José Ribeiro da Silva

TRILUA: um ambiente gamificado para apoio ao ensino de  
lógica de programação

São Leopoldo, 2016.

Sandro José Ribeiro da Silva

**TRILUA: um ambiente gamificado para apoio ao ensino de lógica de programação**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – UNISINOS

Orientador: Professor Dr. Sandro José Rigo

São Leopoldo

2016

Dados Internacionais de Catalogação na Publicação (CIP)

S586e Silva, Sandro José Ribeiro da

TRILUA: um ambiente gamificado para apoio ao ensino de  
lógica de programação /

Sandro José Ribeiro da Silva. – 2016.

95 f. :il.

Dissertação (Mestrado) - Universidade do Vale do Rio dos  
Sinos, Programa de Pós-Graduação em Computação Aplicada,  
2016.

Orientador: Prof. Dr. Sandro José Rigo.

1. Jogos educacionais. 2. Gamificação. 3. Mineração de dados  
educacionais. 4. Ensino de linguagem de programação. I. Rigo,  
Sandro José. II. Título.

CDU 004:37

Bibliotecária responsável: Sabrina Clavé Eufrásio CRB-10/1670

Sandro José Ribeiro da Silva

TRILUA: um ambiente gamificado para apoio ao ensino de lógica de programação

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 03/11/2016

BANCA EXAMINADORA

---

Professora Dra. Patrícia Behar – Universidade Federal do Rio Grande do Sul - UFRGS

---

Professora Dr. Jorge Luis Victória Barbosa – Universidade do Vale dos Sinos - UNISINOS

Prof. Dr. Sandro José Rigo (Orientador)

Visto e permitida a impressão  
São Leopoldo,

Prof. Dr. Sandro José Rigo  
Coordenador PPG em Computação Aplicada

*À Deus, por me permitir chegar até aqui.*

*À minha família, em especial a minha esposa Carla pela ajuda integral neste trabalho.*

*Ao meu filho, Lucca.*

## **AGRADECIMENTOS**

Agradeço à minha família pelo apoio.

Ao meu orientador professor Dr. Sandro José Rigo pela ajuda e paciência dedicada a mim e a este trabalho.

Ao colega Marcio Bigolin pela parceria na execução deste trabalho e do projeto Gamificando o ensino da lógica de programação.

Ao Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul – IFRS pela cedência da estrutura para a aplicação das oficinas de gamificação.

Aos alunos do primeiro ano do curso técnico em Desenvolvimento de Sistemas Integrado ao Ensino Médio pela disposição em participar desta proposta.

Aos docentes das disciplinas de lógica de programação, em especial ao colega Igor Lorenzato Almeida, pela colaboração e contribuição imprescindíveis para construção do ambiente gamificado e validação dos dados minerados.

A todos que, direta ou indiretamente, contribuíram para a construção deste trabalho.

## RESUMO

O desenvolvimento de habilidades de programação de sistemas computacionais é uma necessidade crescente, devido ao amplo uso de recursos computacionais nas mais diversas áreas. Ao mesmo tempo, é conhecida a deficiência existente quanto à quantidade de profissionais sendo graduados nesta área. Alguns estudos indicam dificuldades dos estudantes e ao mesmo tempo falta de metodologias adequadas como possíveis elementos contribuindo para este contexto, corroborando a necessidade de desenvolvimento de pesquisas sobre o aprendizado de linguagens de programação. Entre as possíveis soluções para este problema de motivação, o desenvolvimento de um ambiente gamificado como ferramenta de ensino para linguagens de programação vem sendo explorado em projetos de pesquisa e também em opções comerciais. Uma das deficiências observadas nestas iniciativas é justamente a falta de suporte aos professores para acompanhamento da evolução dos alunos. Buscando atender esta necessidade, o presente trabalho propõe um ambiente de apoio ao ensino de lógica de programação cujo diferencial é a inclusão de recursos de análise do comportamento dos alunos, voltados para o apoio ao professor. Desta forma, o trabalho proposto alia aos jogos eletrônicos o monitoramento on-line de suas etapas, através do uso de técnicas de mineração de dados educacionais. Com base em um framework para Gamificação, foi definido e desenvolvido um ambiente Web para ensino da linguagem Lua, com aspectos de Gamificação e Mineração de Dados Educacionais. Este ambiente foi utilizado em avaliações com alunos do ensino técnico, tendo sido observados resultados promissores nos aspectos motivacionais. As avaliações envolvendo a identificação de vantagens geradas para os professores com uso dos dados sobre o comportamento dos alunos também foram positivas e indicam um bom potencial para esta abordagem.

**Palavras-Chave:** Jogos Educacionais; Gamificação; Mineração de Dados Educacionais; Ensino de Linguagem de Programação.

## ABSTRACT

The development of computer systems programming skills is a growing necessity, due to the wide use of computational resources in different areas. At the same time, it is known the deficiency with respect to the amount of professionals being graduated in this area. Some studies indicates difficulties of students and lack of adequate methodologies as possible elements contributing to this context, supporting the need to develop research on learning programming languages. As a possible solution to this problem of motivation, the development of a gamified environment as a teaching tool for programming languages is being explored in research projects and also commercial options. One of the deficiencies observed in these initiatives is precisely the lack of support to teachers to follow up of the evolution of students, which consists in one of the differentials of the proposed work. In this way, the work integrates to electronic games the online monitoring through the use of educational data mining techniques. Based on the framework for gamification, has been defined and developed a web environment to the Lua language teaching, with aspects of gamification and education data mining. This environment has already been tested preliminarily with technical education students, being observed promising results. A new stage of development and testing is foreseen to deepening the identification of advantages generated for teachers with the use data on the behavior of students.

**Keywords:** Educational games; gamification; Educational data mining; programming language teaching.



## LISTA DE FIGURAS

Figura 1: Contextualização da gamificação.....	24
Figura 2: Etapas da Mineração de dados educacionais .....	29
Figura 3: Tela de desenvolvimento da plataforma App Inventor .....	31
Figura 4: Tela de desenvolvimento da plataforma URI Online Judge .....	33
Figura 5: Tela de desenvolvimento da plataforma Scratch .....	34
Figura 6: Tela de desenvolvimento da plataforma Code Combat .....	35
Figura 7: Tela de desenvolvimento da plataforma Code Hunt.....	36
Figura 8: Tela de desenvolvimento da plataforma CodeAcademy.....	37
Figura 9: Tela de desenvolvimento da plataforma Alice.....	38
Figura 10: Perfil de gosto e frequência por jogos eletrônicos .....	49
Figura 11: Perfil do local aonde mais acontece a utilização dos jogos .....	49
Figura 12: Perfil da plataforma de jogo utilizada .....	50
Figura 13: Perfil da plataforma de jogo favorita .....	50
Figura 14: Perfil da categoria de jogo mais utilizada .....	50
Figura 15: Perfil sobre o auxilio de jogos no aprendizado .....	51
Figura 16: Perfil do uso de jogos eletrônicos em redes sociais.....	51
Figura 17: Perfil do compartilhamento das evoluções do uso de jogos eletrônicos em redes sociais .....	52
Figura 18: Diagrama de Caso de Uso do ambiente gamificado .....	58
Figura 19: Diagrama de Sequencia do ambiente gamificado.....	59
Figura 20: <i>Framework</i> estrutural MVC.....	62
Figura 21: Tela Inicial de <i>login</i> .....	63
Figura 22: Tela Inicial .....	63
Figura 23: Tela de navegação.....	64
Figura 24: Tela de nível da unidade variável .....	64
Figura 25: Tela com os detalhamentos dos menus usados nas telas de níveis .....	65
Figura 26: Tela com os detalhamentos dos demais itens presentes em todos os níveis do ambiente .....	66
Figura 27: Tela módulo do professor .....	67
Figura 28: Exemplo de registro gerado pelo ambiente gamificado.....	68
Figura 29: Conhecimento dos alunos sobre lógica de programação. ....	73
Figura 30: Opinião dos alunos sobre a interface gráfica e navegação do ambiente.....	73
Figura 31: Opinião dos alunos sobre a jogabilidade em termos de funcionalidade. ....	74
Figura 32: Opinião dos alunos sobre a jogabilidade em termos de diversão. ....	74

Figura 33: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da manipulação de variáveis em Lua. ....	75
Figura 34: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização de operadores matemáticos básicos em Lua. ....	75
Figura 35: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização de matrizes e vetores em Lua. ....	76
Figura 36: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da cláusula <i>if</i> em Lua. ....	76
Figura 37: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura <i>for</i> em Lua. ....	77
Figura 38: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura <i>repeat-until</i> em Lua. ....	77
Figura 39: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura <i>while</i> em Lua. ....	78
Figura 40: Quanto os alunos consideram que o ambiente os motivou no aprendizado da linguagem Lua. ....	78
Figura 41: Desempenho dos Alunos nas Avaliações em Lógica de Programação.....	80
Figura 42: Visão do processo realizado.....	81
Figura 43: Base de dados brutos.....	81
Figura 44: Relatório gerado pela API do Weka do algoritmo <i>Simple K-Means</i> .....	82
Figura 45: Relatório gerado pela API do Weka do algoritmo <i>Aprioris</i> .....	84
Figura 46: Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação amplia a motivação dos alunos? .....	85
Figura 47: Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação melhora o desempenho do aluno na disciplina?.....	85
Figura 48: Em sua opinião os dados fornecidos pela utilização do ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação contribui no planejamento da disciplina de lógica de programação?.....	86

## LISTA DE TABELAS

Tabela 1: Índices de desempenho relacionados à disciplina de Lógica de Programação no IFRS - Campus Canoas .....	22
Tabela 2: Tabela comparativa entre os softwares analisados .....	39
Tabela 3: <i>Framework</i> para gamificação .....	41
Tabela 4: Etapas Gerais do Framework de Gamificação Adotado.....	53
Tabela 5: Resultado da primeira avaliação na disciplina de lógica de programação .....	71
Tabela 6: Comparativo das avaliações na disciplina de lógica de programação.....	79

## LISTA DE SIGLAS

EAD	Educação a Distância
ECGBL 2016	10th European Conference on Games Based Learning
EDM	Mineração de dados educacionais
IFPR	Instituto Federal de Educação, Ciência e Tecnologia do Paraná
IFRS	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
MMORPG	Massively Multiplayer Online Role Play Games
PBL	Points, Badges and Leaderboards
PIPCA	Programa Interdisciplinar de Pós-Graduação em Computação Aplicada
SEMESP	Sindicato das Instituições Privadas de Ensino Superior do Estado de São Paulo
TI	Tecnologia da Informação
UNISINOS	Universidade do Vale do Rio dos Sinos

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
1.1	JUSTIFICATIVA E OBJETIVOS .....	17
1.2	ASPECTOS DA METODOLOGIA ADOTADA.....	17
1.3	ORGANIZAÇÃO DO TEXTO .....	18
<b>2</b>	<b>FUNDAMENTAÇÃO .....</b>	<b>19</b>
2.1	O ENSINO E APRENDIZAGEM .....	19
2.2	A DISCIPLINA DE LÓGICA DE PROGRAMAÇÃO .....	19
2.2.1	<i>Problemas no Ensino e Aprendizagem de Lógica de Programação</i> .....	20
2.3	GAMIFICAÇÃO .....	23
2.3.1	<i>A Gamificação no processo de ensino aprendizagem</i> .....	25
2.4	LINGUAGEM DE PROGRAMAÇÃO LUA .....	27
2.5	MINERAÇÃO DE DADOS EDUCACIONAIS.....	29
<b>3</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>31</b>
3.1	APP INVENTOR .....	31
3.2	URI ONLINE JUDGE .....	32
3.3	SCRATCH .....	34
3.4	CODE COMBAT .....	34
3.5	CODE HUNT .....	36
3.6	CODEACADEMY.....	37
3.7	ALICE.....	37
3.8	ANÁLISE DE TRABALHOS RELACIONADOS .....	39
<b>4</b>	<b>METODOLOGIA .....</b>	<b>41</b>
4.1	FRAMEWORK DE GAMIFICAÇÃO .....	41
4.2	QUESTIONÁRIO DE PESQUISA DE OPINIÃO.....	48
4.3	INSTANCIAMENTO DO AMBIENTE GAMIFICADO .....	52
4.3.1	<i>Identificação geral do contexto de aplicação</i> .....	53
4.3.2	<i>Identificação e Compreensão do Problema</i> .....	53
4.3.3	<i>Definição do Escopo</i> .....	54
4.3.4	<i>Construção da Narrativa</i> .....	55
4.3.5	<i>Mecânicas e Dinâmicas</i> .....	56
4.3.6	<i>Competências e Ações</i> .....	57
4.3.7	<i>Ambiente/Natureza da Gamificação</i> .....	57
4.3.8	<i>A Gamificação na Prática</i> .....	57
<b>5</b>	<b>DESENVOLVIMENTO DO AMBIENTE.....</b>	<b>58</b>
5.1	MODELAGEM DO AMBIENTE .....	58
5.2	NÍVEIS DISPONÍVEIS NO AMBIENTE .....	60
5.2.1	<i>Tutorial</i> .....	60
5.2.2	<i>Variáveis</i> .....	60
5.2.3	<i>Operações Matemáticas</i> .....	60
5.2.4	<i>Tabelas</i> .....	61
5.2.5	<i>Cláusulas Condicionais</i> .....	61
5.2.6	<i>Estruturas de Repetição</i> .....	61
5.3	IMPLEMENTAÇÃO DO AMBIENTE .....	61
5.4	INTERFACES DO AMBIENTE .....	62
5.4.1	<i>Telas de Navegação</i> .....	63
5.4.2	<i>Telas de Nível do Ambiente</i> .....	64
5.5	COMPONENTES DOS NÍVEIS E MECÂNICA DO AMBIENTE .....	65
5.6	MÓDULO DE MINERAÇÃO DE DADOS EDUCACIONAIS.....	67
5.6.1	<i>Organização dos Dados</i> .....	68

5.6.2	<i>Mineração de Dados</i> .....	69
5.6.3	<i>Relatório Final</i> .....	70
<b>6</b>	<b>AVALIAÇÕES</b> .....	<b>71</b>
6.1	AVALIAÇÕES COM OS ALUNOS.....	71
6.2	AVALIAÇÕES COM OS PROFESSORES .....	80
6.3	ANÁLISE DOS RESULTADOS DE TESTES DE USO DO SISTEMA.....	86
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>89</b>
7.1	CONTRIBUIÇÕES REALIZADAS .....	90
7.2	TRABALHOS FUTUROS .....	91
<b>8</b>	<b>REFERÊNCIAS</b> .....	<b>92</b>

# 1 INTRODUÇÃO

O desenvolvimento de habilidades de programação de sistemas computacionais vem sendo destacado como uma necessidade crescente, devido ao amplo uso de recursos computacionais nas mais diversas áreas. Ao mesmo tempo é conhecida a deficiência existente quanto à quantidade de profissionais sendo graduados nesta área (SOARES, 2016). Alguns estudos indicam dificuldades dos estudantes e ao mesmo tempo falta de metodologias adequadas como possíveis elementos contribuindo para este contexto (VALENTIM, 2009), corroborando a necessidade de desenvolvimento de pesquisas sobre o aprendizado de linguagens de programação.

Segundo FERREIRA, GONZAGA e SANTOS (2010), “não existe uma *fórmula mágica* para se aprender a programar”. O que é possível observar com o estudo destes autores é que se pode identificar alguns pontos indicativos e importantes nos quais os alunos desejosos em aprender algum tipo de linguagem de programação devem focar-se. Um destes pontos é o da repetição, pois para realmente aprender uma linguagem de programação, é necessário exercitá-la por diversas vezes. O outro ponto de suma importância é a dedicação e o foco no aprendizado, pois repetir um exercício por obrigação não irá contribuir no aprendizado e apenas reforçará a memorização dos passos a seguir, mas nunca o aprendizado pleno.

Existem diversas hipóteses sobre o que leva um aluno a encontrar dificuldades para aprender lógica de programação. Para MENDES (2002) e MOTIL e EPSTEIN (2000), as dificuldades no aprendizado da lógica de programação podem estar atreladas a diversos aspectos, tais como: a) Programar requer um alto nível de abstração, habilidade que muitas pessoas têm dificuldade em aprimorar; b) As linguagens de programação utilizadas em muitas instituições de ensino possuem sintaxes demasiadamente complexas, dificultando a compreensão dos algoritmos.

Para SEBESTA (2011), a abstração pode ser descrita como a capacidade de isolar um problema de um ambiente para então estudá-lo separadamente. Na computação, podemos exemplificar o ato de abstrair da seguinte maneira: suponhamos que exista um processo que é executado em diferentes pontos de um sistema, abstrai-lo seria o ato de observar que este processo poderia ser requerido pelas diferentes partes do sistema, ao invés de copiar o processo para cada um dos pontos onde ele é utilizado.

GOMES e MENDES (2007) ressaltam que outra grande barreira no aprendizado da programação é a dificuldade do aluno em conseguir acompanhar o ritmo de aula proposto pelos professores, uma vez que, na grande maioria das instituições de ensino, as aulas não são individuais e existe grande diversidade entre os alunos. De acordo com MENDES (2002), deve ser questionado pelos professores como pode ser exigido de um aluno que ele se dedique ao exercício da programação dia após dia, quando a falta de motivação é justamente um dos fatores que mais contribui para a evasão de cursos relacionados à Tecnologia da Informação (TI).

Finalmente, para ALMEIDA et al. (2002) o principal obstáculo que impede muitos alunos de aprenderem a programar é a falta de motivação ao depararem-se com as ferramentas e linguagens de programação cada vez mais sofisticados, que requerem cada vez mais conhecimento e abstração por parte dos programadores.

Uma proposta que vem sendo adotada para buscar uma forma de motivar estes alunos, trazendo o ensino da programação para dentro de um ambiente familiar ao mesmo, é a adoção

do uso dos jogos eletrônicos. Para PAULO, JÚNIOR e BONIATI (2015) podemos fazer uso do conceito de gamificação, que consiste em utilizar conceitos, mecânicas e dinâmicas de jogos para motivar os alunos no desempenho de outras ações não relacionadas a jogos.

A utilização de abordagens envolvendo Jogos Digitais e Gamificação como elementos motivadores para o aprendizado vem sendo adotada com relativo sucesso, conforme indicado na literatura da área, apesar de se tratar de esforços recentes e ainda em desenvolvimento, em grande parte (ALVES, [s.d.]). Entretanto, além da motivação para o processo de aprendizado, devem ser tratados com igual cuidado os processos de reconhecimento das necessidades dos alunos e o seu encaminhamento pelos professores. Uma forma de qualificar os resultados obtidos com o aspecto motivacional destes recursos de Jogos Digitais e Gamificação vem sendo observada a partir da integração dos recursos de Mineração de Dados Educacionais e *Learning Analytics*, aspecto que será explorado neste trabalho.

Com base nos referenciais acima apresentados, a motivação para este trabalho consiste no desenvolvimento de um ambiente gamificado que possibilite ao aluno aprender conceitos básicos da lógica de programação através do uso da linguagem de programação Lua<sup>1</sup>. Este ambiente será utilizado também como base para a coleta de dados a serem analisados com ferramentas de Mineração de Dados Educacionais e *Learning Analytics*. A escolha de linguagem Lua deu-se por esta mostrar-se uma linguagem de sintaxe simples, porém com uma robusta gama de recursos.

O ambiente desenvolvido possui como foco o auxílio aos alunos que possuem algum tipo de dificuldade no aprendizado de lógica de programação, em especial aos que se encontram no período inicial de cursos técnicos de nível médio da área da tecnologia da informação, para facilitar a exploração da adequação do uso da abordagem gamificada com este público. Acredita-se que este ambiente seria capaz de estimular os alunos, que possuem este perfil descrito, no que se refere a duas das principais dificuldades no aprendizado da lógica de programação: a falta de motivação e a dificuldade de abstração.

O desenvolvimento de um ambiente gamificado como ferramenta de ensino de linguagens de programação vem sendo explorado em projetos de pesquisa e também em opções comerciais (ALVES, [s.d.]). No mercado existem diversas opções, como por exemplo, os jogos *Code Combat* e *Code Hunt*. Uma das deficiências observadas nestas iniciativas é justamente a falta de suporte aos professores para acompanhamento da evolução dos alunos, o que consiste em um dos diferenciais do trabalho proposto. Desta forma, o trabalho alia aos jogos eletrônicos o monitoramento on-line de suas etapas, monitoramento este que se apresenta através do uso de técnicas de mineração de dados educacionais e *Learning Analytics*.

Portanto, o tema principal deste trabalho está associado com a exploração da Gamificação como estratégia no ensino de Lógica de programação no Curso Técnico em Desenvolvimento de Sistemas integrado ao Ensino Médio. Como principal justificativa, considera-se os novos cenários da educação, frente aos avanços da ciência e da tecnologia, nos quais o modo de ensino e aprendizagem deixou de ser exclusividade de um único ambiente, sendo que junto aos novos dispositivos e as novas tecnologias emerge a necessidade da atualização nas estratégias educacionais. Os ambientes educacionais/digitais necessitam se tornar pedagogicamente atrativos ao educando.

---

<sup>1</sup> Lua é uma linguagem de programação poderosa, rápida e leve, projetada para estender aplicações. Disponível em: <https://www.lua.org/portugues.html>



## 1.1 Justificativa e objetivos

O ensino da lógica de programação passa anualmente por adaptações significativas, entretanto estas adaptações não são suficientes para suprir as dificuldades enfrentadas pelos estudantes. Vários são os fatores que contribuem para a dificuldade no ensino e aprendizagem, dentre eles, podemos destacar, no que tange ao ensino, a metodologia tradicional com ênfase excessiva na construção de algoritmos repetitivos e desconexos à realidade do estudante, a busca inadequada de novos recursos pedagógicos e, no que se refere à aprendizagem, o conceito pré-formado de que “lógica de programação é uma disciplina difícil”, a necessidade de dedicação e empenho na realização das atividades propostas pelos professores e falta de motivação KAPP (2012).

Uma possível solução a esta problemática seria o desenvolvimento de um ambiente gamificado, a partir do conceito de gamificação, como uma ferramenta de ensino e aprendizagem que contribua para tornar a disciplina mais atrativa e contextualizada. A gamificação, segundo KAPP (2012), utiliza elementos, mecânicas e dinâmicas de jogos para envolver as pessoas em contextos não relacionados a jogos. Por sua vez, segundo o mesmo autor, o jogo eletrônico “é um sistema em que os jogadores se envolvem em um desafio abstrato, definido por regras, interatividade e *feedback*, o que resulta em um desfecho quantificável, muitas vezes provocando uma reação emocional”.

A questão de pesquisa definida para este trabalho, está definida da seguinte forma: Quais os ganhos para o processo de ensino e aprendizagem da lógica de programação que podem ser obtidos através da gamificação, como forma de despertar o interesse intrínseco do aluno, aliada com a Mineração de Dados Educacionais, como elemento de apoio ao professor?

O objetivo geral do trabalho consiste em analisar a potencialidade do uso das técnicas de gamificação aplicadas a uma ferramenta de apoio ao ensino e aprendizagem da lógica de programação em cursos técnicos integrados ao ensino médio e as vantagens advindas da sua integração com a Mineração de Dados Educacionais, como elemento de apoio à atividade do professor.

Os objetivos específicos consistem em:

- Construir um modelo computacional do ambiente proposto, utilizando o conceito de gamificação e mineração de dados educacionais;
- Construir um protótipo do ambiente, utilizando o conceito de gamificação e mineração de dados educacionais;
- Avaliar o protótipo do ambiente desenvolvido;
- Contribuir na discussão da utilização de ambientes educacionais/digitais pedagogicamente atrativos ao aluno.

## 1.2 Aspectos da metodologia adotada

A metodologia adotada para este trabalho está descrita a seguir.

Em um primeiro momento, foi realizada uma ampla pesquisa bibliográfica e também o estudo em grupo de trabalho com foco em Educação e Gamificação, com o intuito de ampliar o entendimento do problema a ser tratado e também ampliar o conhecimento do estado da arte na área de gamificação e de ensino de linguagem de programação e de lógica de programação. Parte destes estudos em grupo de trabalho ocorreu no contexto de atividade acadêmica cursada pelo autor do trabalho junto do Programa de Pós-Graduação em Educação da UNISINOS, cuja temática foi a Gamificação e seus desdobramentos do ponto de vista

pedagógico. Além disso, foram realizados acompanhamentos práticos com alunos e professores atuando no ensino técnico, com vistas a colher subsídios para a caracterização de aspectos relevantes.

Com estes passos foram identificados requisitos para um ambiente gamificado de apoio à educação de lógica de programação, o que permitiu gerar um protótipo que possibilitou testes práticos. O desenvolvimento do ambiente gamificado para este projeto, nomeado "Tri-Lua", foi implementado a partir dos seguintes requisitos: primeiramente foi feito um levantamento de requisitos através do uso de um *framework* de gamificação com a finalidade de descobrir quais são as características necessárias a um ambiente educativo para o ensino de lógica de programação. Após isto, ocorreu o desenvolvimento do ambiente propriamente dito. Depois da conclusão do desenvolvimento do ambiente, foi aplicada uma bateria de testes em uma turma de alunos de um curso técnico da área da TI. Por fim, foi realizado uma relação cruzando as premissas levantadas durante a fase de concepção e desenvolvimento do ambiente, com os dados obtidos na fase de testes.

As etapas seguintes do trabalho envolvem a caracterização das necessidades dos professores e seu efetivo atendimento com base nos recursos de Mineração de dados educacionais. Foram então realizados novos testes práticos, tendo em vista a verificação das hipóteses do trabalho, efetuados em contextos educacionais reais. Os dados gerados foram analisados para compor a escrita do volume final do trabalho, bem como de artigos científicos na área.

### **1.3 Organização do texto**

O texto está organizado como segue. O capítulo 2 apresenta elementos de fundamentação para o trabalho. O capítulo 3 descreve trabalhos relacionados, tanto do ponto de vista de trabalhos comercialmente disponíveis como de trabalhos científicos na área. A metodologia geral adotada é descrita no capítulo 4. A abordagem proposta para a implementação do ambiente gamificado está definida no capítulo 5, sendo que avaliações realizadas são apresentadas e comentadas no capítulo 6. As considerações finais estão organizadas no capítulo 7.

## **2 FUNDAMENTAÇÃO**

Neste capítulo estão resumidos aspectos que formaram a base que permitiu o desenvolvimento e justificativa deste trabalho. Como este se caracteriza como um trabalho com aspectos interdisciplinares, diversas áreas foram elencadas e resumidas. Alguns aspectos descritos neste capítulo de fundamentação também servem para elucidar parte dos requisitos do trabalho e descrever elementos do campo de aplicação escolhido.

### **2.1 O ensino e aprendizagem**

Segundo BODENAVE e PEREIRA (1991), o aluno é o agente de seu processo de aprendizagem e ninguém pode aprender por ele. Sua aprendizagem está vinculada ao seu nível de motivação, sua maturidade, sua personalidade, seu modo de perceber a realidade e o funcionamento de suas estruturas cognitivas, entre outros fatores.

Considerando que cada aluno tem um potencial perceptivo próprio, que cada um deles tem habilidades diferenciadas, que todos podem aprender, mas não aprendem da mesma forma e que as aulas são ministradas como se todos pudessem aprender da mesma maneira, é importante fornecer aos alunos dos cursos que possuem a disciplina de Algoritmos a possibilidade de trabalhar esse conteúdo conforme seu ritmo.

O professor como ator do processo ensino-aprendizagem precisa estar ciente de seu papel e é necessário que ele possua a percepção sobre o que o aluno não consegue assimilar. Mais do que ensinar, é necessário refletir sobre sua prática e revisar suas metodologias para que possa atingir o seu objetivo maior, que é o de fazer com que o aluno aprenda.

É muito importante que a primeira disciplina que visa trabalhar os conceitos básicos de resolução de problemas através da lógica forneça ao aluno subsídios que sustentem abstrações e o formalismo exigido nas áreas da tecnologia. Faz-se necessário estabelecer, desde o início, estratégias para organizar as ideias, o raciocínio lógico e a representação simbólica, a fim de que o aluno incorpore esses hábitos e os utilize sempre na resolução dos problemas durante o decorrer da disciplina e/ou curso e, futuramente em sua vida profissional.

### **2.2 A Disciplina de Lógica de Programação**

A disciplina de Lógica de Programação é fundamental para a aprendizagem de qualquer linguagem de programação constituindo-se como primeiro passo para que o aluno domine seus conceitos para avançar na aprendizagem. Por seu caráter introdutório esta disciplina localiza-se, geralmente, no primeiro ano/semestre dos cursos na área de desenvolvimento de sistemas em Tecnologia da Informação, sendo ele de nível técnico, tecnológico ou bacharelado.

Na disciplina de Lógica de Programação o desenvolvimento do raciocínio é crucial para que os objetivos de ensinar e aprender sejam atingidos. Como então “despertar” e “motivar” os alunos?

O processo mental para a aprendizagem da Lógica de Programação é novo para o aluno que está iniciando nas áreas de tecnologia e isso representa uma dificuldade, gerando, muitas vezes, resistência ao aprendizado. Nos primórdios da computação para que fosse possível programar um computador foi criada a linguagem Assembler que permitia desenvolver programas com a utilização de representações simbólicas das instruções de máquina. Em seguida, foram disponibilizadas as linguagens de alto nível e a programação

estruturada. Para acompanhar a complexidade dos problemas, foi criada, mais recentemente, a técnica de orientação a objetos que permite que um programa seja decomposto em unidades chamadas objetos.

Independentemente da metodologia e dos paradigmas utilizados, a lógica de programação é um requisito fundamental nos cursos de tecnologia, sendo um instrumento importante na estruturação de raciocínio lógico e formulação de algoritmos corretos. No entanto, muitas vezes a condução desse estudo é visto com preocupação pelos alunos, sendo que mesmo antes do início das atividades já é possível observar certa resistência perante o conteúdo.

Desta forma, é de extrema importância ressaltar que apropriação ou não dos conceitos iniciais de programação tem uma relação diretamente relacionada com o desempenho dos alunos no decorrer de todo o curso, visto que as disciplinas avançadas dependem fortemente destes conceitos.

Inicialmente, nossa primeira abordagem junto ao aluno deve ser motivacional. O professor desperta a curiosidade no aluno, explica como funciona internamente o caixa eletrônico do banco, por exemplo: Você já parou para pensar, o porquê de todo o processo que você realiza durante o saque de R\$100,00 no caixa eletrônico? E através deste pequeno exemplo, alguns conceitos de programação são repassados aos alunos, como o armazenamento (banco de dados), variáveis e o próprio funcionamento algorítmico do computador (atividades sequenciais).

Em seguida, pode-se propor aos alunos problemas de lógica, os quais devem ser resolvidos sem o auxílio do professor. O aluno deverá ao final da disciplina ser capaz de raciocinar sem o modelo do professor. O raciocínio a partir de seu próprio conhecimento é o que se propõe ao aluno. Em um primeiro exercício não é dada nenhuma explicação, no primeiro momento os alunos ficam revoltados, “o professor não explica”, “eu não estou entendendo nada”, “eu vim para aprender”, porém quando alguns colegas começam a solucionar os exercícios, inicia-se o processo de integração e trocas, e o aluno percebe que cada colega encontrou a solução de forma diferente, e ele mesmo começa o seu processo de solução.

Em grande parte dos cursos técnicos a implementação de algoritmos localiza-se tão somente ao final dos conteúdos. Como um algoritmo é algo abstrato, é importante que a partir do conteúdo de estrutura sequencial, inicie-se o processo de concretização, pois o aluno tem dificuldade em compreender o algoritmo, a menos que o professor o traduza em uma linguagem de programação e ele possa então consolidar/concretizar o seu funcionamento. Aliar a teoria com a prática no momento em que se ensinam as estruturas de construção dos algoritmos facilita a sua compreensão pelos alunos.

### 2.2.1 Problemas no Ensino e Aprendizagem de Lógica de Programação

Vários são os fatores dos insucessos em disciplinas de Lógica de Programação, o baixo nível de abstração, a falta de competências na resolução de problemas, a inadequação dos métodos pedagógicos aos estilos de aprendizagem e, principalmente, o de que as linguagens de programação possuem sintaxes adequadas a profissionais mais experientes e não a iniciantes inexperientes.

Para HOC (1987), “problema é a representação que um sistema cognitivo humano ou artificial constrói a partir de uma tarefa, antes mesmo de determinar o procedimento para resolvê-la”. A atividade de solucionar um problema resume-se na compreensão da tarefa e a estratégia de resolução a fim de chegar à solução. É um complicador afirmar, *a priori*, se uma

determinada situação é um problema para um aluno ou não, pois o que para uns é uma problemática, para outros é uma atividade rotineira ou um desafio.

A palavra “problema” está intrinsicamente associada às disciplinas de cálculos matemáticos. Essa palavra em diversas situações é apresentada de forma errônea, pois se refere a uma série de exercícios que necessitam da aplicação rotineira de um procedimento já conhecido. Um problema real é definido como uma situação que é nova ao aluno que vai resolvê-la, enquanto que o exercício é o uso de alguma habilidade e/ou conhecimento mecânico matemático já conhecido. A resolução de um exercício envolve mera aplicação e o problema, basicamente, envolve a criação de procedimentos para chegar à solução. Dentre outros problemas, alguns bastantes objetivos, podemos identificar a dificuldade de interpretação do próprio exercício (FALKEMBACH et al., 2003), muito antes da dificuldade de interpretação de algum tipo de representação, além da dificuldade de interpretar enunciados.

Alunos com dificuldades advindas de disciplinas relacionadas a cálculos matemáticos tendem a não acompanhar o conteúdo em sua plenitude, o que pode ocasionar perda de assimilação e, igualmente importante, de interesse, fazendo com que os mesmos passem a ver a disciplina como algo aversivo e preocupando-se essencialmente em conseguir a média mínima para seguir adiante no curso. Como se pode administrar e até mesmo ministrar atendimento individual, respeitando as individualidades e, ao mesmo tempo, buscar trabalho cooperativo?

Observa-se que são inúmeras as possibilidades da origem das dificuldades no ensino da Lógica de Programação, como destaca RAABE e SILVA (2005), seja pela exigência lógico-matemático predominante na disciplina, ou até mesmo pela dificuldade, por parte do professor e o ritmo de aprendizagem de cada aluno. No que diz respeito ao ritmo de aprendizagem, é importante destacar que o andamento da disciplina não necessariamente é conduzido no ritmo de assimilação de cada aluno, tendo em vista que um professor precisa atender uma considerável quantidade de alunos ao mesmo tempo. Este fator pode acarretar dificuldades na apropriação dos conceitos fundamentais, podendo ocorrer como consequência o desinteresse pelo conteúdo ministrado ou, até mesmo, certa aversão à disciplina.

Sob este panorama, é de fundamental importância que se aprimore a qualidade do processo de ensino e aprendizagem nos cursos técnicos relacionados à tecnologia. Um dos maiores gargalos nestes cursos é o alto índice de reprovação e/ou evasão, desde o início do curso, em disciplinas relacionadas à Lógica de Programação. Este é um dos principais fatores que contribuem para que a quantidade de egressos concluintes seja consideravelmente inferior à quantidade de alunos ingressantes. Nesse cenário, estão incluídos os alunos de primeiro ano do Curso Técnico em Desenvolvimento de Sistemas ofertado pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) - Campus Canoas, conforme se pode observar na Tabela 1:

**Tabela 1: Índices de desempenho relacionados à disciplina de Lógica de Programação no IFRS - Campus Canoas**

	2011	2012	2013	2014	2015
<b>Aprovados</b>	32	20	26	24	26
<b>Reprovados</b>	7	7	5	11	8
<b>Infrequentes</b>	1	1	3	5	0
<b>% aprov</b>	82,1%	74,1%	83,9%	68,6%	76,5%
<b>% reprov</b>	17,9%	25,9%	16,1%	31,4%	23,5%

**Fonte: Elaborado pelo autor, Canoas 2016**

Em um cenário mais amplo junto aos cursos técnicos ofertados pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) – Campus Canoas, pode-se observar que as disciplinas de *Lógica de Programação* possuem um dos maiores índices de reprovação. Segundo ROCHA et al. (2010), as disciplinas de lógica, algoritmos e programação iniciam com uma média de 50 alunos, e em poucos meses constata-se que a taxa de reprovação (ou desistência) chega a 60%. Segundo GALHARDO (2006), os altos índices de reprovação estão relacionados diretamente a forma como ocorre o processo de ensino-aprendizagem. RAABE e SILVA (2005) identificaram três tipos de aspectos que contribuem para a dificuldade de aprendizagem de algoritmos:

- Problemas de natureza didática: envolvem a diversidade de cultura e de experiência dos alunos ingressantes, a excessiva quantidade de alunos por turma, o que inviabiliza um atendimento e avaliação individual, a dificuldade de expressar e compreender a lógica desenvolvida e a ausência de materiais de referência de qualidade aos alunos. Além disso, há problemas particulares, que dizem respeito à escolha equivocada do curso.
- Problemas de natureza cognitiva: aspectos que normalmente estão atrelados a vivência precedente ao ingresso a um curso técnico. São encontrados em alunos cujos estudos foram interrompidos há algum tempo ou cujo ensino fundamental não os permitiu desenvolver adequadamente as habilidades cognitivas.
- Problemas de natureza afetiva: um aluno pode estar convicto da sua escolha quanto ao curso, ter aptidão para resolução de problemas que envolvem raciocínio lógico e, mesmo assim, não conseguir obter o sucesso na disciplina. Nesses casos, o foco deve ser direcionado a possíveis causas afetivas que interferem no desenvolvimento.

Segundo ALVES ([s.d.]), a motivação intrínseca, para nós, em seus aspectos relacionados ao processo de aprendizagem, acontece quando o aprendiz quer aprender o que é proposto, percebe a relevância da atividade proposta e desfruta do processo investigando, explorando e se engajando por conta própria, independente da existência de algum tipo de recompensa.

Na maioria das vezes, os alunos iniciantes nas disciplinas de Lógica de Programação, estão frequentemente habituados a disciplinas nas quais é possível ser bem sucedido através

de abordagens de estudo baseadas em leituras sucessivas, memorização de fórmulas e certa mecanização de procedimentos. Assistir às aulas e estudar em um livro não é o suficiente neste caso. Programar exige um envolvimento fora da sala de aula, através da realização de exercícios e isso é necessário para o acompanhamento das dificuldades da disciplina.

### 2.3 Gamificação

A utilização das diversas tecnologias em sala de aula proporciona à educação, em diversos aspectos, um ambiente mais interativo, porém, um antigo problema relacionado à motivação e ao engajamento dos estudantes ainda persiste nestes ambientes. Em um sistema normal de ensino, a desmotivação dos alunos é resultado de diferentes aspectos, tais como a falta de compreensão do plano de ensino, a didática do professor, a dificuldade em visualizar os benefícios de ir à escola, a falta de confiança, *bullying*, medo, cansaço, etc. (ATKIN, 2012).

Diversas iniciativas vêm sendo estudadas para melhorar a motivação e o engajamento do estudante, um fenômeno emergente que vem surgindo derivado diretamente da popularização dos jogos digitais e de sua capacidade intrínseca de motivar a ação, resolver problemas e potencializar a aprendizagem na mais diversas áreas, é o intitulado gamificação (WERBACH; HUNTER, 2012), que consiste na utilização de elementos dos jogos (mecânicas, estratégias, pensamentos) fora do contexto dos jogos, com a finalidade de motivar as pessoas à ação, auxiliar na solução de problemas e promover aprendizagens (KAPP, 2012).

Inicialmente criada como um método a ser aplicado em programas de marketing e aplicações para web, com a finalidade de motivar, engajar e fidelizar clientes e usuários (ZICHERMANN; CUNNINGHAM, 2011), a gamificação presume na utilização de elementos rotineiramente encontrados nos jogos, como narrativa, *feedback*, recompensas, conflito, cooperação, competição, objetivos e regras claras, níveis, tentativa e erro, diversão, interação, interatividade, entre outros, em outras atividades que não diretamente relacionadas aos jogos, com o intuito de obter-se um grau de envolvimento e motivação normalmente encontramos em jogadores.

Em gamificação o objetivo é conseguir vislumbrar um determinado problema ou contexto e pensar em soluções a partir do ponto de vista de um *game designer*<sup>2</sup>, já que esse profissional geralmente possui uma capacidade de produzir experiências que concentram a energia e o foco na resolução problemas em mundos virtuais (MCGONIGAL, 2011).

Não necessariamente a gamificação implica em se criar um jogo que aborde o problema, recriando a situação dentro de um mundo virtual, mas sim, em usar as mesmas estratégias, métodos e estratégias utilizados para resolver aqueles problemas nos mundos virtuais em situações do mundo real.

Para posicioná-la e destacá-la de outros contextos a gamificação é apresentada entre dois eixos, conforme Figura 1. O eixo horizontal vislumbra a ideia de um jogo completo até os seus elementos e o eixo vertical vai da ludificação (brincadeira) até o jogo propriamente dito. Desta maneira, a gamificação alicerça-se no uso de elementos dos jogos, sem que o resultado final seja um jogo completo e também se diferencia da construção lúdica na medida em que este apenas propõe um aspecto mais libertador, quanto ao contexto em que está inserido. Ou seja, nesta concepção, vislumbrar um problema de forma lúdica não implica em

---

<sup>2</sup> Pessoa que projeta jogos para computador, videogames ou dispositivos móveis.

contemplar objetivos e seguir um método mais preciso, que é a que a gamificação propõe (DETERDING et al., 2011).

**Figura 1: Contextualização da gamificação**



**Fonte: (DETERDING et al., 2011)**

Neste cenário de jogo, do qual a gamificação deriva, temos invariavelmente a implicação em concebê-lo como um software, composto por elementos interconectados que, ao agruparem-se como um todo, resulta em um fenômeno que é maior do que a soma de suas partes. Desta maneira, pode-se utilizar desde um número pequeno de elementos, até uma vasta quantidade, fazendo com que o jogo ao final produza uma experiência próxima a de um jogo completo.

Podemos elaborar softwares gamificados baseados tão somente em pontuação, medalhas e *rankings*<sup>3</sup> (PBL – Points, Badges and Leaderboards), que nada mais são do que as mecânicas mais básicas de um jogo, com o intuito único de prover alterações comportamentais dos indivíduos através de recompensas extrínsecas, semelhantes às ideias da economia comportamental (LADLEY, 2011), ou pode-se apostar na construção de uma experiência mais enfática e que vá muito mais além do que as mecânicas básicas dos jogos oferecem, motivando intrinsecamente os jogadores a desempenharem seus papéis da melhor forma possível dentro do contexto em que se encontram (WERBACH; HUNTER, 2012).

Segundo DECI e RYAN (2002), a partir da teoria da autodeterminação, há três necessidades básicas da motivação intrínseca: competência, autonomia e a sensação de se pertencer a uma comunidade. Trabalhando com pontuações e níveis de experiência, o usuário é instigado a buscar as atividades a fim de cumprir metas e atingir objetivos.

Em um cenário mais amplo, a gamificação mostra-se um fenômeno resultante de muitas potencialidades e aplicações em diversas áreas do conhecimento, pois como seu jargão e métodos gamificados são bastante populares e eficazes na resolução de problemáticas, a

<sup>3</sup> Classificação ordenada de acordo com critérios determinados.



mesma foi naturalmente inserida pelas atuais gerações em seu crescimento interativo com este tipo de entretenimento. Desta maneira, a gamificação se justifica a partir de uma perspectiva sociocultural, encontrando na educação formal uma área bastante fértil para a sua aplicação, pois lá ela encontra os indivíduos que carregam consigo muitas aprendizagens advindas das interações com o mundo dos jogos, encontra também uma vasta área que anseia por novas estratégias para dar conta de indivíduos cada vez mais inseridos no contexto das tecnologias digitais e mostram-se desinteressados aos métodos passivos de ensino e aprendizagem amplamente utilizado na maioria dos ambientes escolares.

### 2.3.1 A Gamificação no processo de ensino aprendizagem

Para GLOVER (2003), assim como no processo de aprendizagem de um aluno, em um ambiente gamificado ou em um jogo é necessário rastrear o progresso de um “jogador”, porque sem isso, seria impossível identificar as tarefas e metas que ainda faltam para o jogador chegar em uma condição de vitória (ou no caso de ambientes educacionais - aprendizagem).

Desta forma, para aplicarmos a gamificação como método transformador nos processos de ensino e aprendizagem através da utilização de estratégias e pensamentos dos jogos, no firme propósito de aproximarmos estes processos dos indivíduos da geração *gamer*, algumas ponderações iniciais devem ser adotadas.

Diversos encaminhamentos para adotar esta perspectiva são descritos por GLOVER (2003) e resumidos a seguir.

- **Disponibilizar diferentes formas de resolução:** assim como na grande maioria dos jogos, não há um único caminho que conduz ao sucesso. Proporcionar diferentes maneiras de se alcançar a resolução de um problema, incorporar diversas características pessoais de aprendizagem, contribui na experiência educativa de cada um;
- **Incluir curtos ciclos de *feedback*:** durante um jogo, os jogadores são sempre capazes de visualizar o efeito de suas ações em tempo real. Já em nossas instituições de ensino acontece basicamente o inverso e os alunos só conseguem visualizar seus resultados depois de certo tempo, muito maior do que aquele que estão acostumados durante um jogo. Tornar mais rápido este processo de *feedback* estimula a procura de novos caminhos para atingir os objetivos, bem como o redirecionamento de uma estratégia, caso ela não esteja apresentando os resultados esperados. O *feedback* ao usuário é fundamental para dar suporte à sua tomada de decisão frente a situações que podem causar uma falha e para um melhor aproveitamento do usuário dentro do ambiente, ou ainda para aumentar seu nível de engajamento (FADEL et. al, 2014).
- **Aumentar dificuldade dos níveis conforme a habilidade dos alunos:** em jogos de grande sucesso os jogadores sempre encontram desafios no limite de suas habilidades. Proporcionar diferentes níveis de dificuldade para os desafios propostos pode auxiliar na construção de um senso de crescimento e avanço pessoal, proporcionando que cada jogador possa seguir seu próprio ritmo de aprendizagem.

Existem três tipos diferentes de níveis que são descritos por (KAPP, 2012):

- **Níveis de jogo:** Espera-se atender três objetivos principais. O primeiro é manter a percepção de que há um progresso na história do

jogo/sistema, proporcionando o engajamento do usuário. Fazendo essa relação com os ambientes educacionais, espera-se que os estudantes percebam seu progresso no processo de ensino-aprendizagem. O segundo é focado no desenvolvimento das habilidades do usuário e visa que a cada nível avançado, habilidades aprendidas sejam reforçadas ou novas sejam aprendidas, gerando aprendizagem gradativa. Por último, os níveis servirão como motivação para os usuários, pois, ao avançar de nível, ele vai querer alcançar os novos objetivos do nível atual e avançar para níveis mais difíceis;

- **Níveis de dificuldade:** O desenvolvimento de um jogo que sempre oferece a mesma dificuldade pode criar um problema. Se um jogo/sistema for muito fácil, afastará pessoas que gostam de ser desafiados e, se for muito difícil, afastará pessoas que preferam jogos que aumentam gradativamente a dificuldade (KAPP, 2012). Para contornar esta situação, uma prática comum dos *designers* é o desenvolvimento de três níveis de dificuldade; fácil - para iniciantes com desafios mais simples; médio - onde a maior parte dos usuários se encontra e proporciona um sistema com dificuldade e desafios equilibrados; difícil - mais indicado para usuários experientes, pois possuem desafios com maior complexidade.
- **Níveis de jogador:** Conhecidos como níveis de experiência, demonstram o progresso do jogador/usuário no sistema e são atribuídos ao usuário de acordo com a realização de tarefas, missões, desafios e por sua fidelização com o sistema. Geralmente esses níveis são acumulados e utilizados para alcançar habilidades especiais, adquirir itens novos ou como moeda dentro do ambiente. Os usuários valorizam atingir níveis mais altos e obter mais pontos, pois cada novo nível é mais difícil que o anterior e isso gera sensação de domínio e de realização (KAPP, 2012).
- **Dividir tarefas mais complexas em níveis menores:** na ampla maioria dos jogos, os objetivos maiores são divididos em uma série de objetivos menores e de mais fácil superação. Resultando em uma construção gradual de conhecimento, pois ao observar as partes do problema e de que modo elas se relacionam, proporciona-se uma maior motivação e preparo na superação do desafio maior;
- **Incluir a falha como parte do processo de aprendizagem:** em um cenário real nenhum jogador espera interagir com um jogo sem se deparar com uma dificuldade. Na aprendizagem sistematizada, a dificuldade normalmente não é bem tolerada. Incluí-la e aceitá-la como parte do processo de aprendizagem é estimular a reflexão de seus motivos;
- **Incorporar narrativas desafiadoras como contexto dos objetivos:** nos jogos temos um motivo para as ações dos personagens, um roteiro que justifica o porquê de estarem fazendo aquilo. Muitas aprendizagens ocorrem fora de contexto e os estudantes têm dificuldade em entender os motivos de estarem se dedicando a aquilo, de relacionar a parte com o todo. Construir um contexto para a aprendizagem pode fornecer um bom motivo para os estudantes empenharem suas energias em aprender;

- **Promover a competição e a colaboração:** dois aspectos essenciais nos jogos são a competição e a colaboração e eles não precisam ser mutuamente exclusivos. Esses contextos podem ocorrer juntos com a narrativa podendo haver competição entre grupos, potencializando a interação podendo fornecer mais um contexto para os objetivos. *Rankings* possuem em seu propósito principal a comparação entre os jogadores envolvidos. Serve como uma forma de visualizar a progressão dos usuários dentro do ambiente e gera um senso de competição entre eles. (KAPP, 2012) descreve dois tipos de *Rankings*.
  - **Ranking motivador:** Com o advento das redes sociais, os *rankings* passaram a contar com uma de suas principais características que é a promoção do incentivo social. Assim, o usuário geralmente será colocado em alguma posição do *ranking*, conforme seu desempenho, tendo acesso às informações dos usuários mais próximos (melhores e piores que ele) ou ainda a colocação de seus amigos;
  - **Ranking infinito:** Dependendo da quantidade de usuários de um ambiente, não há uma forma de exibi-los todos em um *ranking*, além disso, há situações em que o usuário pode ter feito uma pontuação muito alta ou muito baixa, fazendo com que ele fique preso em uma posição. Desta forma, o objetivo principal desse tipo de *ranking* é fazer com que o usuário não fique preso em uma posição e/ou que venha a ser ultrapassado por muitos outros usuários em pouco tempo. Para alcançar este objetivo, o *ranking* pode ser dividido em categorias diferentes que farão com que o usuário intercale sua posição.
- **Promover sempre a diversão:** a aprendizagem deve ser prazerosa, pois em nossos primeiros anos escolares existe um consenso que aprender deve ser divertido e, após isso, a seriedade deve prevalecer. Bons jogos de uma maneira geral são divertidos por uma série de motivos e também boas ferramentas para a aprendizagem. Pensar esse aspecto durante o ensino e aprendizagem pode melhorar a experiência que os estudantes têm dentro dos ambientes escolares, o que acaba por potencializar a aprendizagem como um todo.

O conceito de lúdico é muito próximo ao conceito de gamificação, isso porque a palavra “lúdico” deriva da palavra “*ludus*”, jogo em Latim. Para CRESPO e GIACOMINI (2011) a "ludicidade" vai muito além dos jogos, envolvendo qualquer tipo de atividade prazerosa.

É crescente o número de autores que defendem a ideia de que atividades lúdicas devem ser utilizadas com maior frequência dentro dos ambientes escolares, com alunos de todas as idades. Para DINELLO (2004), este tipo de atividade estimula os alunos a concentrarem-se naquilo que estão aprendendo, transformando impulsos negativos em relações criativas, ampliando sua capacidade criativa de socialização e estimulando até mesmo que estes se tornem mais independentes e autodidatas.

## 2.4 Linguagem de programação LUA

Neste item são descritos alguns aspectos da linguagem de programação Lua, que corroboraram para que esta fosse escolhida para uso neste projeto.

Segundo CELES, FIGUEIREDO e IERUSALIMSKY (2003):

“... Lua é uma linguagem de programação poderosa e leve, projetada para estender aplicações. Isso quer dizer que ela foi projetada para ser inserida a programas maiores que precisem ler e executar programas escritos pelos usuários. Lua é uma linguagem, com sintaxe semelhante à de Pascal, mas com construções modernas, como funções anônimas, inspiradas no paradigma funcional, e poderosos construtores de dados. Isso faz com que Lua seja uma linguagem de grande expressão com uma sintaxe familiar.”

Para IERUSALIMSCHY e MOURA (2016), atualmente a linguagem Lua é utilizada no desenvolvimento de diversos tipos de aplicação que vão desde sistemas embarcados, como o middleware Ginga<sup>4</sup> para TV digital, até jogos, como o Angry Birds<sup>5</sup>, por exemplo.

Ainda segundo CELES, FIGUEIREDO e IERUSALIMSCHY (2003):

“Desta forma a adoção de uma mesma linguagem para varias tarefas na aplicação permitirá um crescimento importante na produtividade global, pois não se faz necessário definir, documentar ou manter diversos formatos diferentes. Para o usuário da aplicação, isso se traduz no reaproveitamento automático e inconsciente dos conceitos aprendidos para realizar as várias tarefas, encorajando a exploração.”

“... Mesmo uma linguagem simples como essa já dá a uma grande flexibilidade à aplicação, pois permite ao usuário controlar a aplicação externamente, bastando editar um arquivo texto. ...”

“A linguagem Lua tem sido amplamente utilizada no desenvolvimento de jogos. A Lucasarts, por exemplo, usou a versão 3.1 de Lua para desenvolver os títulos “*Grim Fandango*”<sup>6</sup> e “*Scape from Monkey Island*”<sup>7</sup>. A versão 3.1 de Lua foi por eles modificada para tratar cotinas.”

Para além dos motivos acima apresentados e fundamentados, a seguir são descritos dois dos principais motivos pelo qual Lua foi à linguagem escolhida para a produção deste trabalho. O primeiro foi por esta possuir uma sintaxe bastante simples, lembrando muito o Pascal<sup>8</sup>, detalhe que pode ser relevante para quem nunca teve contato com alguma outra linguagem de programação.

---

<sup>4</sup> **Ginga®** é o middleware do sistema nipo-brasileiro (ISDB-TB) e da Recomendação ITU-T para TV Digital em serviços IPTV. Ginga foi implementado através de um conjunto de padrões tecnológicos e inovações brasileiras que utilizam as mais avançadas especificações para middleware.

<sup>5</sup> **Angry Birds** é uma série de jogos desenvolvidos pela finlandesa *Rovio Entertainment*. Lançado para Apple iOS em Dezembro de 2009.

<sup>6</sup> Jogo de computador do gênero aventura feito pela LucasArts em 1998. É o primeiro jogo de computador feito em 3D pela LucasArts.

<sup>7</sup> Jogo de aventura para computador desenvolvido e lançado pela LucasArts em 2000. É o quarto jogo da série *Monkey Island*.

<sup>8</sup> Linguagem de programação estruturada, que recebeu este nome em homenagem ao matemático e físico Blaise Pascal. Criada em 1970 pelo suíço Niklaus Wirth, tendo em mente encorajar o uso de código estruturado.

Outro fator motivacional na escolha de Lua foi sua importância global, uma vez que há alguns anos a linguagem vem constantemente aparecendo em listas das linguagens de programação mais utilizadas ao redor do mundo, segundo TIOBE (2016).

## 2.5 Mineração de dados educacionais

Para BRUSILOVSKY e PEYLO (2003), nos últimos anos os sistemas de ensino baseados em web cresceram exponencialmente, estimulado pelo fato que tanto o professor quanto os alunos não têm a obrigação de estarem em um local específico e que essa forma de ensino independe de plataforma computacional ou hardware específico.

Segundo MOSTOW et al. (2005), esses tipos de sistemas educacionais podem acumular grande quantidade de informação que são extremamente valiosas para a análise do comportamento dos alunos. Informações tais como a leitura, a escrita, exercícios, provas e até a comunicação com seus pares podem ser gravadas em logs e facilmente acessadas.

Para ZORRILLA et al. (2005), esse novo conhecimento pode ser explorado não só pelos alunos, mas também pelos professores, para que estes possam dessa forma, orientar de maneira eficiente, levando em consideração o ponto de vista de um conjunto ou de um determinado aluno.

Já para GARCÍA et al. (2011) as técnicas de mineração de dados educacionais (MDE) convertem os dados brutos de sistemas educacionais em informação útil que pode ser utilizadas por desenvolvedores de softwares educacionais, professores, pesquisadores educacionais etc. Este processo não difere muito de outras áreas de aplicação da mineração de dados porque baseia-se nos mesmos passos do processo de mineração de dados em geral, conforme pode ser observado na Figura 2. Esta abordagem tem permitido melhorar os processos de ensino-aprendizagem e os mecanismos de gestão acadêmica e pedagógica das instituições de ensino”.

**Figura 2: Etapas da Mineração de dados educacionais**



**Fonte: Adaptado de GARCÍA et al., 2011**

Várias abordagens utilizando recursos de MDR podem ser encontradas, para fins diversos. A seguir são comentadas algumas destas. Em SHEN et al. (2003), os estudantes são agrupados de acordo com suas preferências com o intuito de melhor adaptar os sistemas de ensino de acordo com suas necessidades. Ainda em SHEN et al. (2003), são construídos modelos representativos de cada grupo, que são por fim utilizados para identificar as melhores práticas de ensino e sugestão de material de acordo com as preferências dos membros de cada grupo.

Outra abordagem de interesse é a técnica de agrupamento que segundo COSTA et al., (2012) tem como objetivo dividir o conjunto de dados em grupos, de forma que os objetos contidos nos dados fiquem agrupados naturalmente de acordo com a semelhança entre eles.

Os algoritmos de agrupamento são técnicas de aprendizado não supervisionado, logo os grupos ou categorias, e até mesmo suas quantidades, não são conhecidas inicialmente.

Para BAKER; YACEF, (2009), a MDE tem emergido como uma relevante fonte de pesquisa nos últimos anos em diversas áreas, por exemplo: ciência da computação, educação, psicometria, estatística, sistemas tutores inteligentes, e-learning entre outros, em que, se analisam grandes conjuntos de dados, a fim de resolver as questões de investigação educacional.

Ainda para BAKER; YACEF, (2009), a EDM é a área de pesquisa que tem como principal foco o desenvolvimento de métodos para explorar bases de dados coletadas em ambientes educacionais. Assim, é possível compreender de forma mais eficaz e adequada os alunos, como eles aprendem, o contexto na qual a aprendizagem ocorre, além de outros fatores que a influenciam.

Segundo (ROMERO et al., 2008), os dados gerados pelos alunos e instrutores em ambientes *e-learning* podem fornecer rápidas e importantes compreensões acerca do desempenho, da motivação e do nível de participação dos alunos no curso. Essas compreensões podem sugerir mudanças no curso, intervenções significativas na metodologia ou mesmo um contato individual com alunos desmotivados ou com baixa interação.

KAMPFF et al., (2014), explicam que a partir do desenvolvimento da EDM é possível desenvolver mecanismos e ferramentas educacionais mais eficientes, modelos para identificar alunos com dificuldades, aperfeiçoar os materiais didáticos e desenvolver métodos pedagógicos mais eficazes. A partir da análise dos dados referentes às interações dos alunos, diferentes informações podem ser obtidas, tais como perfis e padrões relevantes ao planejamento futuro das atividades educacionais.

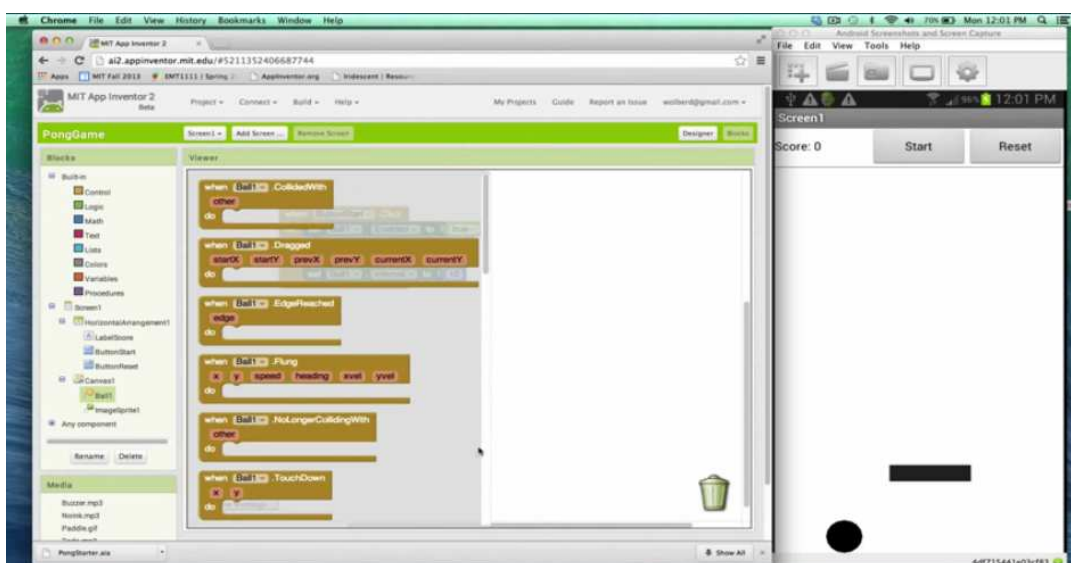
### 3 TRABALHOS RELACIONADOS

Neste capítulo estão descritas análises de jogos que possuem relação com o trabalho proposto. Inicialmente foram escolhidos jogos relacionados aos ambientes de aprendizagem para o ensino da lógica de programação que aplicam algum tipo de gamificação. Em seguida, buscou-se identificar os melhores jogos que aplicam algum tipo de ação relacionada à habilidade de melhoria no ensino/aprendizagem do aluno através do monitoramento do seu conhecimento. Foram consultados trabalhos em um conjunto de conferências, periódicos e bibliotecas<sup>9</sup>, relacionados à área da Informática na Educação. A seguir são descritos, de forma integrada, os principais aspectos tanto dos jogos analisados como de artigos científicos que descrevem elementos complementares destes mesmos trabalhos.

#### 3.1 APP Inventor

O **APP Inventor** é um aplicativo de código aberto desenvolvido pela Google<sup>10</sup> cujo propósito é a criação de aplicativos através da programação em blocos para dispositivos móveis que executem sistema operacional Android. Utiliza programação em blocos para a construção dos programas. Na figura 3 pode ser observado o ambiente de desenvolvimento com um exemplo de trecho de código.

**Figura 3: Tela de desenvolvimento da plataforma App Inventor**



Fonte: <https://xsrv.mm.cs.sunysb.edu/c4g/lectures/AppInventor/AIQuiz/AppInventorQuiz.html>

<sup>9</sup> Research Gate: Acessível em <https://www.researchgate.net>; WEI – Workshop sobre Educação em Computação: Acessível em: <http://www.csbc2016.com.br/#!/wei2016/hmrvt>; ACM – Digital Library: Acessível em <http://dl.acm.org/>; IEEE: Acessível em <https://www.ieee.org/index.html>; Capes: Acessível em <http://www.capes.gov.br/>; Cinted – UFRGS: Acessível em <http://www.cinted.ufrgs.br/bibliotecavirtual.html>; CBIE – Congresso Brasileiro de Informática na Educação: Acessível em <http://www.cbie2016.facom.ufu.br/pt/>; Scielo: Acessível em <http://www.scielo.org/php/index.php>.

<sup>10</sup> **Google** é uma empresa multinacional americana de serviços online e software. O **Google** hospeda e desenvolve uma série de serviços e produtos baseados na internet.

Apesar do site do fabricante informar que não há a necessidade de um conhecimento prévio em lógica de programação, na prática, e após a análise, o mesmo mostrou-se bastante complexo. A falta de intuitividade e motivação perante o jogador é outro aspecto importante a ser ressaltado, sendo que esta plataforma se apresenta mais adequada para desenvolvedores experientes do que um ambiente de aprendizagem a usuários iniciantes. No quesito tutorial o mesmo se mostra muito básico, apresentando ao jogador apenas um botão de ajuda.

O artigo de DUDA (2014) descreve a utilização do aplicativo **APP Inventor** em um projeto de extensão intitulado “Desenvolvimento de aplicativos para dispositivos com sistema operacional Android com uso do App Inventor”. Este projeto foi executado no Campus Irati do Instituto Federal do Paraná (IFPR), que visou explorar as potencialidades do aplicativo no desenvolvimento de atividades e materiais que eventualmente pudessem ser utilizados por docentes da área de matemática em suas aulas. As atividades do projeto objetivaram a abordagem do uso da representação algébrica na solução de problemas relacionados aos temas estudados em sala de aula, com o intuito de desenvolver o pensamento algébrico dos discentes envolvidos buscando também a autonomia dos mesmos com relação à busca de informações auxiliares no processo de elaboração dos aplicativos. Cabe enfatizar que para estruturar os aplicativos no App Inventor não há a necessidade de prévio conhecimento sobre linguagem de programação, pois as funções dos elementos necessários para os aplicativos são pré-dispostos no designer de aplicativos na forma de blocos lógicos, que podem ser justapostos para gerar as funções necessárias. Ao final, os autores concluíram que embora ainda não tenha sido viabilizada a aplicação dessa dinâmica de ensino na sala de aula convencional, observou-se que o roteiro de estudos e demais produtos decorrentes do processo de elaboração de um aplicativo podem ser utilizados como objetos auxiliares para a avaliação em matemática, possibilitando a análise qualitativa do processo de ensino e aprendizagem no que diz respeito à autonomia do discente em buscar diferentes fontes e formas de solucionar ou descrever um problema, bem como estruturar logicamente o pensamento.

### 3.2 URI Online Judge

O **URI Online Judge** é um projeto que está sendo desenvolvido pelo Departamento de Ciência da Computação da URI University. Seu principal objetivo é proporcionar a prática de programação para linguagens C++, Java ou Python. Na figura 4 observa-se a tela inicial do ambiente de desenvolvimento.

Conforme informação fornecida pelo próprio site, há sim a necessidade de um conhecimento prévio em lógica de programação. Diferente de outros sites que propõem esta mesma temática, o URI possui uma interface muito clara e funcional, tornando-se assim um ambiente muito agradável aos olhos do usuário. Entretanto, observa-se a falta de um tutorial explícito, o que, por vezes, torna-se um complicador. Seu método de ensino é desenvolvido através de divulgação de exercícios.

O artigo de SELIVON; BEZ e TONIN (2015) descreve o uso do projeto **URI Online Judge** visando desenvolver funcionalidades que ofereceram uma alternativa ao método tradicional de ensino de Algoritmos e Programação, como apoio as aulas de Algoritmos e Estruturas de Dados, através de problemas que incluíram a prática de conceitos específicos utilizados nessas disciplinas, importantes para uma melhor compreensão dos estudantes. Uma característica importante presente na ferramenta é a determinação do intervalo válido de datas para resolução de uma lista de exercícios. Assim, resoluções feitas previamente à data de início ou posteriores à data de fim da lista não são contabilizadas e nem exibidas ao professor.



Também é possível que o professor possa conferir, por exemplo, o histórico de submissões de cada estudante para cada problema.

**Figura 4: Tela de desenvolvimento da plataforma URI Online Judge**



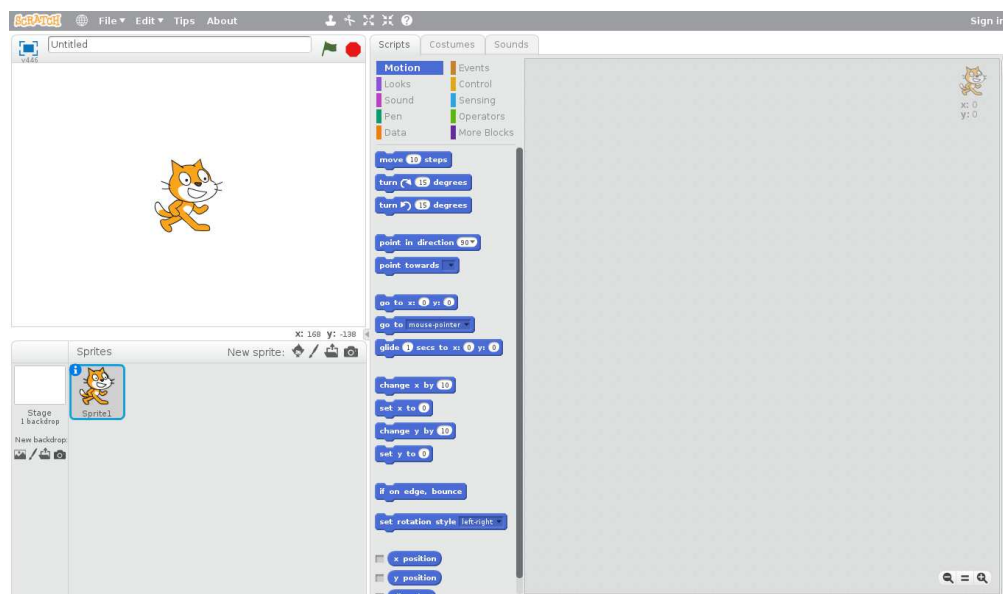
Fonte: <http://forum.sa-mp.com/showthread.php?t=530946>

Durante a criação das listas de exercício é possível, para o professor, estabelecer a linguagem que deve ser utilizada para a resolução, escrever instruções que julgar importantes para a orientação dos estudantes e delimitar o prazo de conclusão das atividades. Os exercícios de cada lista podem ser facilmente selecionados entre os 800 problemas disponíveis no acervo do URI Online Judge. Após a conclusão da criação das listas, o professor deve enviar, através do portal, um convite aos estudantes selecionados. Assim que o convite for aceito o professor pode visualizar de forma fácil e objetiva o avanço de cada estudante através de uma apresentação percentual e da marcação dos problemas já solucionados. Por fim, os autores concluem que o portal URI Online Judge constitui-se uma ferramenta que proporciona facilidade e estímulo a prática de Algoritmos e Linguagens de Programação, isto ocorre porque, além de auxiliar no processo de planejamento e correção de exercícios, o professor pode acompanhar a evolução dos estudantes tanto de forma individualizada quanto no contexto da turma ou equipe.

### 3.3 Scratch

O **Scratch** é uma plataforma de desenvolvimento que, diferentemente de outros ambientes, foi desenvolvida com o intuito de auxiliar aos usuários iniciantes nas áreas de tecnologias no aprendizado de conceitos matemáticos e computacionais, permitindo aos mesmos a criação de cenários animados, jogos e outros softwares interativos. Na figura 5 pode ser observado o ambiente de desenvolvimento com um exemplo de trecho de código.

**Figura 5: Tela de desenvolvimento da plataforma Scratch**



Fonte: <http://olimpiada-programacao-unipampa.blogspot.com.br/p/sobre-o-sc.html>

Diferente de outros aplicativos *online*, sua interface atrativa e colorida instiga o usuário a começar os primeiros passos rumo à aprendizagem da lógica de programação. Porém, isto não exige o usuário de ter pelo menos algum conhecimento básico em lógica de programação, pois esta não é a proposta da plataforma.

O artigo de SÁPIRAS; VECCHIA e MALTEMPI (2015) descreve a utilização da plataforma **Scratch** em oficinas oferecidas a alunos do 7º e 8º anos de uma escola do estado do Rio Grande do Sul. Estas oficinas foram divididas em duas etapas. A primeira etapa envolveu atividades que visavam a apresentação das funcionalidades do software Scratch para os alunos. Nesse primeiro momento, foram sugeridas construções dadas por Scratch Cards, que são cartões que contêm atividades pré-definidas utilizadas para que os alunos possam desenvolver pequenas sequências de comandos de forma independente e autônoma. A segunda etapa consistiu no desenvolvimento de um jogo ou animação que fosse do interesse dos próprios alunos, utilizando os recursos do Scratch. Assim, os autores consideraram que o desenvolvimento de habilidades relacionadas à simulação pode contribuir para a capacidade de lidar e interpretar as mídias digitais, envolvendo assim o conceito conhecido como literacia digital.

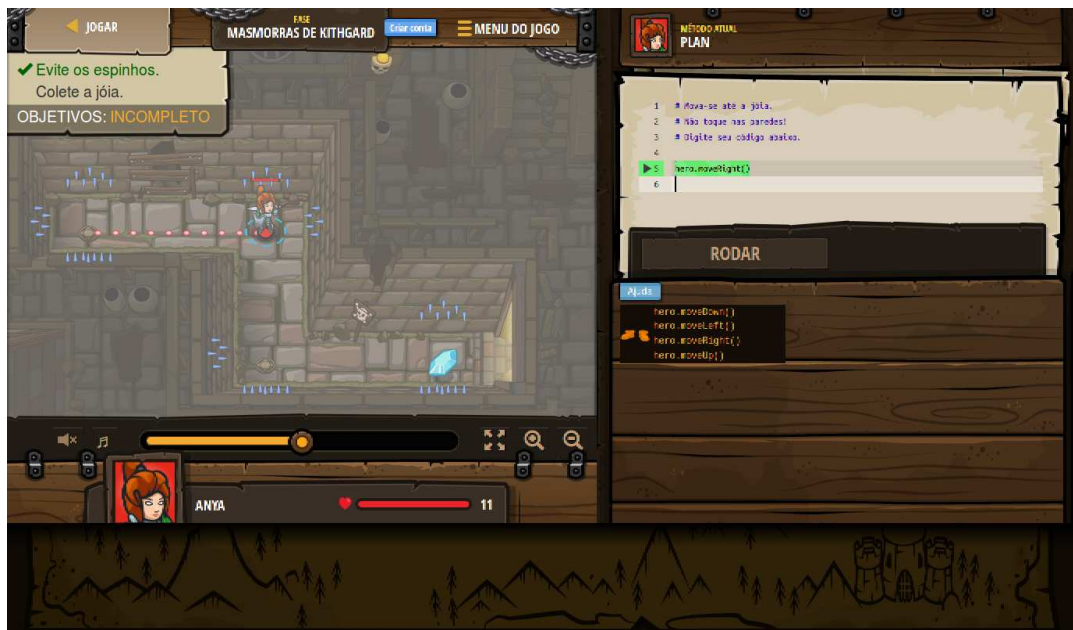
### 3.4 Code combat

Desenvolvido pela Code Combat Inc, o **Code Combat** é uma plataforma destinada a aprendizagem de diversas linguagens tais como Python, JavaScript, C++ entre outras. A plataforma conta com mais de 80 níveis diferentes, além de possuir interfaces diferenciadas

para estudantes e professores. Como outras plataformas estudadas, o Code Combat exige do usuário um conhecimento prévio em lógica de programação, bem como o conhecimento básico na linguagem escolhida pelo usuário.

Seu diferencial frente a maioria das plataformas estudadas é a apresentação de diversos agentes animados customizáveis pelo usuário, bem como diversos cenários e dinâmicas desafiadoras e motivadoras. Como as plataformas anteriormente estudadas, essa também não possui um tutorial explícito, o que a mesma possui é apenas uma fórum de discussão e dicas.

**Figura 6: Tela de desenvolvimento da plataforma Code Combat**



Fonte: <http://klaibert.com.br/educacao/code-combat-o-game-que-te-ensina-a-programar>

Para RAABE; ZANCHETT e VAHLICK (2015) o **Code Combat** mostra-se como uma plataforma gamificada que aborda conceitos de programação através de uma aventura medieval, onde o jogador controla heróis e, ao longo de sua jornada, vai coletando cristais e derrotando diversos inimigos, tais como ogros, bandidos, entre outros. O jogo disponibiliza uma IDE<sup>11</sup> própria para que o jogador implemente seus códigos. Esta IDE trabalha com 6 linguagens de programação, sendo elas: Python, JavaScript, CoffeeScript (Experimental), Clojure (Experimental), Lua e IO (Experimental). O jogador pode escolher entre 23 idiomas na página inicial do site do jogo. O jogo é dividido em 6 mundos, dentre eles 5 disponíveis com 165 fases gratuitas. Os dois últimos mundos ainda estão em desenvolvimento e abordam o que o jogo denomina de técnicas avançadas, entretanto não deixa claro quais são estas técnicas. O quinto mundo é acessível, enquanto o sexto não está disponível, e nem possui informações a seu respeito. A interface do jogo - ilustrada na figura 6 - mantém a IDE para digitar o código das soluções sempre no canto direito da tela. Os objetivos e dicas estão sempre escritos em azul no início do código, em forma de comentário. Abaixo da IDE, estão

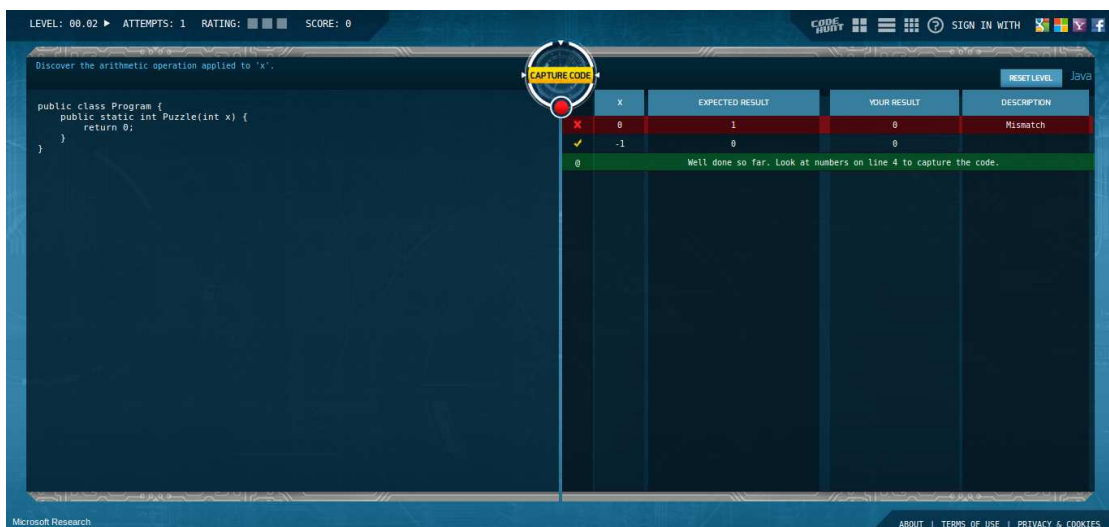
<sup>11</sup> *Integrated Development Environment* ou **Ambiente de Desenvolvimento Integrado** é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

dispostos todos os métodos que podem ser utilizados pelo jogador. Em caso de dúvida, basta o jogador passar o mouse sobre os métodos e uma caixa de diálogo se abrirá com uma explicação. O jogo possui um botão de ajuda, que leva o jogador a uma tela com tutoriais em vídeo sobre a fase em questão. Os autores concluem que apesar de utilizar a notação textual considerada por diversos autores como uma notação intimidadora à primeira vista, o **Code Combat** mostra-se uma ferramenta simples e envolvente no quesito aprendizado através de jogos.

### 3.5 Code hunt

Desenvolvido pela Microsoft Research, o **Code Hunt** apresenta-se como uma plataforma de ensino/aprendizagem das linguagens de programação Java e C#, exigindo de seus usuários um conhecimento intermediário em lógica e desenvolvimento das linguagens acima citadas. Na figura 7 observa-se o ambiente de desenvolvimento com um exemplo de trecho de código.

**Figura 7: Tela de desenvolvimento da plataforma Code Hunt**



Fonte: <http://www.up.edu.br/blogs/engenharia-da-computacao/wp-content/uploads/sites/6/2016/04/codehunt.png>

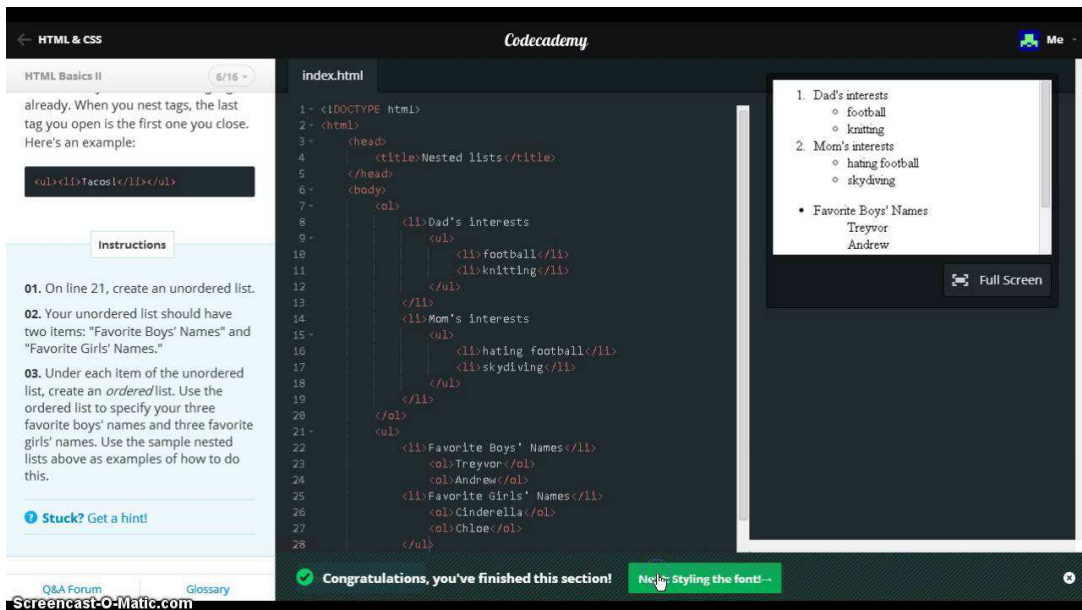
Diferente de seu concorrente direto, o **Code Combat**, sua plataforma mostra-se muito pobre, não possuindo um ambiente lúdico, forçando assim que o jogador seja diretamente imerso em algoritmos de Java e C# já nas primeiras fases do jogo. Seu foco não é o lúdico, apesar da temática investigativa. Seu diferencial é possuir o teste de mesa em seu tutorial, que ensina porém ao mesmo tempo torna-se complicado por ser todo em Inglês.

O artigo de TILLMANN et al. (2014) apresentam a ideia principal do ambiente de desenvolvimento intitulado **Code Hunt** através da existência de uma sequência de código secretamente escondida nos bastidores do jogo sendo o jogador incentivado a buscar esta sequência, de forma lúdica e interativa. Este processo se dá de forma tão simples que a maioria dos jogadores acaba por nem perceber que seu aprendizado se deu de forma natural. No momento de escrita deste artigo, os autores ainda estavam na fase da coleta de dados, posteriormente, e através de investigações empíricas, os mesmos pretendem justificar se a ferramenta apresenta-se como forma de melhoria no aprendizado de lógica de programação.

### 3.6 Codeacademy

Desenvolvido pela empresa de educação Codeacademy homônimo da própria plataforma de aprendizagem, o Codeacademy, mostra-se um ambiente mais sério, com cores sóbrias, sistema de ensino centrado e focando no ensino de diversas linguagens como java, php, todas voltadas a plataforma web. Na figura 8 pode ser observado o ambiente de desenvolvimento com um exemplo de trecho de código.

Figura 8: Tela de desenvolvimento da plataforma CodeAcademy



Fonte: <https://i.ytimg.com/vi/YMw293DELs0/maxresdefault.jpg>

A plataforma promete abranger diversas faixas etárias, porém mostra-se pouco atrativa para o público infanto-juvenil por conta de seu design escuro e de nenhum atrativo lúdico. A proposta da plataforma é a de não trabalhar lógica de programação, exigindo assim um conhecimento prévio, como outras plataformas aqui apresentadas.

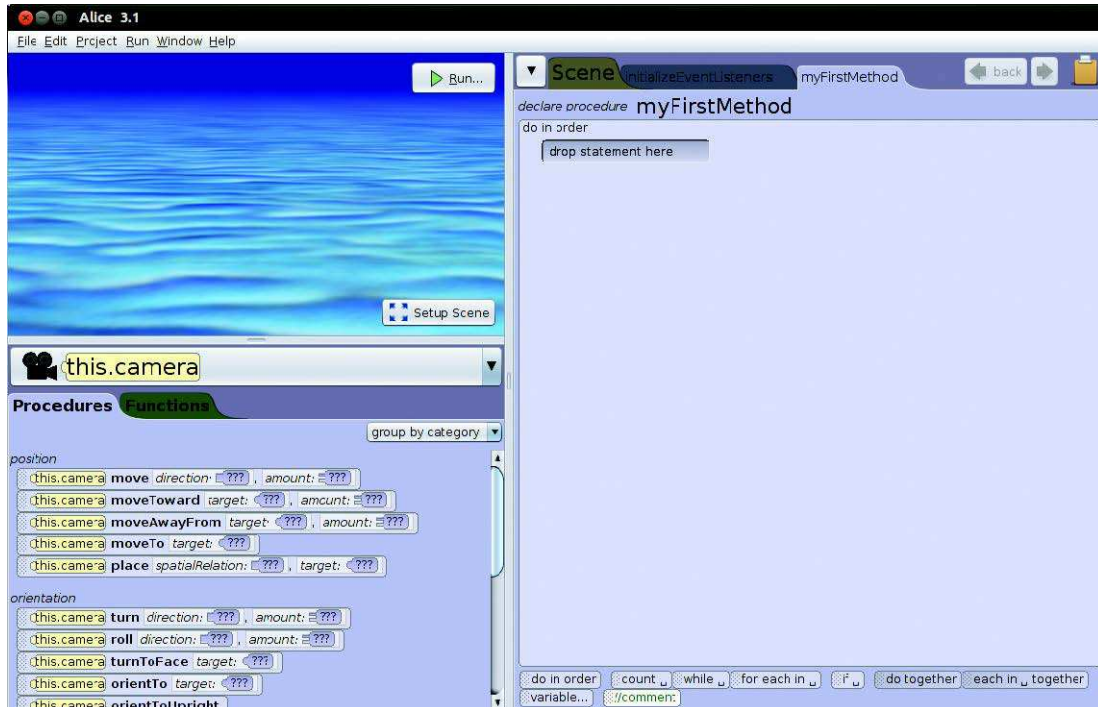
O artigo de BARBOSA et al. (2013) descreve a utilização da plataforma **CodeAcademy** para a construção de mapas para o aprendizado de educação ambiental descrevendo que a plataforma apresentou várias vantagens, pois seus cursos são bastantes acessíveis a novatos em tecnologia, pois se apresentam de forma prática compostos de lições explicativas seguido de instruções de fácil acesso em caso de dúvidas. A ferramenta combina o aprendizado, um editor de código e uma janela de visualização tudo numa mesma tela, à medida que se digita o código a ferramenta atualiza a visualização do código. Ao salvar um componente a ferramenta verifica, de imediato, se o mesmo executou corretamente. Para o autor a uso da plataforma mostrou-se eficaz, satisfatória e de fácil compreensão para a implementação do projeto.

### 3.7 Alice

**Alice** mostra-se uma plataforma inovadora, permitindo ao usuário a criação jogos com a manipulação de objetos 3D. Desenvolvido por alunos da Carnegie Mellon University, o

mesmo trabalha com ambientação às linguagens de programação orientadas a objeto através de sua representação na animação compartilhável.

**Figura 9: Tela de desenvolvimento da plataforma Alice**



Fonte: <http://imasters.com.br/wp-content/uploads/2014/05/fig02.jpg>

O ambiente possui dois tutoriais, um em PDF, que pode ser baixado no site, e outro apresentado durante a execução no próprio ambiente. Ambos mostram-se muito grandes e de pouca eficiência, entediando o usuário. O ambiente requer noções em lógica de programação. Sua interface é pouco organizada, dando maior ênfase a linguagem e assim deixando de lado o aspecto motivacional.








O artigo de MACKENZIE e ENGENHARIA (2008) descreve o uso do ambiente **Alice** no processo educacional junto aos cursos de Engenharia da Universidade Presbiteriana Mackenzie, **Alice** é um ambiente de programação que proporciona a usuários com pouca ou nenhuma experiência de computação possam programar caracteres ou objetos em um mundo virtual, de forma muito parecida a um moderno filme ou a um videogame. O software Alice tem sido usado como um recurso didático no processo de ensino de programação orientada a objetos em várias instituições de ensino ao redor do planeta. No nível superior, uma de suas aplicações está definida em cursos de introdução a programação orientada a objetos porque a criação de ambientes e animação utiliza esse paradigma. No ensino médio, a introdução à programação no ALICE para a construção de narração de histórias teve como objetivo estimular meninas a seguirem carreiras relacionadas à computação, possibilitar às crianças a exploração de ideias e estimular a auto expressão. O que torna este ambiente interessante e inovador é a parte dedicada para a edição do texto programa, ou editor do código programa (Figura 9), o que em outros ambientes de programação seria reconhecido como Programa Fonte (ou Código Fonte). Enquanto nos ambientes convencionais, esta parte se constitui de um simples editor de texto, no ALICE a edição ocorre por meio da técnica de arraste e solte. Em outras palavras, no ALICE não há a necessidade de digitação dos códigos, não ocorrendo

erros de sintaxe e eliminando uma grande barreira ao ensino de programação. Os autores acreditam que o software ALICE pode ter um impacto no processo educacional se devidamente incorporado junto ao ensino, tanto no nível médio como no ensino superior. Nesse sentido, acredita-se que a proposta apresentada neste artigo possa ser divulgada e, na medida do possível, incorporada, aos diversos cursos de engenharia com vistas à melhoria do ensino e aprendizagem de programação e computação.

### 3.8 Análise de trabalhos relacionados

Os trabalhos analisados estão resumidos na Tabela 2, de modo a permitir uma composição de aspectos e também a identificação de motivadores para o desenvolvimento do trabalho proposto.

**Tabela 2: Tabela comparativa entre os softwares analisados**

	1	2	3	4	5	6	7
	 App Inventor	 URI	 SCRATCH	 CODE COMBAT	 CODE CADEMY	 codecademy	 Alice
A que propõe o Ambiente (Ensino de Programação)?	Aplicativos móveis	C++, Java ou Python	Conceitos Matemáticos e Computacionais	Python, JavaScript, C++ entre outras.	Java e C#	Java e PHP	POO
Exige conhecimento prévio de linguagens de programação ou de lógica?	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Informa ao usuário quais são os conhecimentos prévios exigidos?	Não	Não	Não	Não	Sim	Não	Não
Apresenta elementos motivacionais?	Baixo	Baixo	Baixo	Alto	Baixo	Baixo	Médio (3D)
O Ambiente é Lúdico?	Não	Não	Não	Sim	Não	Não	Não
Possui Narrativa?	Não	Não	Não	Sim	Não	Não	Não
Possui Tutorial On-Line?	Sim	Não	Não	Não	Sim	Sim	Sim
Nível de Conhecimento ao qual se destina?	Iniciante	Intermediário	Intermediário	Intermediário	Intermediário	Intermediário	Intermediário
Tipo de Ambiente?	Web	Web	Web (Online)	Web	Web	Web e Mobile	Desktop
Possui recursos para Mineração de Dados?	Não	Não	Não	Não	Não	Não	Não

**Fonte: Elaborado pelo autor, Canoas 2016**

Os ambientes gamificados estudados descrevem um cenário onde diversos elementos e dimensões estão envolvidas. Convém explicitar que a linguagem de programação Lua, escolhida na elaboração deste trabalho, é pouco explorada no âmbito de jogos educativos, tanto no meio acadêmico quanto no meio comercial, sendo difícil encontrar exemplos de sua utilização.

Também pode ser observado que existem jogos educativos de ensinamento de linguagens de programação, para todos os níveis de conhecimento. Desde o App Inventor, que requer um nível de conhecimento baixo, ao Alice, que requer um nível de conhecimento prévio mais elevado. O ambiente produzido neste trabalho almejou posicionar-se em um nível anterior ao App Inventor, pois, além de requerer nenhum conhecimento prévio, nele o aluno não escreverá os algoritmos, os produzirá ao realizar atividades lúdicas.

Nos trabalhos analisados entre outros não estão presentes narrativas que são utilizadas para transmitir informações e guiar os alunos, permitindo assim uma maior interatividade com o ambiente e em consequente motivando-os a superar seus desafios de uma forma lúdica e prazerosa. Outro ponto observado nos trabalhos analisados refere-se a falta de ludicidade. Em ambientes lúdicos, a carga de informação pode ser muito maior, os apelos sensoriais podem ocorrer em grande número e isso mantém o interesse do aluno e facilita a aprendizagem.

Também observou-se que nos trabalhos analisados não estão presentes elementos de apoio ao professor, o que pode ocasionar uma perda importante de informações geradas pelo

processo de uso do ambiente pelos alunos. Este processo, caso acompanhado, pode ser de interesse para geração de indicações sobre as dificuldades e necessidades dos alunos.



## 4 METODOLOGIA

Neste capítulo é apresentada a metodologia adotada para implementar os aspectos da gamificação neste trabalho.

Serão apresentados abaixo a estrutura do *Framework* para gamificação adotado e a pesquisa de opinião realizada junto ao público alvo para identificar qual o cenário mais adequado e motivador em um ambiente gamificado de ensino de lógica de programação.

### 4.1 *Framework* de Gamificação

Para estruturar a abordagem de gamificação do trabalho e para apoiar o levantamento dos requisitos foi utilizado um modelo de *framework* para gamificação elaborado pela professora Dra. Eliane Schlemmer e sua equipe<sup>12</sup>.

A dinâmica de utilização do *framework* consiste em responder uma série de perguntas que permitem esclarecer as necessidades de cada situação, conforme apresentado na tabela 3, já que um ambiente gamificado pode ser dividido em três componentes que dão nome ao *framework*: mecânicas, dinâmicas e estéticas, que representam em um ambiente, respectivamente, as regras, o sistema e o aspecto lúdico (HUNICKE; LEBLANC; ZUBEK, 2004).

Na primeira coluna da tabela 3 são apresentados os elementos que deverão compor o ambiente gamificado. A segunda coluna descreve de forma sucinta o que se espera do elemento apresentado.

**Tabela 3: *Framework* para gamificação**

ELEMENTOS	DESCRIÇÃO
<b>Jogue (considere diferentes naturezas de jogos – analógicos, digitais e híbridos; diferentes plataformas: arcade, consoles, PC, tablets, smartphones; diferentes tipos de jogos: quebra-cabeça, ação, suspense, aventura, luta, esporte, corrida, simulação, RPG, RTS, MMORPG,...</b>	Você precisa vivenciar o mundo dos games, portanto, jogue para que você possa atribuir/construir significados e assim ter elementos para identificar o que sente/aprende ao jogar, como se sente e como aprende nesse contexto? O que mais lhe chama atenção? O que te proporciona um maior engajamento? Isso vai possibilitar que você preste atenção nos diferentes estilos de games e suas respectivas mecânicas e dinâmicas, bem como se sente em relação a elas. Dessa forma você dará sentido aos games no seu próprio processo de aprendizagem e, como é professor, poderá melhor estabelecer relações entre a forma como se sentiu/aprendeu ao jogar, as competências específicas da área de conhecimento em que atua e as

<sup>12</sup> Material apresentado na disciplina Seminário em Educação Digital do Programa de Pós Graduação em Educação da Universidade do Vale dos Sinos – Unisinos.

	competências didático-pedagógicas que possui.
<b>Identificação e Compreensão do problema.</b>	Ao refletir sobre o cotidiano, qual é o problema que identifica e acredita que a gamificação pode contribuir?
<b>Compreensão do contexto</b>	<p>Qual é o contexto no qual o problema está inserido?</p> <p>Quem são os sujeitos, qual é a cultura dos sujeitos?</p> <p>Qual é a cultura do ambiente?</p> <p>Caso haja locais/parceiros e fundamental mapear:</p> <p>1) Tipos de Informações: dados brutos, produto principal, relações, descrição de processos, modelos, conceitos principais, narrativas, patrimônio histórico (material e imaterial).</p> <p>2) Tipos de Interações: que os locais já proporcionam Exposição (organizadas a partir de uma lógica indutiva comparativas-temporalidades, espacialidades, ordem de grandeza).</p> <p>✓ Demonstração: há algum local onde é possível assistir alguma demonstração de um fenômeno. Tem algum vídeo para os visitantes assistir, o que eles mostram, se tem um tour mostrando algo, etc.</p> <p>✓ Experimentação: participa de um experimento prático de demonstração ou vivência de um fenômeno.</p> <p>✓ Investigação ou exploração: problematiza e escolhe as variáveis para fazer a experimentação ou é convidado a descobrir coisas numa postura dedutiva.</p> <p>✓ Simulação: demonstração ou investigação representada. (Há algum tipo de simulação digital nos lugares?).</p> <p>3) Tipos de atividades de ampliação - que tipo de ações podemos criar para ampliar a interação do público com o espaço:</p> <p>✓ Chamar atenção (o que pode nos chamar atenção por ser curioso ou diferente). O que os visitantes consideram mais atrativo no local?</p> <p>✓ Estabelecer relações (o que pode</p>

	<p>nos despertar para relacionar com o que conhecemos ou vivenciamos).</p> <ul style="list-style-type: none"> <li>✓ Dar pistas e propor exploração (o que pode virar pista).</li> <li>✓ Propor uma produção (escrita, visual, captura de imagem, composição, modelagem, dramatização, etc.) - ao sair dos espaços cada um pode escrever sua história, o que sentiu, o que lembrou, etc. ...um campo aberto para que o jogador possa dizer como se sentiu, lembrou ou achou que foi mais significativo.</li> <li>✓ Criar ou explorar histórias (narrativas pesquisadas e descoberta, inventadas, negociadas ou vivenciadas).</li> <li>✓ Explorar conteúdos simbólicos e imaginários - Trazer os símbolos e arquétipos mobilizados e até encontrados no espaço.</li> </ul> <p>Diferenciais.</p> <p>Possível tema para a missão.</p>
<p><b>Definição do escopo</b></p>	<p>A partir da identificação e compreensão do problema e da compreensão do contexto, é possível identificar:</p> <p>Quais são os objetivos com a gamificação?</p> <p>Que áreas do conhecimento que serão envolvidas?</p> <p>Quais as temáticas/temas a serem abordados?</p> <p>Que competências/habilidades poderão ser desenvolvidas e que conceitos estão a ela relacionados?</p> <p>Que ações, tipos de ações e input serão potencializadas?</p>
<p><b>Como desejo que os jogadores se sintam?</b></p>	<p>Quais são as emoções que desejo despertar nos sujeitos?</p>
<p><b>Quais são as mecânicas e dinâmicas?</b></p>	<p>Que mecânicas e dinâmicas podem ajudar a garantir que os sujeitos se sintam daquela forma e atinjam os objetivos pretendidos?</p> <p>A mecânica é a lógica do jogo, é o conjunto de regras, procedimentos e programações.</p> <p>Os elementos mais conhecidos da mecânica é o sistema de pontuação, quadro de ranking,</p>

	<p>recompensa, premiação, conhecidos como (points, badges, and leaderboards), ou seja, ponto, medalhas e quadro de ranqueamento. Mas a presença desses elementos podem ser indicadores de uma lógica indutiva e reativa que geralmente está presente numa abordagem mais empirista de educação e que pode ser redutora transformando tudo em estímulo e resposta, dessa forma, persuasiva promovendo verdades únicas e competitiva, pois o outro é sempre ou melhor ou pior que você, nunca o diferente com quem se pode aprender.</p> <p>Outros elementos podem estar presentes nessa mecânica como: desafios e missões mais complexas e problematizadoras que envolvem a relação com diversas variáveis e uma postura dedutiva e exploratória na busca de pistas, na realização de descobertas, no desenvolvimento de estratégias, na organização em clãs ou guildas que agrupam jogadores com características diversas e gera um empoderamento em grupo, para construção colaborativa e cooperativa – o que do ponto de vista pedagógico nos leva à perspectiva epistemológica interacionista-construtivista-sistêmica (inspirados, por exemplo, por elementos presentes nos Massively - (Multiplayer Online Role Play Games – MMORPG).</p> <p>É importante que essas mecânicas sejam coerentes com as competências ou habilidades que pretendemos proporcionar, ou seja, a ação do jogador no jogo deve refletir a nossa proposta educacional.</p>
<p><b>Construção da NARRATIVA</b></p>	<p>Qual a história que pode ser criada envolvendo a problemática identificada? A narrativa está aderente ao contexto, ao escopo, as emoções que desejo despertar? Propicia engajamento dos sujeitos? Fornece elementos para as mecânicas e dinâmicas pensadas? No caso do uso de símbolos e metáforas elas provocam o movimento simbólico esperado e fazem sentido para os sujeitos e para o objetivo da gamificação, áreas do conhecimento e temáticas/temas a serem abordados? A estética e os elementos</p>

	<p>sensoriais de narrativa são provocadores e imersivos?</p> <p>É possível criar narrativas ficcionais, mas que aproveitem dados de realidade e fatos históricos ou cotidianos e, muitas vezes faz parte da gamificação ter elementos que o jogador insira ou, pelo menos projete ou relacione com questões da sua própria história de vida. Uma boa narrativa é aquela em que os personagens são bem construídos e podem despertar emoções e mexer com questões inconscientes do jogador.</p> <p>A narrativa além de um enredo, de um tempo-espaco e de personagens, tem uma trajetividade, ou seja, uma linha que une os acontecimentos que pode ter uma característica mais ascendente ou heroica, que significa ter mais enfrentamento e mais revelações; ou um linha mais mística ou descendente, que proporcione situações mais contemplativas, mais questionamentos e questões reflexivas ou, ainda, a lógica dramática, que são aquelas que têm vários altos e baixos. Por tudo isso, o ato de contar histórias é muito mágico e transformador e pode ser determinante para o engajamento e principalmente pela oportunidade de trabalhar com o imaginário que é fundamental para o aprofundamento dos aprendizados.</p> <p>Um exercício que ajuda na construção de personagens e narrativas é pensar em histórias clássicas, mitos, contos de fada, lendas e associar com os personagens e com o destino da história. Mesmo que a história seja mais realista, essa associação arquetípica ajuda a perceber o movimento que queremos.</p> <p>Por exemplo: se queremos gamificar uma atividades sobre contabilidade. Qual o movimento associado? Equilíbrio. Um mito associado é da Temperança que equilibra os mundos, que pode ser associado a Deusa Perséfone, que vive tanto no mundo inferior como na terra equilibrando as coisas, ou seja, o movimento da gamificação tem que ser algo que lembre equilíbrio e os personagens devem buscar esse equilíbrio.</p>
--	--

<p><b>Caso a mecânica envolva o desenvolvimento de Missões observe a relação com os objetivos da gamificação, com as competências a serem desenvolvidas e ações a serem potencializadas.</b></p>	<p>Para a construção das Missões:</p> <p>Título da missão</p> <p>Qual o contexto?</p> <p>Qual o conceito principal da missão?</p> <p>Qual o objetivo? Como se relaciona com o objetivo geral, com as competências a serem desenvolvidas e ações a serem desenvolvidas?</p> <p>Quais as desafios e pistas que podem ser criadas?</p> <p>Quais as regras?</p> <p>Quais os retornos solicitados? (interação dos sujeitos)</p> <p>Quis os feedbacks? (fornecidos pelo sistema)</p> <p>Quais os achievements (XP) que essa missão vai possibilitar?</p> <p>Para a criação de missões é possível buscar inspiração nas pistas do método cartográfico de pesquisa-intervenção, inicialmente proposto por Kastrup (2007; 2008) e (Passos, Kastrup, Escóssia, 2009), mais especificamente, no que se refere a atenção cartográfica que é definida como concentrada e aberta, e que caracteriza-se por quatro variedades: o rastreio, o toque, o pouso e o reconhecimento atento. Essas quatro variedades da atenção cartográfica, podem ser adaptadas enquanto metodologia para o desenvolvimento, acompanhamento dos percursos de aprendizagem e avaliação das missões, conforme a seguir:</p> <p>Rastreio - (varreduras do campo – COMPREENSÃO DO CONTEXTO) – criar/encontrar as pistas (baseada no mapeamento anterior, presente no item “Compreensão do Contexto” na relação com os elementos presentes no item “Definição do escopo”), encontrar parceiros, criar/dar continuidade a narrativa da gamificação;</p> <p>Toque - (aciona o processo de seleção – CONCEITO PRINCIPAL DA MISSÃO, OBJETIVO DA MISSÃO) - desvendar os problemas da área, desvendar os mistérios (eureka), define o que é relevante;</p>
--	--

	<p>Pouso - (parada – zoom – DESAFIO E PISTAS A SEREM CRIADAS) - propor caminhos, construir a gamificação;</p> <p>Reconhecimento Atento (construir o mapa da gamificação – REGRAS, RETORNOS SOLICITADOS, FEEDBACKS, ACHIEVEMENTS)</p> <p>Para a construção de achievements por missão, novamente podemos nos inspirar na cartografia:</p> <p>Rastreo - explorador (caça as pistas) articulador (articulando as pistas com outras informações e experiências)</p> <p>Toque - observador (entendendo as pistas, desvendando os mistérios) e selecionador (selecionando o que é relevante)</p> <p>Pouso - tecelão (tecendo as pistas), colaborador (discutindo e construindo junto) e ator (desenvolvendo a autoria criativa)</p> <p>Reconhecimento Atento - o cartógrafo (mapeando o caminho e construindo o mapa da gamificação).</p> <p>REGRAS</p> <p>A gamificação é composta por trilhas e/ou missões.</p> <p>Cada missão tem objetivos, achievements e, também um conjunto de pistas que podem ser humanas (Pistas Vivas) e não humanas (Pistas em QR Codes e Realidade Aumentada). A ampliação do nível de EXP ou XP (experiência) o que permite progredir no jogo, ocorre na medida em que os jogadores desbloquearem os achievements de cada missão (aprendizagem).</p>
<p><b>Definição do ambiente/natureza da gamificação</b></p>	<p>Os sujeitos vão participar em ambientes físicos (na sala de aula, na escola, de casa, de locais/parceiros específicos, da cidade como um todo,...), digitais (aplicativo, mídias sociais, MDV3D, etc.) ou híbridos (RM, RA, etc); Qual a interface principal com o sujeito... tablets e/ou smartphones?</p>
<p><b>Como a gamificação vai acontecer na prática?</b></p>	<p>Descrição do processo como um todo, incluindo:</p> <p>Terá um aplicativo específico? Como os</p>

	<p>sujeitos terão acesso a ele? Do que vão precisar? Terá um mapa? Haverá deslocamento físico e geográfico? Como a narrativa surge? Como as missões são ativadas? Como serão apresentadas? (informações em áudio, texto, cutscene... haverá personagens em 2D ou 3D interagindo com os sujeitos? Quando for realizar as missões, essas serão compostas por um conjunto de pistas (QRCode, RM, RA e "Pista Viva", ou seja, uma pessoa do local que seja referência sobre o conhecimento necessário para desenvolver aquela missão). Como o sujeito ao realizar as missões amplia o seu nível de EXP ou XP, ou seja, nível de experiência?</p>
--	--

Fonte: (SCHLEMMER et al., 2015)

## 4.2 Questionário de pesquisa de opinião

Além da adoção da metodologia também foi realizada uma pesquisa de opinião junto aos alunos do primeiro ano do Curso Técnico em Desenvolvimento de Sistemas Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS – Campus Canoas para saber qual poderia ser o cenário mais adequado e motivador para um ambiente gamificado de ensino e aprendizagem de lógica de programação. A seguir são descritos os elementos desta pesquisa e destacados os principais resultados.

Para o levantamento do perfil do jogador quanto ao cenário gamificado a ser implementado, foi elaborado um questionário denominado “Perfil do Jogador”, composto por 7 questões que buscam caracterizar o perfil dos usuários do ambiente.

O questionário foi aplicado no início do trimestre de 2016 junto a um grupo de alunos. Este grupo de alunos, que será referenciado aqui como grupo de controle, foi composto por 20 alunos voluntários do Curso Técnico em Desenvolvimento de Sistemas. A turma no total possui 40 alunos, mas apenas o grupo de controle utilizará o ambiente gamificado.

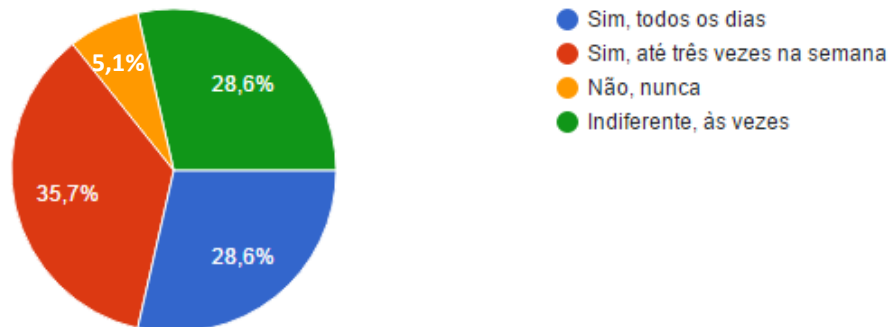
A seguir são descritas as perguntas deste questionário e as respostas obtidas.

A pergunta 1 foi sobre a idade dos respondentes. Neste caso, a resposta levou à média de idade de 15 anos entre os respondentes.

A pergunta 2 foi “Você gosta de jogos eletrônicos? Com que frequência costuma jogar?”. A figura 10 ilustra graficamente os resultados obtidos para a pergunta 2. Deve se destacar que 64,3% joga todos os dias ou até 3 vezes por semana.



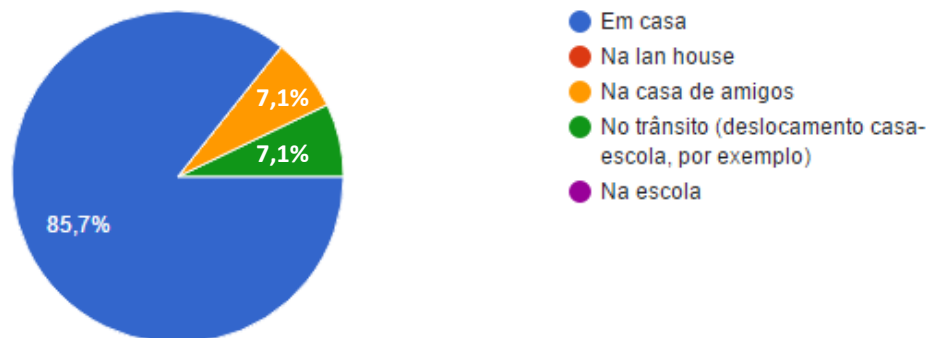
**Figura 10: Perfil de gosto e frequência por jogos eletrônicos**



Fonte: Elaborado pelo Autor, Canoas: 2016

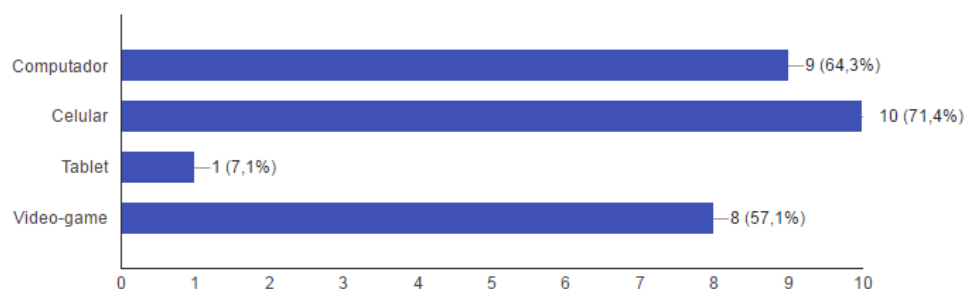
A pergunta 3 foi “Onde você mais costuma jogar?”. A figura 11 ilustra graficamente os resultados obtidos para a pergunta 3. Deve se destacar que a grande maioria usa jogos em casa.

**Figura 11: Perfil do local aonde mais acontece a utilização dos jogos**



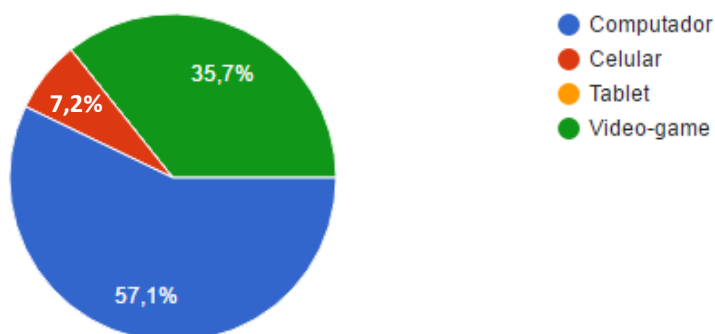
Fonte: Elaborado pelo Autor, Canoas: 2016

A pergunta 4 foi “Em qual plataforma você costuma jogar?”. A figura 12 ilustra graficamente os resultados obtidos, contendo a indicação clara de uso de dispositivos móveis em grande parte dos casos.

**Figura 12: Perfil da plataforma de jogo utilizada**

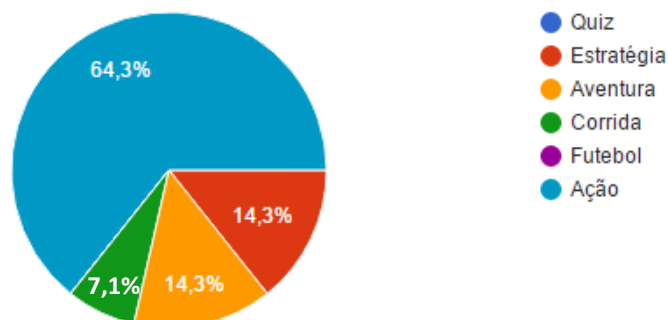
Fonte: Elaborado pelo Autor, Canoas: 2016

A pergunta 5 foi “Qual a sua plataforma favorita?”. A figura 13 ilustra graficamente os resultados obtidos. Neste caso, a preferência por computador e videogame está associada com as facilidades de utilização e resultados visuais proporcionados.

**Figura 13: Perfil da plataforma de jogo favorita**

Fonte: Elaborado pelo Autor, Canoas: 2016

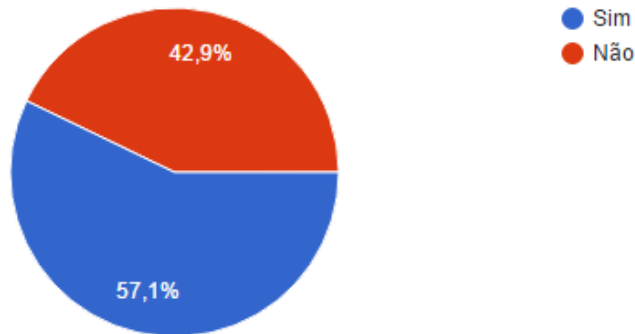
A pergunta 6 foi “Qual a categoria de jogos eletrônicos que você mais joga?”. A figura 14 ilustra graficamente os resultados obtidos. Observa-se a preferência por jogos com desafios, como o caso de Quiz e Estratégia.

**Figura 14: Perfil da categoria de jogo mais utilizada**

Fonte: Elaborado pelo Autor, Canoas: 2016

A pergunta 7 foi “Em sua opinião, o uso de jogos auxilia no seu aprendizado acadêmico?”. A figura 15 ilustra graficamente os resultados obtidos. Observa-se que neste caso não foi identificado um consenso a respeito, mas considera-se que este fator pode estar associado com as experiências prévias dos alunos.

**Figura 15: Perfil sobre o auxílio de jogos no aprendizado**



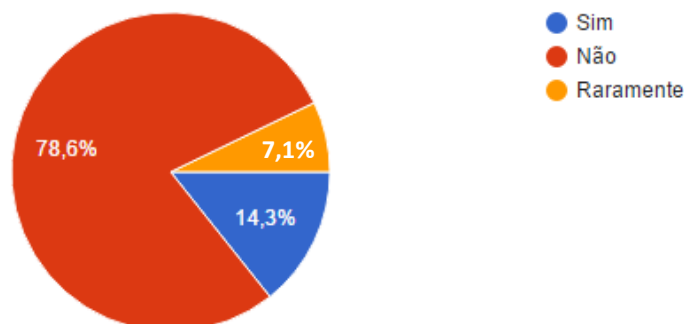
**Fonte: Elaborado pelo Autor, Canoas: 2016**

Como complemento da pergunta 7 foi adicionada a pergunta sobre o quanto o aluno acredita que “O uso de jogos auxilia no seu aprendizado acadêmico?”.

A seguir estão agrupadas as respostas obtidas, de acordo com sua similaridade: a) Estimulação cognitiva; b) Ajuda a melhorar o raciocínio lógico e o nível de conhecimento em Inglês; c) Computador; d) Melhorar a lógica e criatividade do aluno; e) Alguns jogos nos auxiliam a desenvolver raciocínio rápido e nos fazem usar a imaginação; f) Raciocínio; g) O auxílio no aprendizado depende muito da abordagem do jogo, o seu tema, sua categoria, e muitos outros fatores. Mas em geral podem nos ajudar em uma melhor concentração, pensamento mais rápido, como também influência na memória, o que pode nos ajudar dentro da sala de aula; h) De forma que os alunos tem mais interesse em aprender quando são motivados por coisas que eles gostam de fazer, como jogar, pro exemplo.

A pergunta 8 foi “Você costuma jogar em redes sociais?”. A figura 16 ilustra graficamente os resultados obtidos.

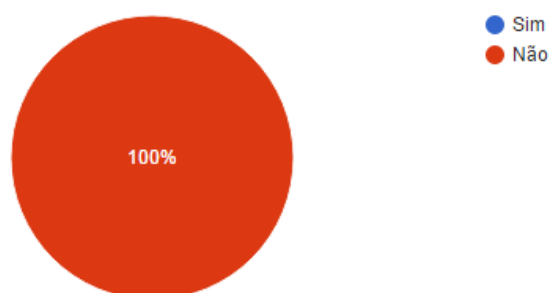
**Figura 16: Perfil do uso de jogos eletrônicos em redes sociais**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

Como complemento da pergunta 8 foi feito o seguinte questionamento: “Se sim, você compartilha sua evolução no perfil?”. A figura 17 ilustra graficamente os resultados obtidos. Estas duas últimas figuras indicam para este grupo de entrevistados um pequeno grau de interação e compartilhamento em redes sociais.

**Figura 17: Perfil do compartilhamento das evoluções do uso de jogos eletrônicos em redes sociais**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

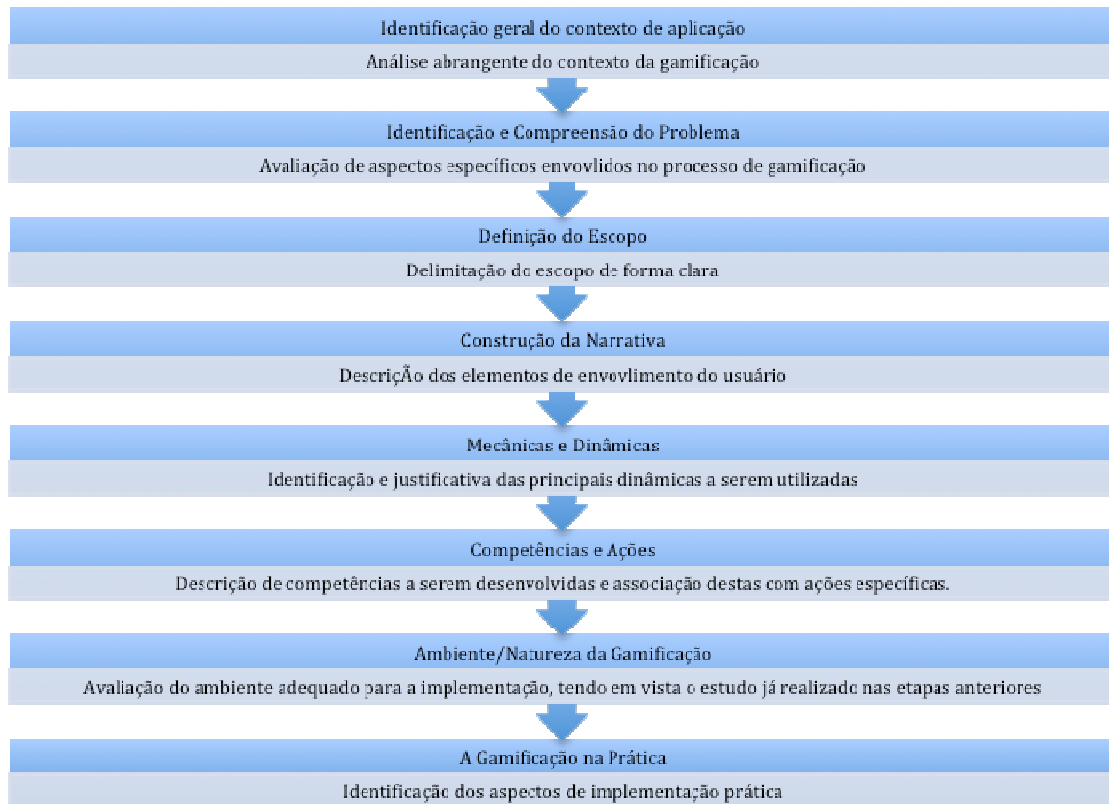
Após estas etapas, iniciou-se o desenvolvimento do ambiente propriamente dito, sendo que, na sequência, foram realizados testes junto aos alunos desta mesma instituição, como apresentado no capítulo 6.

Depois da implementação e dos testes, foram cruzados os resultados obtidos com as premissas levantadas durante a fase de levantamento de requisitos e desenvolvimento do ambiente.

### **4.3 Instanciação do Ambiente Gamificado**

Para o caso do ambiente desenvolvido, as perguntas do framework de gamificação adotado foram respondidas, conforme seguem descritas nos itens 4.1.1 até 4.1.8 a seguir.

A tabela 4, abaixo, resume a visão geral destas etapas.

**Tabela 4: Etapas Gerais do Framework de Gamificação Adotado**

**Fonte: Elaborado pelo autor, Canoas 2016**

A seguir são apresentados os detalhes do trabalho realizado em cada etapa.

#### 4.3.1 Identificação geral do contexto de aplicação

O mercado de trabalho na área da Tecnologia da Informação (TI) vive uma grande ascensão no Brasil. Com isso observa-se também um crescente aumento na procura por cursos de todos os níveis (superiores e técnicos) relacionados a este mercado (CL; GIAMBIAGI, 2010).

Contudo, este grande aumento na oferta de vagas em cursos relacionados às áreas da tecnologia não se reflete em um grande número de estudantes diplomados. Segundo pesquisa do Sindicato das Instituições Privadas de Ensino Superior do Estado de São Paulo (SEMESP), a evasão em cursos relacionados à Tecnologia da Informação (TI) supera a marca dos 73%.

Segundo Célio Antunes, presidente da Impacta e um dos fundadores da Microtec, fabricante brasileira de computadores que fechou as portas no início dos anos 2000, muitos calouros se surpreendem com a demanda de cálculos e outros conhecimentos nas áreas de exatas. "Realizo várias palestras para os vestibulandos e indico cursos técnicos, de menor duração, para quem tem dúvidas de qual carreira seguir", explica Antunes (TEECH, 2016).

#### 4.3.2 Identificação e Compreensão do Problema

É muito comum deparar-se com alunos que enfrentam grandes dificuldades já nas primeiras disciplinas de cursos relacionadas à TI, especialmente em disciplinas ligadas à

programação (MENDES, 2002). Existem diversas hipóteses sobre o que leva um aluno a encontrar dificuldades para aprender lógica de programação. MENDES (2002) e também MOTIL e EPSTEIN (2000) defendem a ideia de que as dificuldades no aprendizado da programação estão atreladas a diversos aspectos, tais como:

- Programar requer um alto nível de abstração, habilidade que muitas pessoas têm dificuldade em aprimorar;
- As linguagens de programação utilizadas em muitas instituições de ensino possuem sintaxes demasiadamente complexas, dificultando a compreensão dos algoritmos.

GOMES e MENDES (2007) ainda ressaltam que outra grande barreira no aprendizado da programação é o ritmo de ensino e aprendizagem das aulas imposto ao aluno pelos professores, uma vez que na esmagadora maioria das instituições de ensino sejam estas públicas ou privadas as aulas não se dão de forma individualizada, devido, é claro, ao grande contingente de alunos.

SÁNCHEZ (2004) ainda observa que existem dificuldades originadas no ensino inadequado ou insuficiente seja porque a organização do mesmo não está bem sequenciado ou não se proporcionam elementos de motivação suficientes; seja porque os conteúdos não se ajustam às necessidades e ao nível de desenvolvimento do aluno ou não estão adequados ao nível de abstração, ou não se treinam as habilidades prévias; seja porque a metodologia é muito pouco motivadora e muito pouco eficaz.

NG e BEREITER (1991) ainda enfatizam que a falta de motivação encontrada por alguns alunos é algo realmente grave, pois este tipo de aluno dificilmente conseguirá tornar-se um profissional bem sucedido.

Finalmente, para ALMEIDA et al. (2002) o principal obstáculo que impede a muitos alunos de aprenderem a programar é a falta de motivação ao se depararem com as ferramentas e linguagens de programação cada vez mais sofisticadas, requerendo cada vez mais conhecimento e principalmente a abstração por parte dos programadores.

#### 4.3.3 Definição do Escopo

Este projeto gamificado tem como o objetivo a produção de um ambiente no formato digital voltado a educação da Lógica de Programação que permitisse aos estudantes iniciantes em cursos de TI obterem uma ampla visão de forma muito simplificada e objetiva a respeito da linguagem de programação Lua, abordando tanto sua sintaxe básica, como também recursos fundamentais da linguagem, baseado no primeiro capítulo do livro “Programando em Lua” (IERUSALIMSKY, 2013). Neste contexto, foram desenvolvidos questionamentos em forma de exercícios de formato bem simples como uma forma de estimular o aluno iniciante a não só compreender como também manipular variáveis, utilizar tabelas (vetores e matrizes), cláusulas condicionais (*if*) e estruturas de repetição (*for*, *repeat until*).

Para CASTRO et al. (2002), “um grande número de teorias de aprendizagem hoje utilizadas são categóricas ao afirmar que a aprendizagem só ocorre com a experiência, é o aprender fazendo”.

Segundo C. RAPKIEWICZ e G. FALKENBACH (2006), os ambientes que possuem atividades que exercitam a habilidade mental e a imaginação agradam, entretêm e prendem a atenção, além de entusiasmar e ensinar com maior eficiência. Isso acontece porque estes ambientes transmitem as informações necessárias de diversas formas, estimulando vários sentidos. Em um ambiente lúdico, a carga de informação pode ser muito maior, os apelos sensoriais inúmeros e isso mantém o interesse do aluno e facilita a aprendizagem. Assim, seguem os autores, “[...] toda a atividade que incorporar a ludicidade pode se tornar um recurso

facilitador do processo de ensino e aprendizagem em qualquer área” (C. RAPKIEWICZ, G. FALKENBACH, 2006).

A escolha pela linguagem Lua foi motivada por esta ser uma linguagem de programação que alia uma sintaxe simples, algo que vai ao encontro de um dos principais motivos pela alta evasão em curso de TI: a complexidade das ferramentas utilizadas já nos primeiros períodos dos cursos a recursos robustos (IERUSALIMSKY; MOURA, 2016).

A interação do aluno com o ambiente gamificado será feito através da ação de apontar e clicar, permitindo assim ao mesmo alcançar o objetivo de cada um dos níveis presentes no ambiente. O aluno precisará controlar o *mouse* ou utilizar o toque de tela, em dispositivos que possuam suporte a este tipo de recurso para clicar ou tocar. Desta forma irá movimentar os objetos, realizando as ações que lhe forem solicitadas. A navegação dos menus também será feita por intermédio do *mouse* (MARINACCI, 2003).

Inicialmente, a ideia era que o desenvolvimento do ambiente se desse com a utilização de *frameworks* com a própria linguagem Lua, mas, tendo em mente que este jogo tem como objetivo auxiliar os usuários na busca de um conhecimento, é natural almejar que o mesmo seja facilmente acessível e executável. Assim foi decidido que seriam utilizadas tecnologias web para o desenvolvimento do jogo. Segundo LOPES (2013), sistemas voltados a web podem ser facilmente utilizados por pessoas que possuam praticamente qualquer dispositivo provido de um navegador web<sup>13</sup>. Desta forma três das linguagens mais utilizadas no desenvolvimento web foram utilizadas: HTML<sup>14</sup>, CSS<sup>15</sup> e Javascript<sup>16</sup>.

#### 4.3.4 Construção da Narrativa

O ambiente gamificado foi dividido em uma série de níveis, todos contando com a presença de um agente pedagógico animado, que apresenta o desafio a ser reproduzido além de, em alguns momentos, apresentar informações sobre o recurso da linguagem Lua que será utilizado no nível em questão.

Segundo ROCHA et al. (2010), cabe destacar, ainda, que para a passagem de níveis, não se espera domínio absoluto dos conteúdos ou uma suposta perfeição, algo inclusive difícil de se definir. Entretanto, espera-se que o aluno atinja um grau de excelência, ou seja, não cometa erros fundamentais sobre o conteúdo daquele nível. Também se espera que o mesmo possua boa desenvoltura com seus conceitos.

Segundo REATEGUI et al. (2006), “os agentes pedagógicos animados podem cumprir um papel comunicativo muito importante. Ao assumir uma personalidade interessante e gerar

---

<sup>13</sup> Também conhecido pelos termos em inglês *web browser* ou simplesmente *browser*, é um programa de computador que habilita seus usuários a interagirem com documentos virtuais da Internet.

<sup>14</sup> Sigla de **HyperText Markup Language**, expressão inglesa que significa "Linguagem de Marcação de Hipertexto". **Consiste em uma linguagem de marcação utilizada para produção de páginas na web**, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador.

<sup>15</sup> *Cascading Style Sheets* (CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. O seu principal benefício é a separação entre o formato e o conteúdo de um documento.

<sup>16</sup> Linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade de este script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

empatia nos alunos, o agente animado introduz um componente afetivo que pode facilitar o processo de aprendizagem”. Ao final do nível, o agente irá informar ao aluno se o mesmo conseguiu concluir o nível com sucesso, parabenizando o mesmo caso a resposta esteja correta.

O ambiente não possui um roteiro, ou seja, não existe enredo por trás do mesmo. Na realidade, a narrativa do ambiente é dividida em seis "unidades" distintas:

- 1) **Tutorial:** Apresenta ao aluno informações sobre a linguagem de programação Lua, conceitos básicos sobre a linguagem de programação Lua abordada nas demais unidades do ambiente, além de apresentar a mecânica funcional do ambiente.
- 2) **Variáveis:** Apresenta ao aluno questões sobre o que são e como se dá a manipulação de variáveis na linguagem Lua.
- 3) **Operações Matemáticas:** Apresenta ao aluno questões sobre o aprendizado de operadores matemáticos básicos tais como soma, subtração, divisão e multiplicação na linguagem Lua.
- 4) **Tabelas:** Apresenta ao aluno questões sobre o aprendizado de tabelas através do conceito de matrizes e vetores na linguagem Lua.
- 5) **Estruturas condicionais:** Apresenta ao aluno questões sobre o aprendizado de estruturas condicionais como “se/senão/então” e também um aprendizado sobre as estruturas de repetição tais como “para/repita-até” e “enquanto-faça” na linguagem Lua.

#### 4.3.5 Mecânicas e Dinâmicas

A cada nível o ambiente apresenta ao aluno um pequeno desafio a ser cumprido, onde as ações do aluno são controladas por um *script*<sup>17</sup>, que permite verificar a validade das jogadas e determinar, ao final da partida, se o aluno cumpriu corretamente o desafio proposto armazenando em um log os erros, acertos e tempo de resposta de cada questão. Este mesmo *script* também controlará a mecânica de mineração de dados educacionais que servirá como um norteador ao professor, permitindo assim que, através desta coleta de dados sobre o comportamento do aluno, posteriormente o professor possa elaborar suas aulas de acordo com as dificuldades apontadas neste comportamento. O grau de dificuldade dos níveis é crescente, à medida que novos conteúdos referentes a linguagem Lua vão sendo acrescentados.

O ambiente não conta com nenhum tipo de recompensa, como pontuações ou bonificações, o prêmio entregue ao aluno por concluir um nível do ambiente é o acesso ao nível posterior, além de uma explanação mais detalhada a respeito do nível que o aluno acabou de completar, mostrando um pseudo-código sobre como a ação efetuada seria programada na linguagem Lua.

Segundo SCHLEMMER (2014) a gamificação pode ser pensada a partir de pelo menos duas perspectivas: enquanto persuasão, estimulando a competição, tendo um sistema de pontuação, de recompensa, de premiação etc. Isso, do ponto de vista da educação, reforça uma perspectiva epistemológica empirista. Enquanto construção colaborativa e cooperativa, instigada por desafios, missões, descobertas, empoderamento em grupo, o que do ponto de vista da educação nos leva à perspectiva epistemológica interacionista-construtivista-

---

<sup>17</sup> “Roteiros” seguidos por sistemas computacionais e trazem informações que são processadas e transformadas em ações efetuadas por um programa principal.



sistêmica (inspirados, por exemplo, por elementos presentes nos Massively Multiplayer Online Role Play Games – MMORPG).

#### 4.3.6 Competências e Ações

Os desafios propostos no ambiente estão relacionados a aprendizagem da linguagem Lua mediante a utilização de dois dos conceitos do método cartográfico de pesquisa-intervenção, propostos por KASTRUP (2007): Rastreo e Toque, uma vez que em todos os níveis o jogador deve associar a sua capacidade de observação aos itens dispostos na tela, com o conhecimento na linguagem Lua que ele está obtendo ao completar os níveis do jogo.

#### 4.3.7 Ambiente/Natureza da Gamificação

O ambiente foi implementado e testado para ser executado através de computadores pessoais, independente do sistema operacional disponível, sendo apenas necessário que o mesmo possua uma versão atualizada de um navegador de Internet com motor Gecko<sup>18</sup> ou Webkit<sup>19</sup>.

Por tratar-se de um produto desenvolvido com tecnologias *web*, o jogo também pode ser compatível com dispositivos móveis, porém, tal compatibilidade foi testada unicamente utilizando o simulador do Firefox OS na versão 2.0, podendo apresentar incompatibilidades em outras plataformas móveis.

Segundo a pesquisa Game Brasil 2016, realizada pela (SIOUX, BLEND, 2016), o jogador brasileiro se consolida com o perfil multiplataforma, apresentando o índice de 70,8%. O smartphone é o mais popular dentre as plataformas (77,2%), seguido pelo computador (66,9%).

#### 4.3.8 A Gamificação na Prática

A gamificação dar-se-á através de um ambiente em formato digital, aproximando ao conceito de "aprendizagem através de jogos" (em inglês, Game-Based Learning), onde um jogo é transformado num ambiente de aprendizado (GOIRI, 2015). Sendo o mesmo disponível em formato *web* permitindo assim o seu acesso a qualquer momento pelo aluno.

A narrativa do ambiente foi ambientada sobre a perspectiva de uma delegacia de polícia aonde o aluno através da ajuda do “agente animado” será instigado a resolver diversos questionamentos e como recompensa ao mesmo será liberado um novo nível.

O ambiente conta com menus ilustrados através do uso de texto e imagens, aonde o aluno poderá acessar os níveis ou adquirir informações sobre o desenvolvimento do mesmo.

Com a resolução das tarefas proposta a cada nível o aluno estará intrinsecamente ampliando seus conhecimentos em lógica de programação através da ludicidade do ambiente, ou seja, o aluno estará aprendendo sem perceber que esta.

---

<sup>18</sup> Motor de layout com código aberto usado em aplicativos desenvolvidos pela Mozilla Foundation e a Mozilla Corporation (principalmente o navegador Firefox), como também em muitos outros projetos de software com código aberto.

<sup>19</sup> Motor de renderização utilizado em navegadores web para renderizar páginas. O WebKit é utilizado por navegadores como o Safari, e o Google Chrome.

## 5 DESENVOLVIMENTO DO AMBIENTE

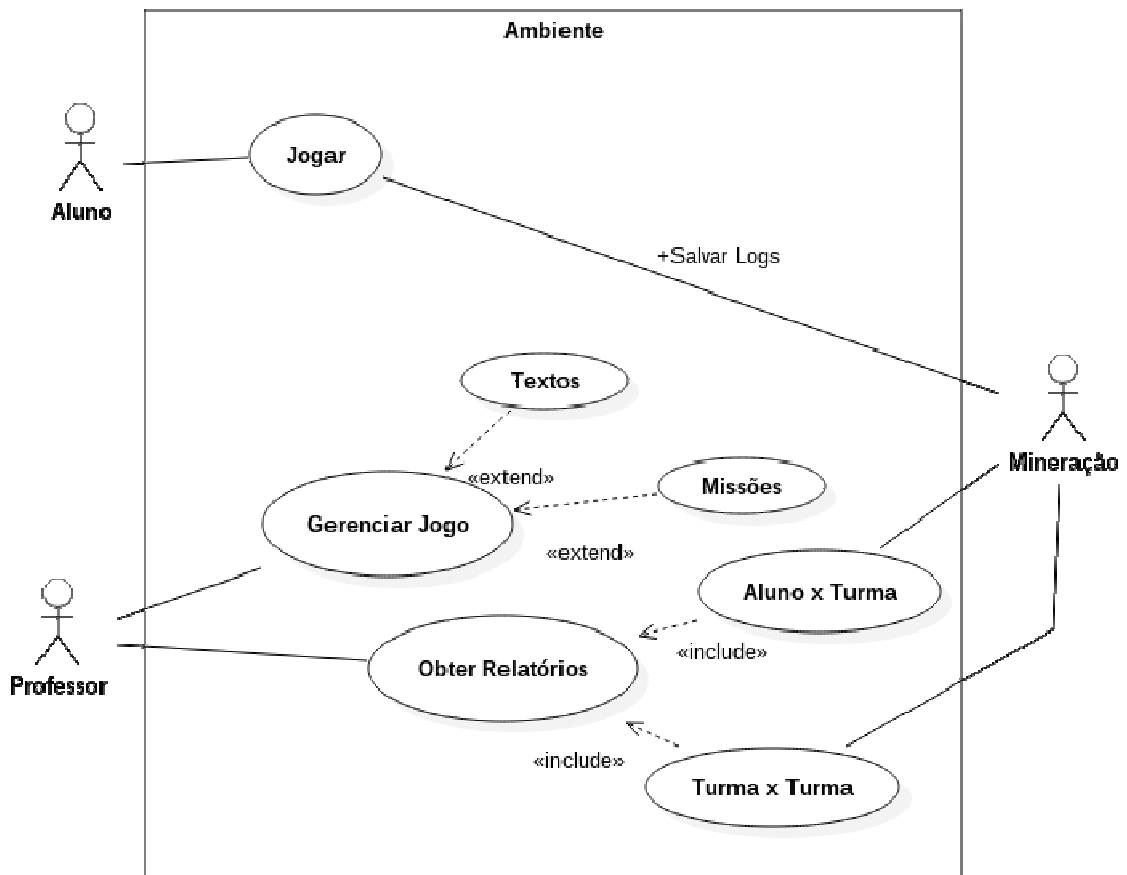
Neste capítulo será apresentado o desenvolvimento da modelagem e a posterior implementação do ambiente gamificado nomeado “Tri-Lua”.

A ferramenta desenvolvida buscou aplicar os conceitos já apresentados de gamificação e de ensino e aprendizagem de lógica de programação, contando com atividades lúdicas envolvendo o uso de agentes animados, em um ambiente que possibilita ao aluno aprender as características iniciais da lógica de programação através de conceitos básicos da linguagem de programação Lua, abordando tanto sua sintaxe, como também recursos fundamentais, gerando subsídios para a aplicação de técnicas de Mineração de Dados Educacionais.

### 5.1 Modelagem do Ambiente

Após o levantamento de requisitos através do *framework* para gamificação, iniciou-se a fase de modelagem do ambiente, inicialmente optou-se pela criação de um diagrama de caso de uso para que se pudesse ter uma visão mais ampla de todo o ambiente a ser implementado conforme apresentado na figura 18.

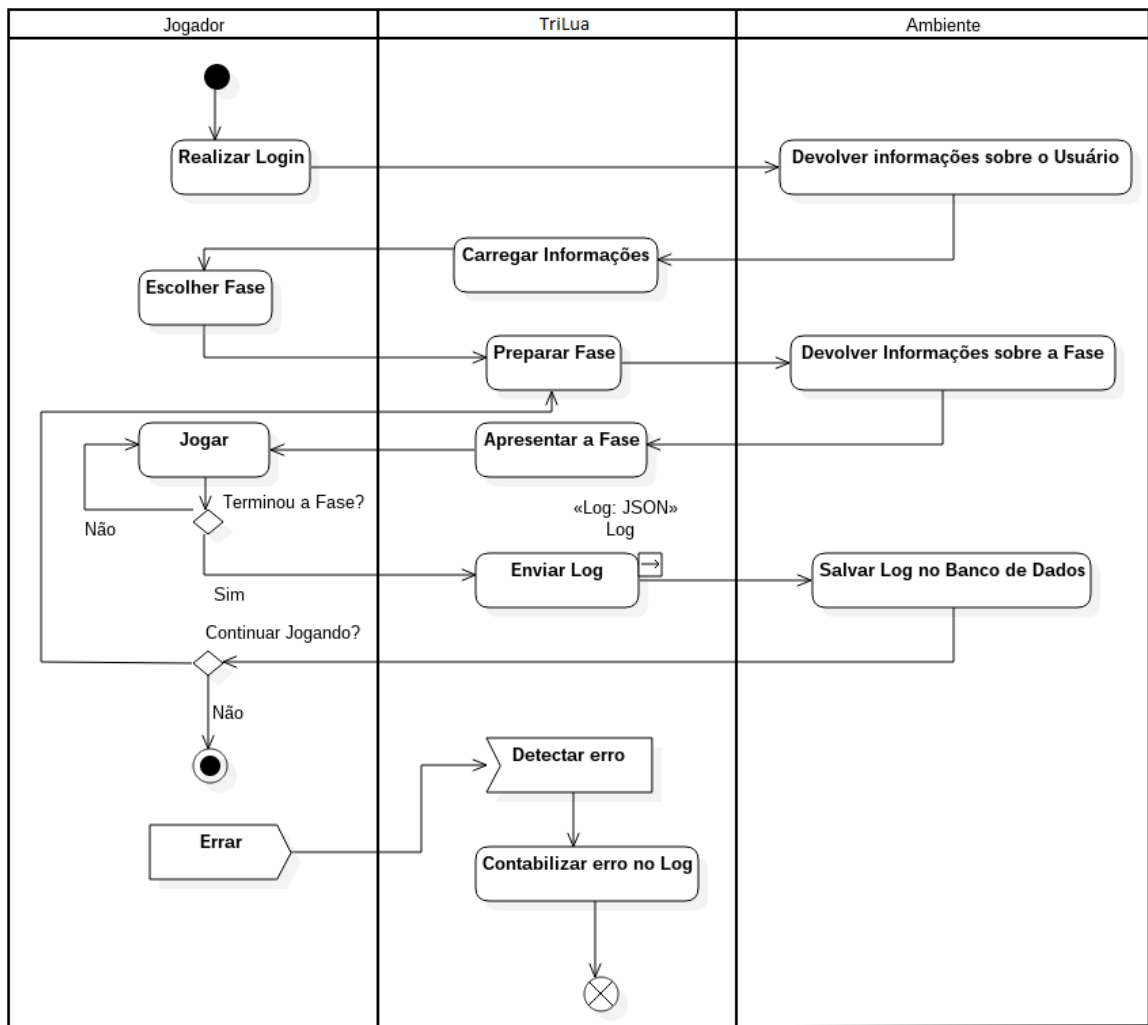
Figura 18: Diagrama de Caso de Uso do ambiente gamificado



Fonte: Elaborado pelo autor, Canoas 2016

Para que se pudesse ter uma visão de como todo o ambiente deveria ser implementado abrangendo desde sua jogabilidade, conforme regras definidas pelo *framework* de gamificação, até a mineração de dados que seria utilizada como ferramenta de suporte ao ensino pelo professor, foi elaborado um diagrama de sequência de todo o ambiente conforme apresentado na figura 19.

**Figura 19: Diagrama de Sequencia do ambiente gamificado**



**Fonte: Elaborado pelo autor, Canoas 2016**

A cada nível o ambiente apresenta ao aluno um pequeno desafio a ser cumprido, onde as ações do aluno são controladas por um *script*, que permite verificar a validade das jogadas e determinar, ao final da partida, se o aluno cumpriu corretamente o desafio proposto armazenando em um log os erros, os acertos e o tempo de resposta de cada questão. Este mesmo *script* também controlará a mecânica de mineração de dados educacionais. O grau de dificuldade dos níveis é crescente, na medida que novos conteúdos referentes a linguagem Lua venham a ser acrescentados.

## 5.2 Níveis disponíveis no ambiente

A seguir serão detalhados cada um dos níveis do ambiente:

### 5.2.1 Tutorial

Este é o nível introdutório do ambiente, estruturado em dois ambientes. No primeiro ambiente, são apresentados conceitos importantes sobre o jogo, como por exemplo, qual será a linguagem de programação e os conceitos básicos que os alunos irão aprender. No segundo ambiente, é apresentada a mecânica do jogo em termos de jogabilidade, baseando-se no conceito de "apontar e clicar" pois, segundo KASTRUP (2007), “O toque é sentido como uma rápida sensação, um pequeno vislumbre, que aciona em primeira mão o processo de seleção”.

Para alcançar o objetivo deste e dos demais níveis do ambiente, o aluno necessitará ter o controle do mouse ou utilizar-se de uma tela de toque no caso de dispositivos móveis, para selecionar e movimentar objetos, e realizar as ações que lhe forem solicitada (MARINACCI, 2003).

### 5.2.2 Variáveis

Este nível do ambiente contempla alguns exercícios sobre a utilização básica de variáveis em linguagem de programação Lua. Nele o aluno será capaz de observar que diferentemente de linguagens de programação tradicionais como C e Java, em Lua não existe a declaração de variáveis.

Também dentro deste nível o aluno é introduzido a outra característica importante de conceitos básicos da linguagem de programação Lua, ou seja, a tipagem dinâmica de variáveis, que permite que uma variável tenha seu tipo alterado, por exemplo, de *String*<sup>20</sup> para número a qualquer momento no código.

Segundo CAVALCANTE (2016) “A variável é uma entidade que possui um conjunto de atributos como nome, tipo e valor. Por exemplo, se imaginarmos a memória de um computador como sendo uma estante de um depósito, em que cada compartimento representa um espaço de memória, podemos também imaginar uma variável como sendo uma caixa que pode ser armazenada nessa estante”.

Segundo a mesma autora podemos definir tipo como “... as especificações de quais informações podem ser guardadas na caixa, ou em outras palavras, o tipo e tamanho do material que pode ser armazenado nela”.

Para LISBOA (2011) “a tipagem dinâmica ocorre quando a linguagem não obriga a prévia declaração de tipo de uma variável. O tipo é assumido na atribuição de valor à variável, que pode ser por presunção”.

### 5.2.3 Operações Matemáticas

Neste nível do ambiente são apresentadas ao aluno as operações matemáticas básicas e fundamentais disponíveis na linguagem de programação Lua: soma, subtração, multiplicação e divisão.

---

20 Em programação de computadores, uma cadeia de caracteres ou *string* é uma sequência de caracteres, geralmente utilizados para representar palavras, frases ou textos.

A opção pelo desenvolvimento de um nível dedicado às operações matemáticas deve-se a importância que a matemática tem sobre a programação, uma vez que a matemática é aplicada nos mais diversos tipos de software (CRUZ, 2014).

#### 5.2.4 Tabelas

Neste nível do ambiente o aluno é introduzido a conceitos básicos e o aprendizado sobre o uso de matrizes e vetores na linguagem de programação Lua.

Segundo PINHO (1999), “Matrizes e vetores podem ser definidos como coleções de variáveis relacionadas, armazenadas continuamente na memória e que podem ser acessadas através de um mesmo nome de variável. O que diferencia uma matriz de um vetor é a sua quantidade de dimensões, matrizes são bidimensionais enquanto vetores são unidimensionais”.

#### 5.2.5 Cláusulas Condicionais

Neste nível do ambiente o aluno é apresentado ao conceito sobre estruturas de controle, abordando desde condições simples até condições de encadeamento. Segundo MANSSOUR (1999), “Os comandos de fluxo de controle de uma linguagem de programação especificam a ordem em que processamento é feito”.

Segundo a mesma autora estes comandos, como o próprio nome diz, permitem fazer uma seleção, a partir de uma ou mais alternativas, da ação que o computador deve tomar.

#### 5.2.6 Estruturas de Repetição

Neste nível do ambiente, o aluno é apresentado ao aprendizado do uso das estruturas básicas de repetição disponíveis na linguagem de programação Lua sendo elas: “enquanto” (estrutura de repetição com verificação antes do laço), “repita-até” (estrutura de repetição com verificação após laço) e “para” (estrutura de repetição controlada por variável).

Para MANSSOUR (1999), “os comandos de iteração (ou repetição) especificam a execução de loops, isto é, fazem com que os outros comandos que eles controlam sejam executados zero ou mais vezes”.

### 5.3 Implementação do Ambiente

Por tratar-se de um ambiente desenvolvido para *web*, as linguagens utilizadas em sua implementação envolvem tecnologias *HTML*, *CSS*, *JavaScript*. Além disso, a linguagem *JSON* foi utilizada como apoio por oferecer estruturas úteis ao ambiente mantendo dados como os textos *strings* do jogo, os dados do jogador e das fases.

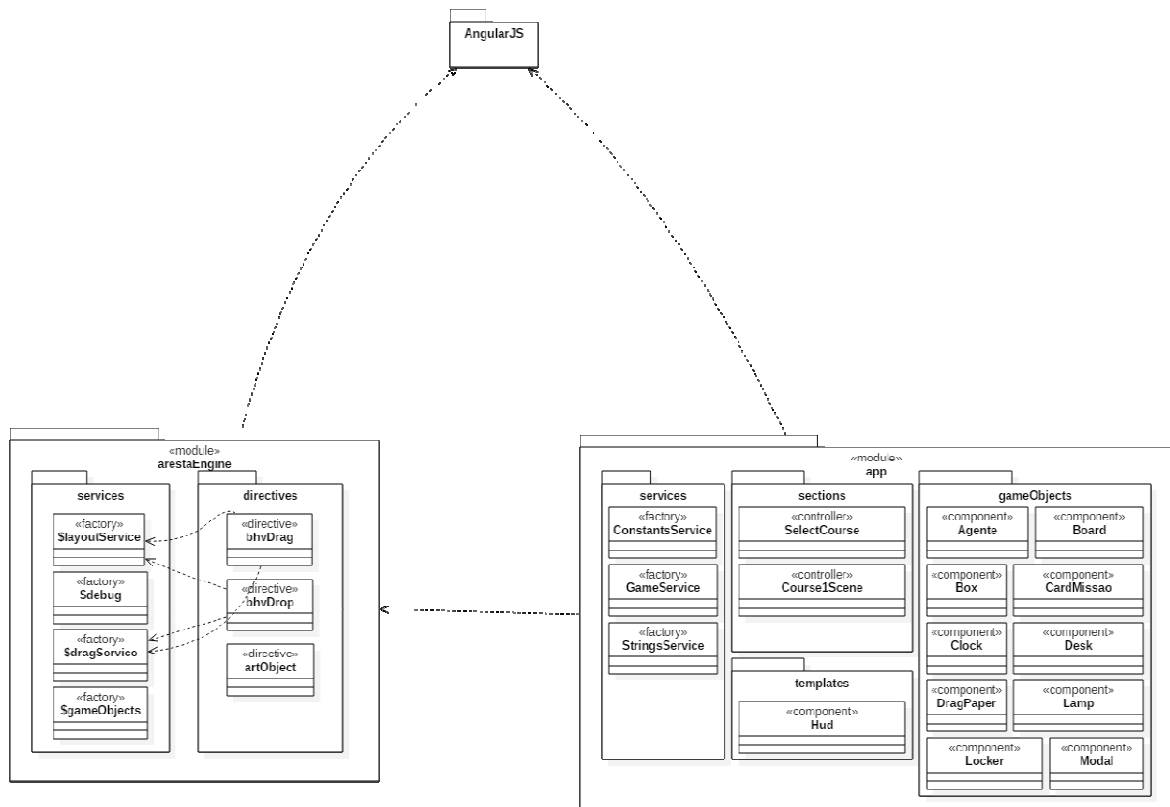
Os seguintes *frameworks* utilizados foram escolhidos por atender a complexidade da aplicação:

O *AngularJS* (figura 20) é utilizado para impor um padrão estrutural MVC (Model, View, Controller) ou MVVM (Model-View, View-Model) ao projeto. O *framework* também oferece conceitos modularizados, como as *Directives* que são utilizadas no reuso dos objetos do jogo. Oferecendo também módulos para animações básicas, como o *ngFx*, integrado com a biblioteca *JQuery*.

Por meio de requisições *ajax*, os logs do ambiente são devidamente salvos em um banco de dados *PostgreSql*. Podendo estes mesmos serem acessados e gerenciados utilizando-se de páginas *PHP* disponibilizadas em um servidor Apache.

O editor de código *Brackets* foi escolhido para a implementação por apresentar um *workflow* rápido e consistente. Além de trata-se de uma ferramenta do cunho de software livre, *Brackets* oferece *plug-ins* compatíveis com as tecnologias escolhidas, e também o *Live Preview*, que facilita a edição de elementos, atualizando a página instantaneamente via *ajax*.

Figura 20: *Framework* estrutural MVC



Fonte: Elaborado pelo Autor, Canoas: 2016

## 5.4 Interfaces do Ambiente

A seguir são descritas brevemente as telas que compõem as interfaces do ambiente.

Para definir a compatibilidade do ambiente com as mais variadas versões dos mais recentes e principais navegadores web disponíveis no mercado, todos os arquivos HTML e CSS produzidos para este ambiente gamificado foram avaliados e aprovados pelas ferramentas de validação da World Wide Web Consortium<sup>21, 22</sup>, (W3C).

A tela de *login* (Figura 21) é a primeira tela a ser apresentada. Neste momento o aluno informa seu número de matrícula e senha para ser validado no ambiente e tão somente após esta validação o mesmo é direcionado para a tela inicial do jogo (Figura 22). Tal fato deve-se a sua simplicidade em termos de função, pois apenas levará o jogador às demais telas do jogo.

<sup>21</sup> A ferramenta de validação e marcação da W3C esta acessível em <http://validator.w3.org>

<sup>22</sup> A ferramenta de validação de estilos da W3C esta acessível em <http://jigsaw.w3.org/css-validator/>

**Figura 21: Tela Inicial de *login***

INSTITUTO FEDERAL  
Rio Grande do Sul | Campus  
Canoas

Entre com o Login e senha

Usuário

Senha

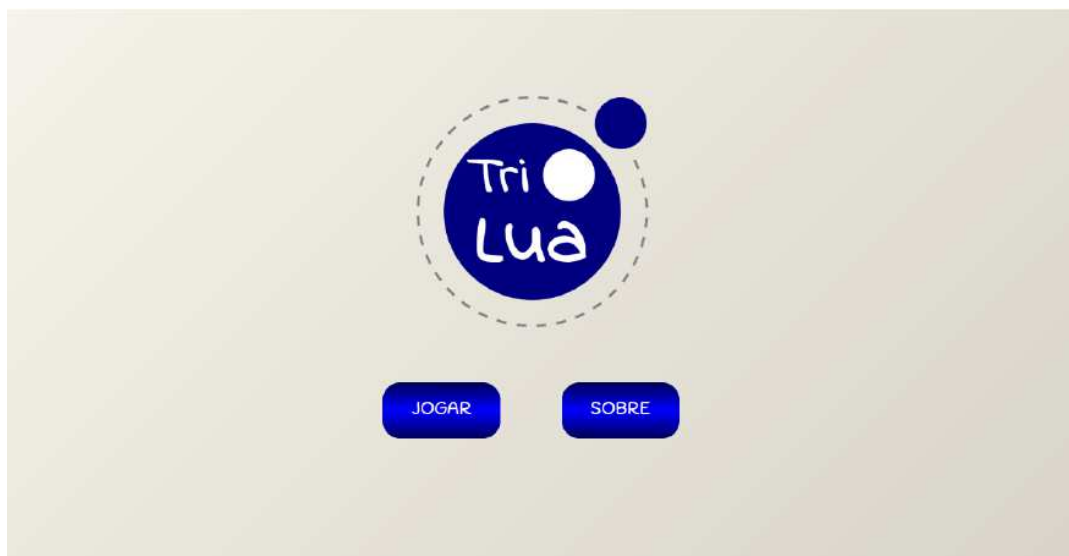
Lembrar de mim

logar

[Esqueci a senha](#)

**Fonte: Elaborado pelo Autor, Canoas: 2016**

**Figura 22: Tela Inicial**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

#### 5.4.1 Telas de Navegação

As telas de navegação incluem todas as telas entre os menus de seleção de modo de execução, nível e também as telas da seção de informações gerais sobre o jogo. Estas telas são compostas pelos seguintes arquivos:

- a) um arquivo HTML utilizado para definir elementos que deverão ser exibidos exclusivamente na tela que estiver aberta;
- b) um componente *JavaScript* o qual carrega *tags* HTML comuns à todas páginas de navegação, como por exemplo o menu, o qual está ilustrado na figura 23;

c) dois arquivos CSS que carregam as *tags* de estilo referentes a elementos comuns à todas páginas de navegação, enquanto o outro carrega *tags* de estilo que são utilizadas unicamente na tela que estiver aberta.

**Figura 23: Tela de navegação**



Fonte: Elaborado pelo Autor, Canoas: 2016

#### 5.4.2 Telas de Nível do Ambiente

As telas de níveis do ambiente incluem todos os níveis jogáveis. Estas telas (figura 24) são compostas por cinco arquivos, sendo estes um arquivo *Javascript* e um arquivo *CSS* comuns à todas as telas de níveis, pois carregam as *tags* de *HTML* e estilo comuns a todos os níveis, além de um arquivo *HTML*, um arquivo *CSS* e um arquivo *JavaScript* próprios.

Enquanto os arquivos de *HTML* e *CSS* servem para designar quais elementos e como os mesmos devem ser exibidos unicamente no nível que estiver sendo executado, o arquivo *JavaScript* tem como função controlar a jogabilidade do nível e validar as ações do usuário.

**Figura 24: Tela de nível da unidade variável**



Fonte: Elaborado pelo Autor, Canoas: 2016



## 5.5 Componentes dos Níveis e Mecânica do Ambiente

Todos os níveis do ambiente possuem um conjunto padrão de componentes de tela (figura 25) juntamente com uma tela do ambiente, onde o aluno deverá executar as ações propostas.

**Figura 25: Tela com os detalhamentos dos menus usados nas telas de níveis**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

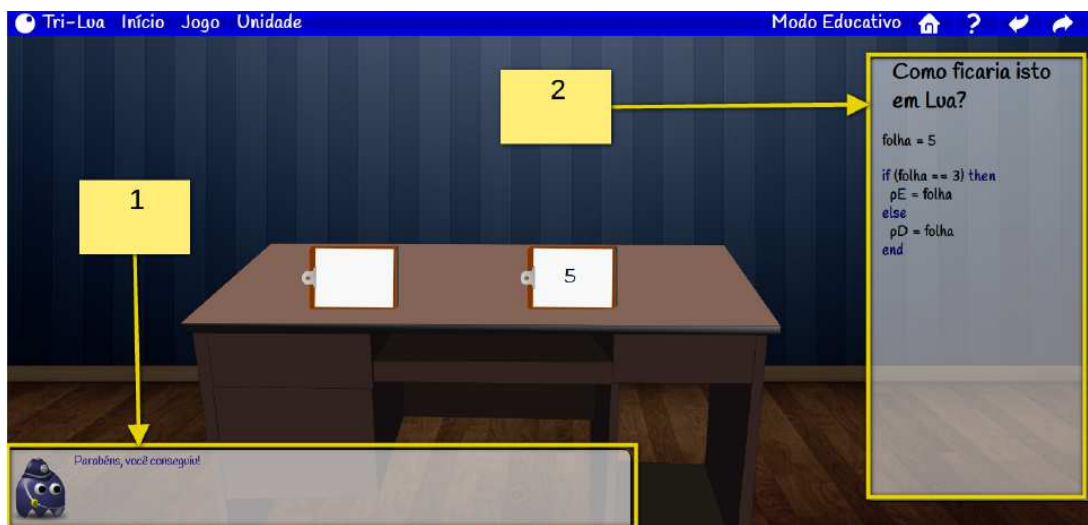
Podem ser observados na figura 25 os seguintes itens:

1. Atalho para a página inicial do jogo;
2. Atalho para a página de seleção de modo de jogo;
3. Atalho para a página de seleção de unidade;
4. Atalho para a página de seleção de níveis;
5. Acesso ao tutorial rápido;
6. Atalho para retroceder ao nível anterior;
7. Atalho para avançar para o próximo nível (apenas disponível após o aluno concluir o nível atual).

Os demais itens do menu, não relacionados, possuem caráter unicamente estético. Quando o aluno clica no botão de acesso ao tutorial rápido, o jogo exibe uma *pop-up* contendo um pequeno texto de ajuda referente ao conteúdo do nível que o aluno estiver jogando.

Além do menu principal, os níveis são compostos por outros dois itens fixos, a caixa de exibição do agente animado e a caixa de exibição do algoritmo em Lua (Figura 26).

**Figura 26: Tela com os detalhamentos dos demais itens presentes em todos os níveis do ambiente**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A caixa do agente animado é composta pela figura do agente animado propriamente dito, além de um espaço para mensagens de texto, que são preenchidas com informações fornecidas pelo agente, que interage com o aluno lhe explicando os conceitos da linguagem de programação Lua, apresentando os desafios do nível e comentando seu desempenho, parabenizando-o por concluir um desafio corretamente, ou motivando-o caso faça algo errado.

A caixa de exibição do algoritmo é apresentada a cada vez que o aluno conclui um nível com sucesso, apresentando o código em linguagem de programação Lua referente ao exercício lúdico concluído pelo aluno. Nesta caixa também são exibidos comentários sobre o algoritmo e curiosidades sobre a linguagem de programação Lua.

Todos os níveis do ambiente estão relacionados a tarefas que envolvem clicar em um objeto para selecioná-lo e então clicar em outra posição para "soltar" o objeto selecionado.

A interface gráfica do ambiente através das telas de navegação foi concebida para que todas as telas seguissem um mesmo padrão de botões e cores, evitando assim que o aluno tenha de "reaprender" a usar o ambiente a cada novo nível liberado.

Praticamente todo o layout do jogo foi construído utilizando CSS, com exceção das *pop-ups*<sup>23</sup>, as quais foram construídas com recursos da biblioteca jQuery<sup>24</sup>. As artes imagens e figuras utilizadas no jogo estão sob a licença de uso de tipo domínio público<sup>25</sup>, retiradas de portais como Open ClipArt<sup>26</sup> e Clker<sup>27</sup>.

<sup>23</sup> Um *POP-UP* é uma janela que abre no navegador ao visitar uma página web ou acessar um hiperlink específico. O *pop-up* é utilizado por criadores de site para abrir alguma informação extra.

<sup>24</sup> jQuery é uma excelente e leve biblioteca para desenvolvimento rápido de javascripts que interagem com sua página html,.

<sup>25</sup> São obras de livre uso, pois não estão sob domínio de direito autoral. Os direitos autorais duram até 70 anos contando do primeiro ano de falecimento do autor.

<sup>26</sup> Acessível em: <https://openclipart.org/>

<sup>27</sup> Acessível em: <http://www.clker.com/>

## 5.6 Módulo de Mineração de Dados Educacionais

O objetivo deste módulo foi o desenvolvimento de um processo que possibilitasse a automatização de descoberta do conhecimento gerado pelos comportamentos dos alunos a partir de dados gerados no ambiente gamificado intitulado Tri-Lua.

A intenção foi a de produzir um modelo simples que possa ter suas funções ampliadas no futuro. O protótipo irá se utilizar da base de dados gerada pelos alunos através do ambiente gamificado, a qual contém todos os dados referentes às ações desempenhadas pelos alunos durante a sessão de uso dos módulos do ambiente, e ao final do processo será gerado então um relatório no formato pdf, contendo informações úteis ao professor.

De um modo geral, o processo considerado neste modelo é descrito na tela a seguir, na figura 27 definindo os seguintes campos:

- Período: O professor irá informar o período que o mesmo deseja ter acesso através dos relatórios de acompanhamento de desempenho do aluno;
- Aluno: O professor irá informar um aluno específico através de parte de seu nome ou caso o campo seja deixado em branco o relatório irá selecionar todos os alunos que participaram da oficina neste período.

**Figura 27: Tela módulo do professor**

**Fonte: Elaborado pelo Autor, Canoas: 2016**

Este processo não considera uma intervenção junto ao ambiente, mas apenas tratará dos dados gerados pelo uso do ambiente, que estarão disponíveis para manipulação através das técnicas de mineração de dados educacionais descritas no item 2.5.

No processo previsto, será realizada a extração dos dados brutos da base de dados gerada pelo ambiente e após este passo os dados são lidos, organizados e analisados por algoritmos de mineração de dados educacionais, com uso da ferramenta WEKA, sendo que então o arquivo final com as informações obtidas será fornecido ao professor através de um arquivo em formato pdf.

Todo este processor será descrito mais detalhadamente a seguir.

### 5.6.1 Organização dos Dados

Os dados gerados pelo ambiente podem conter diferentes tipos e quantidade de informações, portanto, o primeiro passo a ser tomado é realizar a sua análise de acordo com os objetivos previstos para o apoio aos professores e de acordo com os algoritmos a serem usados.

Após essa análise inicial, foi definido o seguinte conjunto de elementos como sendo de interesse para a mineração de dados: nome do usuário; identificação da missão; número de erros. O elemento “usuário” exibe o nome do aluno que desempenhou a atividade. O elemento “missão” se refere ao nome da missão desempenhado pelo aluno. O elemento seguinte se refere ao número total de erros que o aluno comentou ao desempenhar a missão. A figura 28 mostra como as informações organizadas no registro gerado.

**Figura 28: Exemplo de registro gerado pelo ambiente gamificado**

Aluno1	Tutorial	4
Aluno1	Variáveis #1	0
Aluno1	Variáveis #2	4
Aluno1	Variáveis#3	1
Aluno1	Variáveis #4	0
Aluno1	Variáveis#5	4
Aluno1	Variáveis#6	6
Aluno1	Variáveis#7	6
Aluno1	Operadores#1	6
Aluno1	Operadores#2	9
Aluno1	Operadores#3	6
Aluno1	Operadores#4	9
Aluno1	Operadores#5	9
Aluno1	Tabelas_Vetor#1	8
Aluno1	Tabelas_Vetor#2	4
Aluno1	Tabelas_Vetor#3	7

**Fonte: Elaborado pelo Autor, Canoas: 2016**

Uma vez que a base de dados está de acordo com o que o aplicativo irá processar, inicia-se o processo de organização dos dados. Essa organização ocorre inicialmente separando as informações referentes a cada usuário. O algoritmo criará uma matriz, onde o primeiro índice de cada linha corresponde ao nome do usuário e os demais índices aos erros gerados em cada missão. Também será criado um vetor onde posteriormente serão armazenados os índices de aproveitamento de cada missão.

A cada linha lida pelo algoritmo, este irá verificar se o aluno já está contido na matriz e caso esteja contido, o atributo referente aos erros gerados em cada missão será armazenado de acordo com o primeiro índice da linha que corresponde a este aluno. Se o aluno ainda não estiver contido na matriz, será adicionado à primeira linha vazia e logo após será adicionado o atributo referente aos erros gerados em cada missão ao primeiro índice nulo da linha. Ao

terminar a leitura da base de dados, o algoritmo iniciará a conversão do número de erros de cada missão em índices de aproveitamento (ruim, bom, muito bom e excelente) de cada usuário e então fará a inserção no vetor. A classificação dos erros se dará da seguinte forma:

- Número de erros acima de zero e abaixo de 1, classificação como Excelente;
- Número de erros acima de 1 e abaixo de 3, classificação como muito bom;
- Número de erros acima de 3 e abaixo de 5; classificação como bom;
- Número de erros acima de 5 erros; classificação como ruim;

Após esta etapa todos os dados estarão organizados e a partir daí será criado um arquivo no formato csv. O arquivo no formato csv será usado como entrada na etapa de mineração de dados. Os dados que estão classificados pelos índices de aproveitamento serão utilizados para a execução do algoritmo *Apriori*, enquanto que o número de erros será usado pelo algoritmo *Simple KMeans*. Após a criação deste arquivo, será possível utilizar os algoritmos de mineração para a obtenção de informações úteis ao professor.

### 5.6.2 Mineração de Dados

Com o arquivo csv devidamente estruturado, inicia-se a etapa de processamento e mineração de dados propriamente dita. A ferramenta WEKA realiza a análise dos dados de várias formas diferentes, que podem ser feitas de acordo com o tipo de informação que se pretende obter. Nesta implementação serão usados os algoritmos *Simple K-Means* e *Apriori*.

Segundo (CAMILO; SILVA, 2009), *K-Means* é o algoritmo que usa o conceito da centroide. Dado um conjunto de dados, o algoritmo seleciona de forma aleatória  $k$  registros, cada um representando um agrupamento. Para cada registro restante, é calculada a similaridade entre o registro analisado e o centro de cada agrupamento. O objeto é inserido no agrupamento com a maior similaridade. O centro do cluster é recalculado a cada novo elemento inserido. Diferentes variações surgiram: implementando otimizações para escolha do valor do  $k$ , novas medidas de dissimilaridade e estratégias para o cálculo do centro do agrupamento.

Através do uso do algoritmo *Simple K-Means* neste projeto de avaliação preliminar de possibilidades, buscou-se classificar os alunos em três grupos, de acordo com os erros cometidos em cada missão. Deste modo, foi possível separar os que tiveram bom desempenho, os que apresentaram desempenho mediano, e os que tiveram um desempenho ruim. Com esses dados, o professor pode identificar quem são os alunos que estão com dificuldade em assimilar o conteúdo apresentado e necessitam alguma atenção especial.

Segundo AGRAWAL; IMIELINSKI; SWAMI, (1993) *Apriori* consolidou-se como o primeiro algoritmo de mineração de regras de associação assumido como eficiente. Esse algoritmo combina uma estratégia de busca denominada *Breadth-first search* (BFS) com uma estrutura de árvore para contagem de ocorrência de candidatos.

O objetivo no uso do algoritmo *Apriori* no protótipo foi o de encontrar algumas dificuldades ou tendências no nível de grupo. Foi possível, por exemplo, identificar que os alunos em geral possuem dificuldade nas atividades que envolvem matrizes e vetores, caso uma regra indique que os erros estão acontecendo neste tipo de atividade.

Com o uso destes algoritmos foi possível obter diversos tipos de informações relacionadas com o desempenho dos alunos, bem como a obtenção de padrões nas tomadas de decisão.

### 5.6.3 Relatório Final

Depois do processo de mineração de dados, foi necessário selecionar as principais informações, ou seja, as informações que serão fornecidas ao professor.

Primeiramente serão mostradas as informações obtidas a partir do *Simple K-Means*. Entre elas estarão a média de erros contidos em cada *cluster* e uma lista com o nome e o número de erros de cada aluno, além de uma avaliação de desempenho que está relacionada ao *cluster* em que cada aluno foi incluído. Com estas informações foi possível descobrir qual grupo de alunos requer uma atenção especial de um modo geral. Na sequência foi apresentado um conjunto de regras de associação geradas pelo algoritmo *Apriori*. Estas regras podem estar associadas com tipos de atividades ou assuntos em comum e desta forma o professor poderá reforçar apenas essa habilidade.

## 6 AVALIAÇÕES

Neste capítulo será apresentada uma visão geral das avaliações realizadas junto aos alunos e professores atuantes no Curso Técnico em Desenvolvimento de Sistemas Integrado ao Ensino Médio. Inicialmente foi realizada uma prova escrita com os 40 alunos regularmente matriculados na disciplina de Lógica de Programação, que abordava conceitos básicos de lógica de programação. Após a correção da mesma a turma foi dividida em dois grupos sendo que um deles designado grupo de controle participou de uma oficina onde foi apresentado o ambiente gamificado desenvolvido. Nesta oficina foram armazenadas de forma implícita informações sobre a interação dos alunos com o ambiente de maneira a subsidiar posteriormente a mineração de dados educacionais.

A partir do tratamento dos dados brutos gerados pela interação dos alunos junto ao ambiente, aplicou-se técnicas de mineração de dados educacionais que geraram informações pertinentes em forma de relatório que foi apresentado ao grupo de professores das disciplinas de lógica de programação. Estes docentes avaliaram, assim, a validade do ambiente gamificado para o ensino e a aprendizagem de conceitos inerentes à lógica de programação.

Por fim, aplicou-se uma nova prova escrita junto aos 40 alunos da disciplina de Lógica de Programação que objetivou verificar a melhora no aprendizado dos conceitos da disciplina junto ao grupo de controle e junto aos alunos que não tiveram acesso ao ambiente gamificado.

Este capítulo descreve os resultados destas avaliações.

### 6.1 Avaliações com os alunos

Como já mencionado, os 40 alunos do primeiro ano do Curso Técnico em Desenvolvimento de Sistemas integrado ao ensino médio do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) - Campus Canoas realizaram uma prova de conceitos básicos de lógica de programação. Nesta avaliação os 40 alunos obtiveram as notas demonstradas na Tabela 5.

**Tabela 5: Resultado da primeira avaliação na disciplina de lógica de programação**

Identificação do Aluno	Prova 1
ALUNO 1	0,4
ALUNO 2	6,8
ALUNO 3	2,8
ALUNO 4	5,8
ALUNO 5	6,5
ALUNO 6	8,3
ALUNO 7	7,4
ALUNO 8	0,0
ALUNO 9	9,3
ALUNO 10	9,1
ALUNO 11	2,9
ALUNO 12	7,5
ALUNO 13	7,5
ALUNO 14	4,4

ALUNO 15	7,6
ALUNO 16	6,5
ALUNO 17	2,7
ALUNO 18	5,5
ALUNO 19	9,3
ALUNO 20	5,8
ALUNO 21	2,0
ALUNO 22	1,2
ALUNO 23	5,5
ALUNO 24	3,2
ALUNO 25	5,3
ALUNO 26	4,2
ALUNO 27	0,5
ALUNO 28	9,5
ALUNO 29	3,6
ALUNO 30	7,6
ALUNO 31	7,6
ALUNO 32	8,4
ALUNO 33	7,8
ALUNO 34	1,0
ALUNO 35	7,5
ALUNO 36	8,2
ALUNO 37	6,2
ALUNO 38	7,8
ALUNO 39	7,4
ALUNO 40	8,2

**Fonte: Elaborado pelo Autor, Canoas: 2016**

Após esta avaliação os alunos foram separados em dois grupos de 20 alunos, sendo um deles considerado o grupo de controle. Para compor o grupo de controle foram disponibilizadas 20 vagas para a oficina de gamificação. A adesão dos alunos aconteceu de forma voluntária. O grupo de controle realizou uma oficina inserida no projeto **Gamificando o ensino de lógica de programação**, a fim de verificar as possibilidades e aceitação do ambiente implementado. Estes alunos têm idades entre 13 e 16 anos.

A atividade teve a duração de dois meses num total geral de 8 horas, sendo 1 hora por semana, nela foi proposto que os alunos utilizassem o ambiente gamificado em todos os seus níveis e ao final da atividade da última semana respondessem a um questionário de quatorze perguntas, onde os mesmos poderiam expor suas opiniões sobre o ambiente.

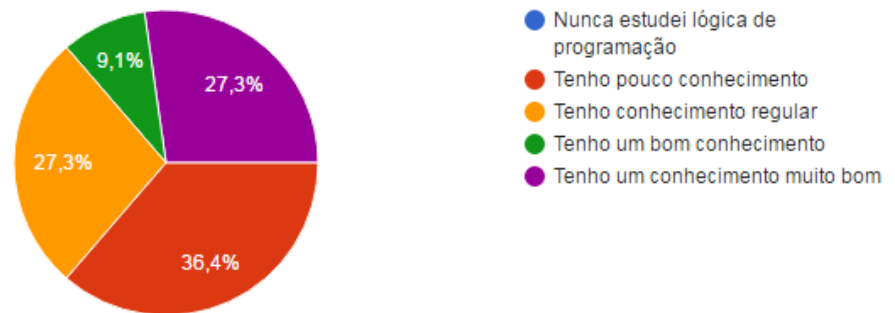
Ao final do período de atividades os alunos conseguiram completar todos os níveis do ambiente e responder o formulário dentro do tempo especificado, com duração média diária das atividades variando de 7 a 23 minutos.

A atividade de testes com os alunos foi completada por todos, dentro do tempo especificado, sem quaisquer necessidades de intervenção dos monitores. A partir das respostas obtidas no questionário, foram produzidos gráficos para organizar a exibição das opiniões dos voluntários no teste sobre o jogo.



A figura 29 ilustra graficamente os resultados obtidos para a pergunta “Como você considera seu conhecimento em relação à lógica de programação?”. Neste caso, a maioria dos alunos afirma ter pouco ou nenhum conhecimento sobre lógica de programação.

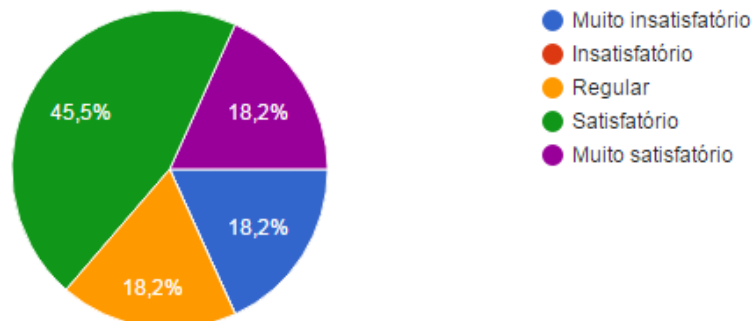
**Figura 29: Conhecimento dos alunos sobre lógica de programação.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 30 ilustra graficamente os resultados obtidos para a pergunta “Qual a sua opinião em relação ao visual e navegação do ambiente?”. A maioria dos alunos aponta que o visual e navegação são satisfatórias.

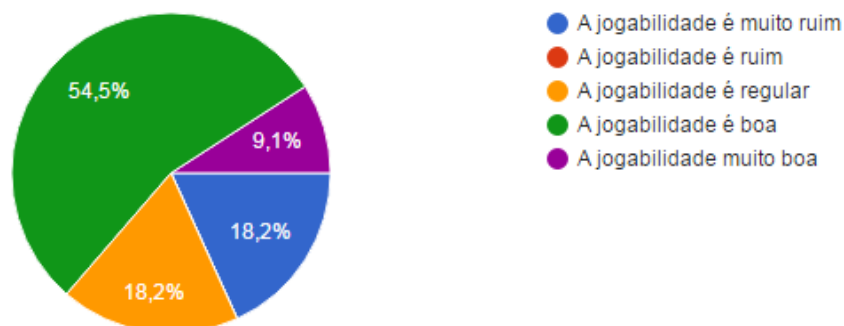
**Figura 30: Opinião dos alunos sobre a interface gráfica e navegação do ambiente.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 31 ilustra graficamente os resultados obtidos para a pergunta “Qual a sua opinião em relação à jogabilidade em termos de funcionalidade?”. Neste caso, a maioria dos alunos afirma que a jogabilidade é boa.

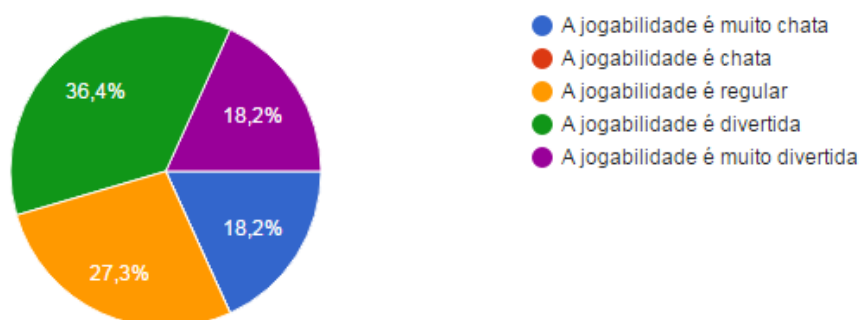
**Figura 31: Opinião dos alunos sobre a jogabilidade em termos de funcionalidade.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 32 ilustra graficamente os resultados obtidos para a pergunta “Qual a sua opinião em relação à jogabilidade em termos de diversão?”. A maioria dos alunos aponta que a jogabilidade é divertida ou regular em relação à diversão.

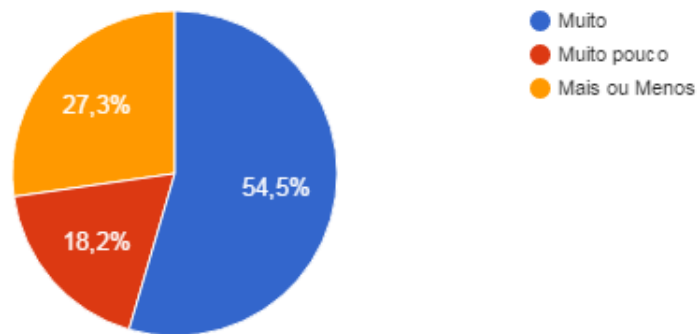
**Figura 32: Opinião dos alunos sobre a jogabilidade em termos de diversão.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 33 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião, o nível de variáveis contribuiu para ser aprendido na linguagem LUA?”. Nesta pergunta, a maioria dos alunos aponta que o nível de variáveis contribuiu muito para seu aprendizado na linguagem LUA.

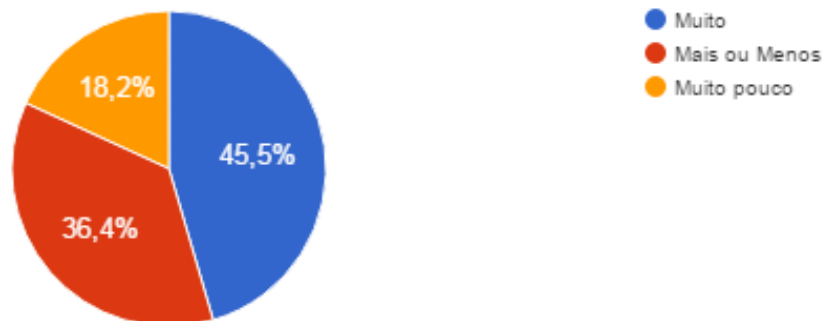
**Figura 33: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da manipulação de variáveis em Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 34 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião, o nível de operadores matemáticos básicos contribuiu para seu aprendizado na linguagem LUA?”. Nesta pergunta, a maioria dos alunos respondeu que contribuiu muito. Uma grande parcela afirma que contribuiu mais ou menos.

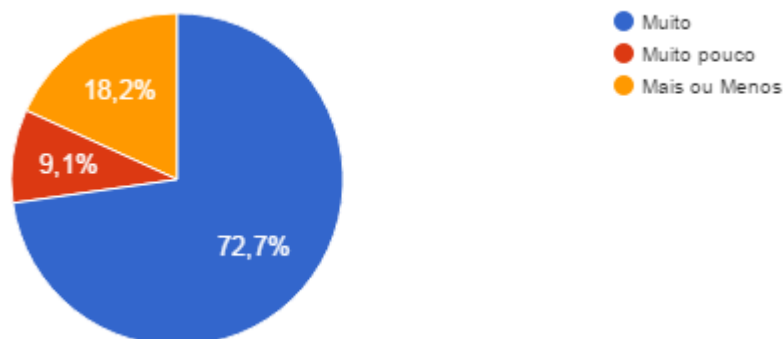
**Figura 34: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização de operadores matemáticos básicos em Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 35 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o nível de manipulação de vetores e matrizes contribuiu para seu aprendizado na linguagem em LUA?”. Neste caso, a maioria dos alunos afirma que este aspecto contribuiu muito para o seu aprendizado.

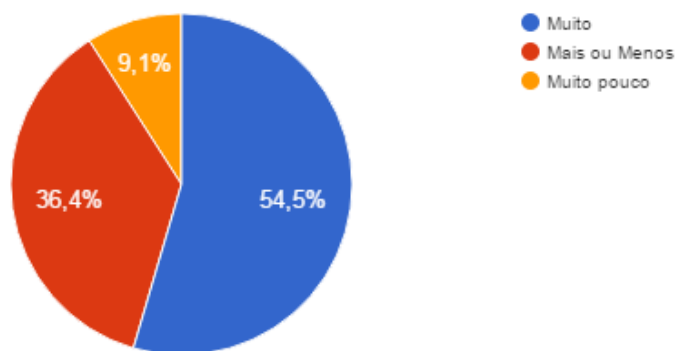
**Figura 35: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização de matrizes e vetores em Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 36 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o nível de utilização da cláusula *if* contribuiu para seu aprendizado na linguagem em LUA?”. Nesta pergunta, a maioria dos alunos afirma que este aspecto contribuiu muito para o seu aprendizado.

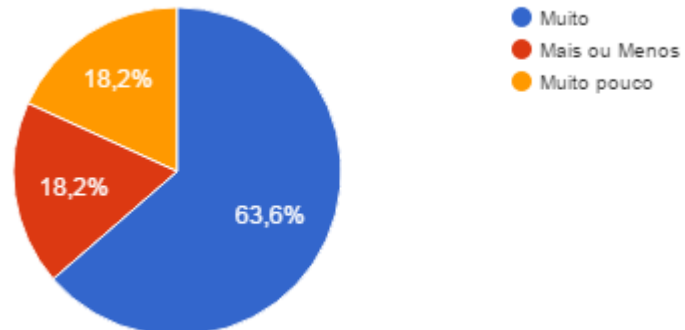
**Figura 36: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da cláusula *if* em Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 37 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o nível de utilização da estrutura *for* contribuiu para seu aprendizado na linguagem em LUA?”. Nesta pergunta, a maioria dos alunos afirma que este aspecto contribuiu muito para o seu aprendizado.

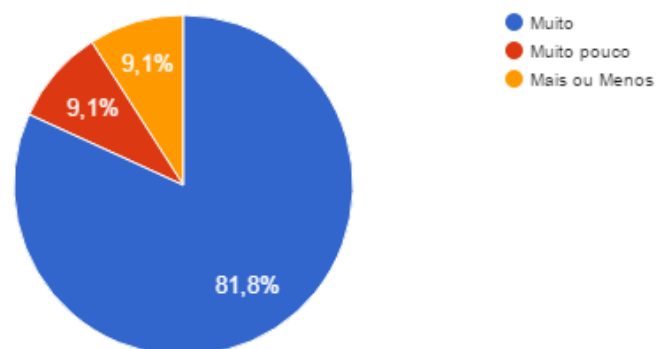
**Figura 37: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura *for* em Lua.**



Fonte: Elaborado pelo Autor, Canoas: 2016

A figura 38 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o nível de utilização da estrutura *repeat-until* contribuiu para seu aprendizado na linguagem em LUA?”. Neste caso, a ampla maioria dos alunos afirma que este aspecto contribuiu muito para o seu aprendizado.

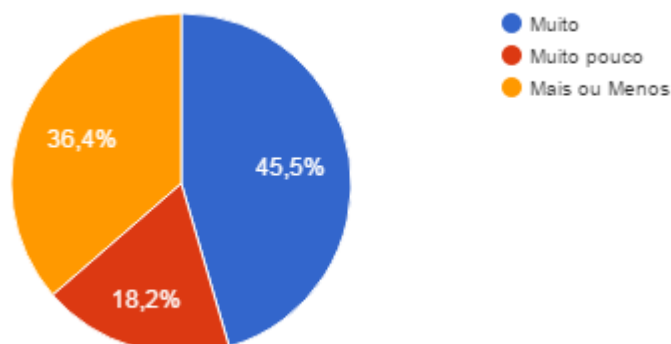
**Figura 38: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura *repeat-until* em Lua.**



Fonte: Elaborado pelo Autor, Canoas: 2016

A figura 39 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o nível de utilização da estrutura *while* contribuiu para seu aprendizado na linguagem em LUA?”. Neste caso, a maioria dos alunos afirma que este aspecto contribuiu muito para o seu aprendizado, mas uma parte considerável dos alunos, afirma que contribuiu mais ou menos.

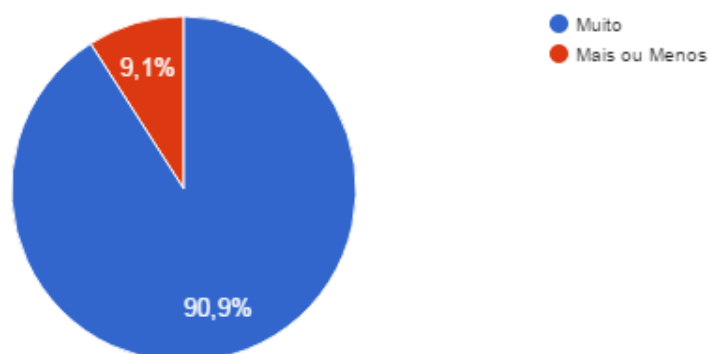
**Figura 39: Quanto os alunos consideram que o ambiente contribuiu para o aprendizado da utilização da estrutura *while* em Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 40 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião o ambiente gamificado motivou e contribuiu para seu aprendizado na linguagem em LUA?”. A ampla maioria dos alunos afirma que o ambiente gamificado contribuiu muito para o aprendizado da linguagem LUA.

**Figura 40: Quanto os alunos consideram que o ambiente os motivou no aprendizado da linguagem Lua.**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

Depois das ações acima descritas, os 40 alunos realizaram uma nova avaliação na disciplina de Lógica de Programação que visava verificar a aprendizagem dos conceitos básicos da disciplina.

A Tabela 6 demonstra um comparativo das notas obtidas pelos alunos nas duas avaliações. Os alunos que compõem o grupo de controle aparecem em negrito.

**Tabela 6: Comparativo das avaliações na disciplina de lógica de programação**

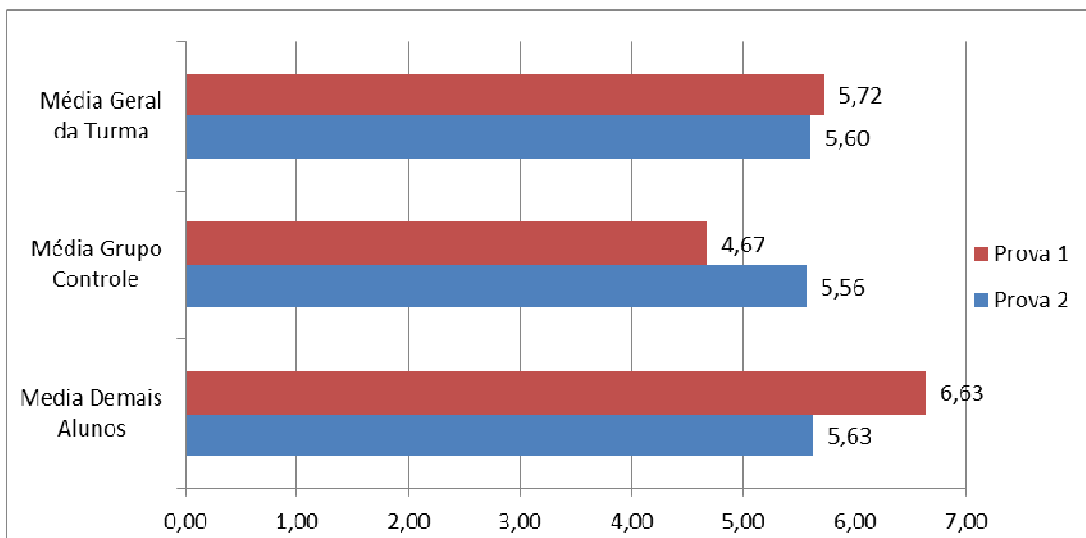
Identificação do Aluno	Prova 1	Prova 2
<b>ALUNO 1</b>	<b>0,4</b>	<b>1,3</b>
ALUNO 2	6,8	4,0
ALUNO 3	2,8	2,0
<b>ALUNO 4</b>	<b>5,8</b>	<b>8,9</b>
<b>ALUNO 5</b>	<b>6,5</b>	<b>5,0</b>
ALUNO 6	8,3	9,5
ALUNO 7	7,4	4,8
ALUNO 8	0,0	0,0
ALUNO 9	9,3	8,0
ALUNO 10	9,1	9,8
<b>ALUNO 11</b>	<b>2,9</b>	<b>3,5</b>
<b>ALUNO 12</b>	<b>7,5</b>	<b>7,5</b>
ALUNO 13	7,5	4,0
<b>ALUNO 14</b>	<b>4,4</b>	<b>5,7</b>
ALUNO 15	7,6	6,8
<b>ALUNO 16</b>	<b>6,5</b>	<b>4,8</b>
<b>ALUNO 17</b>	<b>2,7</b>	<b>7,3</b>
ALUNO 18	5,5	4,0
ALUNO 19	9,3	5,8
<b>ALUNO 20</b>	<b>5,8</b>	<b>6,5</b>
<b>ALUNO 21</b>	<b>2,0</b>	<b>3,2</b>
<b>ALUNO 22</b>	<b>1,2</b>	<b>1,5</b>
<b>ALUNO 23</b>	<b>5,5</b>	<b>4,6</b>
<b>ALUNO 24</b>	<b>3,2</b>	<b>6,7</b>
<b>ALUNO 25</b>	<b>5,3</b>	<b>7,3</b>
ALUNO 26	4,2	3,1
<b>ALUNO 27</b>	<b>0,5</b>	<b>0,5</b>
ALUNO 28	9,5	8,5
<b>ALUNO 29</b>	<b>3,6</b>	<b>5,8</b>
ALUNO 30	7,6	9,3
<b>ALUNO 31</b>	<b>7,6</b>	<b>7,6</b>
<b>ALUNO 32</b>	<b>8,4</b>	<b>10,0</b>
ALUNO 33	7,8	5,0
ALUNO 34	1,0	0,8
ALUNO 35	7,5	5,9
ALUNO 36	8,2	6,7
<b>ALUNO 37</b>	<b>6,2</b>	<b>5,5</b>
ALUNO 38	7,8	7,3
<b>ALUNO 39</b>	<b>7,4</b>	<b>8,0</b>
ALUNO 40	8,2	7,3

Fonte: Elaborado pelo Autor, Canoas: 2016

Observa-se, na análise do comparativo das avaliações que a maioria dos alunos que compuseram o grupo de controle obteve uma melhora acima de 20% em seu desempenho na disciplina de Lógica de Programação.

A figura 41 demonstra a evolução dos alunos do grupo de controle nas avaliações em comparação aos demais alunos que não participaram das oficinas de gamificação. Observa-se que a média geral da turma, entre uma avaliação e outra, teve pequena queda. Já o grupo de controle apresentou uma melhora significativa enquanto houve queda na avaliação dos alunos que não realizaram as atividades das oficinas.

**Figura 41: Desempenho dos Alunos nas Avaliações em Lógica de Programação**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

Durante as oficinas foram obtidos dados para subsidiar a mineração de dados para os procedimentos apresentados no item que segue.

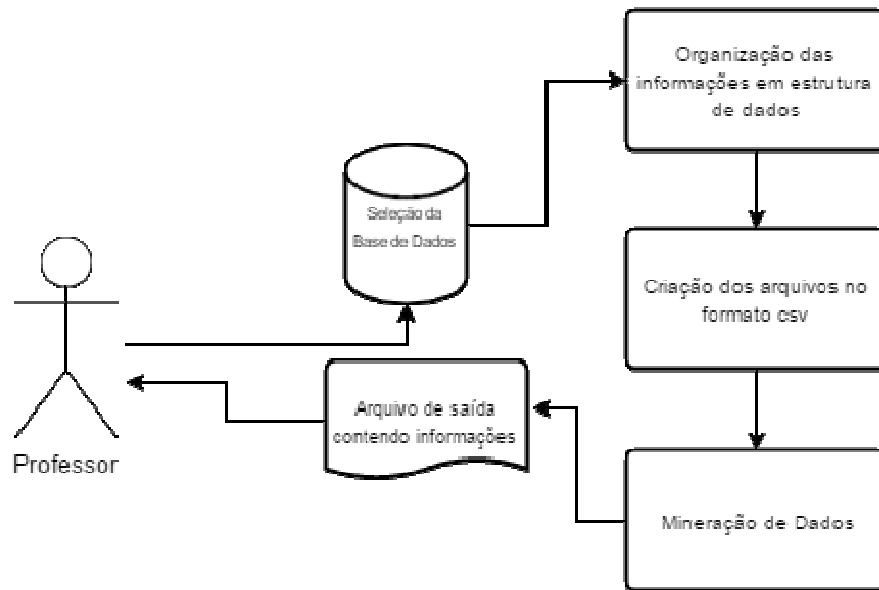
## 6.2 Avaliações com os professores

As oficinas de utilização do ambiente gamificado geraram dados brutos que foram coletados de forma implícita. Nestes dados aplicaram-se técnicas de mineração de dados educacionais que geraram informações pertinentes em forma de relatórios apresentados ao grupo de professores da disciplina de lógica de programação. Estes docentes avaliaram, assim, a validade do ambiente gamificado para o ensino e a aprendizagem de conceitos inerentes à lógica de programação.

De um modo geral, o processo realizado está descrito no diagrama representado na figura 42.



**Figura 42: Visão do processo realizado**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

O conjunto de dados brutos gerado para esta avaliação está organizado como exibido na figura 43.

**Figura 43: Base de dados brutos**

Aluno1,0,2,7,3,4,3,2,12,3,9,12,3,5,10,2,11,1,9,2,8,3,8,7,12,5,7  
 Aluno4,0,7,0,5,4,2,0,2,4,5,5,7,1,1,2,7,5,7,2,0,7,5,5,3,5,3  
 Aluno5,0,3,3,3,1,5,2,4,2,0,0,0,5,1,3,1,2,3,3,5,1,0,4,4,2,5  
 Aluno11,0,4,2,2,6,7,1,6,3,3,4,0,0,8,3,9,9,4,3,5,7,1,0,6,7,2  
 Aluno12,0,2,5,1,9,5,7,9,2,8,0,8,0,7,0,1,7,0,9,0,5,4,6,3,0,3  
 Aluno14,0,7,6,1,5,4,1,10,1,3,7,5,4,9,5,6,1,2,5,3,0,8,11,2,0,6  
 Aluno16,0,6,3,6,5,1,6,1,1,2,1,7,2,0,1,7,6,3,3,3,3,0,2,0,5,5  
 Aluno17,0,4,2,3,1,3,8,2,4,5,0,7,2,7,1,7,5,1,2,0,1,0,5,3,2,2  
 Aluno20,0,1,2,8,2,8,7,1,1,0,1,9,8,7,3,5,4,8,1,5,9,4,8,0,9,8  
 Aluno21,0,3,10,12,12,2,2,3,0,6,6,12,10,0,1,8,8,3,9,8,11,9,2,0,3,9  
 Aluno22,0,11,2,9,14,11,12,10,5,8,14,0,9,5,13,1,10,7,8,2,6,9,5,16,9,15  
 Aluno23,0,8,7,4,9,1,7,5,7,6,10,1,2,6,3,4,5,1,10,9,10,8,9,0,10,10  
 Aluno24,0,2,3,8,1,2,5,13,2,12,9,2,12,8,0,14,8,14,6,3,0,7,5,3,4,11  
 Aluno25,0,2,8,7,1,6,8,8,12,4,5,4,0,8,8,11,9,4,7,7,10,5,8,7,7,9  
 Aluno27,0,1,0,0,0,0,2,0,1,2,0,1,1,2,2,2,2,1,2,0,1,2,2,0,1  
 Aluno29,0,4,7,6,0,7,4,0,11,0,3,7,0,5,5,1,8,3,1,11,2,1,0,7,0,5  
 Aluno31,0,4,0,3,1,2,0,3,4,0,4,0,1,1,4,4,4,4,4,4,0,3,1,4,0,3  
 Aluno32,0,1,0,7,5,2,2,3,4,5,7,7,2,2,5,1,0,6,1,6,0,1,2,2,0,3  
 Aluno37,0,5,3,1,1,0,3,0,5,5,1,1,5,5,0,2,3,2,5,2,4,5,1,3,0,3  
 Aluno39,0,13,2,3,4,17,8,9,13,7,0,5,13,17,5,18,12,6,2,0,2,7,16,1,12,15

**Fonte: Elaborado pelo Autor, Canoas: 2016**

Seguindo o processo proposto pelo ambiente gamificado, cada registro foi composto por 27 elementos: aluno, fase<sup>28</sup>, fase2, fase3, fase4, fase5, fase6, fase7, fase8, fase9, fase10, fase11, fase12, fase13, fase14, fase15, fase16, fase17, fase18, fase19, fase20, fase21, fase22, fase23, fase24, fase25 e fase 26. Para os testes com a ferramenta foi produzido um arquivo em que todos os alunos nele contidos tivessem realizados as 26 fases do ambiente.

A seguir são comentados os resultados obtidos com estes testes.

O primeiro algoritmo utilizado foi o *Simple K-means*, que já foi abordado e detalhado anteriormente. Definiu-se como padrão da ferramenta que o número de *clusters* fosse três, pois a intenção era a de dividir os alunos de acordo com seu desempenho – excelente, bom e regular. A figura 44 foi gerada pela API do *Weka* e a partir dela foram decididas junto ao grupo de professores as informações que comporiam o relatório final.

**Figura 44: Relatório gerado pela API do Weka do algoritmo *Simple K-Means***

```

=== Run information ===
Scheme: clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A
"weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Relation: BD Mineracao Tri-Lua WEKA - Cluster-weka.filters.unsupervised.attribute.Remove-R1
Instances: 20
Attributes: 26
  Tutorial - Variáveis #1 - Variáveis #2 - Variáveis #3 - Variáveis #4 - Variáveis #5 - Variáveis #6 - Variáveis #7
  Operadores #1 - Operadores #2 - Operadores #3 - Operadores #4 - Operadores #5 - Vetores #1 - Vetores #2
  Vetores #3 - Matrizes #1 - Matrizes #2 - Matrizes #3 - Condicionais #1 - Condicionais #2 - Condicionais #3
  Condicionais #4 - Repetidores #1 - Repetidores #2 - Repetidores #3

Test mode: evaluate on training data

=== Clustering model (full training set) ===

kMeans
=====
Number of iterations: 3
Within cluster sum of squared errors: 27.003640112022715
Initial starting points (random):

Cluster 0: 0,2,8,7,1,6,8,8,12,4,5,4,0,8,8,11,9,4,7,7,10,5,8,7,7,9
Cluster 1: 0,2,7,3,4,3,2,12,3,9,12,3,5,10,2,11,1,9,2,8,3,8,7,12,5,7
Cluster 2: 0,4,7,6,0,7,4,0,11,0,3,7,0,5,5,1,8,3,1,11,2,1,0,7,0,5

Missing values globally replaced with mean/mode
Final cluster centroids:

```

---

<sup>28</sup> Cada uma destas fases corresponde a uma atividade proposto no ambiente gamificado.

Attribute	Full Data (20.0)	Cluster#		
		0 (7.0)	1 (3.0)	2 (10.0)
Tutorial	0	0	0	0
Variáveis #1	4.5	5.7143	3.6667	3.9
Variáveis #2	3.6	5.1429	5.3333	2
Variáveis #3	4.6	6.2857	4	3.6
Variáveis #4	4.25	7.2857	3.3333	2.4
Variáveis #5	4.4	7.1429	3	2.9
Variáveis #6	4.35	7.2857	2.6667	2.8
Variáveis #7	5.05	6.4286	11.6667	2.1
Operadores #1	4.25	5.7143	2	3.9
Operadores #2	4.5	5.5714	8	2.7
Operadores #3	4.45	5.1429	9.3333	2.5
Operadores #4	4.3	5.5714	3.3333	3.7
Operadores #5	4.1	6	7	1.9
Vetores #1	5.45	7.1429	9	3.2
Vetores #2	3.3	4.7143	2.3333	2.6
Vetores #3	6	6.8571	10.3333	4.1
Matrizes #1	5.45	7.8571	3.3333	4.4
Matrizes #2	4.45	4.1429	8.3333	3.5
Matrizes #3	4.2	6.5714	4.3333	2.5
Condicionais #1	4.15	4.4286	4.6667	3.8
Condicionais #2	4.05	7.5714	1	2.5
Condicionais #3	4.3	6.5714	7.6667	1.7
Condicionais #4	4.95	7.7143	7.6667	2.2
Repetidores #1	3.9	3.8571	5.6667	3.4
Repetidores #2	4	7.1429	3	2.1
Repetidores #3	6.25	9.8571	8	3.2

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

```
0      7 ( 35%)
1      3 ( 15%)
2     10 ( 50%)
```

**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 44 mostra o número de interações realizadas pela ferramenta e os resultados obtidos, com o número *cluster* e o número de alunos que compõem cada *cluster* junto da média geral do mesmo. Dos 20 alunos que compuseram o grupo de controle obteve-se o seguinte resultado:

- 10 alunos, correspondendo a 50% (cinquenta por cento) do total, tiveram um desempenho excelente na maior parte das fases.
- 7 alunos, correspondendo a 35% (trinta e cinco por cento) do total, tiveram um desempenho bom na maior parte das fases.

- 3 alunos, correspondendo a 15% (quinze por cento) do total, tiveram um desempenho regular na maior parte das fases.

O segundo algoritmo utilizado foi o *Apriori*, ao executar os métodos fornecidos pela *API do Weka*, foram obtidas as dez regras de maior confiança, como mostrado na figura 45. Esta figura também foi obtida a partir da *API do Weka*.

**Figura 45: Relatório gerado pela API do Weka do algoritmo *Aprioris***

```

=== Run information ===
Scheme: Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: min@apr
Instances: 20
Attributes: 25
    Variáveis #1 - Variáveis #2 - Variáveis #3 - Variáveis #4 - Variáveis #5 - Variáveis #6 - Variáveis #7 - Operadores #1 - Operadores #2 - Operadores #3
    Operadores #4 - Operadores #5 - Vetores #1 - Vetores #2 - Vetores #3 - Matrizes #1 - Matrizes #2 - Matrizes #3 - Condicionais #1 - Condicionais #2
    Condicionais #3 - Condicionais #4 - Repetidores #1 - Repetidores #2 - Repetidores #3
=== Associator model (full training set) ===
Apriori
=====
Minimum support: 0.8 (16 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 4
Generated sets of large itemsets:
Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 8
Size of set of large itemsets L(3): 1
Best rules found:

1. Variáveis #1=Incorreto 17 ==> Repetidores #3=Incorreto 17 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.85)
2. Matrizes #3=Incorreto 16 ==> Variáveis #1=Incorreto 16 <conf.(1)> lift:(1.18) lev:(0.12) [2] conv:(2.4)
3. Variáveis #2=Incorreto 16 ==> Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.8)
4. Variáveis #3=Incorreto 16 ==> Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.8)
5. Variáveis #5=Incorreto 16 ==> Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.8)
6. Matrizes #3=Incorreto 16 ==> Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.8)
7. Matrizes #3=Incorreto Repetidores #3=Incorreto 16 ==> Variáveis #1=Incorreto 16 <conf.(1)> lift:(1.18) lev:(0.12) [2] conv:(2.4)
8. Variáveis #1=Incorreto Matrizes #3=Incorreto 16 ==> Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.05) lev:(0.04) [0] conv:(0.8)
9. Matrizes #3=Incorreto 16 ==> Variáveis #1=Incorreto Repetidores #3=Incorreto 16 <conf.(1)> lift:(1.18) lev:(0.12) [2] conv:(2.4)
10. Variáveis #1=Incorreto 17 ==> Matrizes #3=Incorreto 16 <conf.(0.94)> lift:(1.18) lev:(0.12) [2] conv:(1.7)

```

**Fonte: Elaborado pelo Autor, Canoas: 2016**

Na regra 5, por exemplo identifica-se que alunos que erraram a fase Variáveis #5 também erraram a fase Repetidores#3. É possível notar que algumas regras envolvem mais de duas fases, um exemplo é a regra 9 onde identifica-se que alunos que erraram a fase Matrizes#3 erraram a fase Variáveis#1 e, também, erraram a fase Repetidores#3.

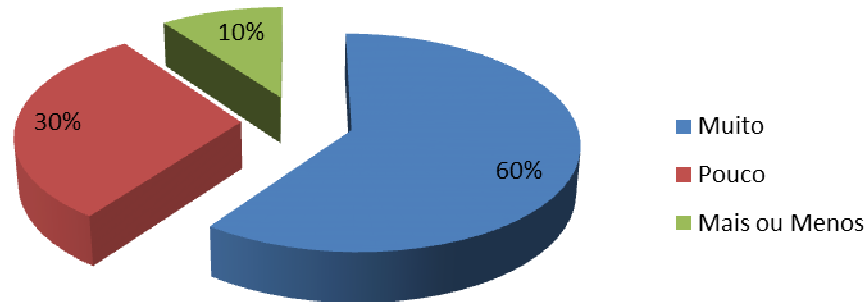
As regras criadas podem ser esclarecedoras pois trazem informações importantes sobre o desempenho dos alunos no conjunto de atividades do ambiente gamificado. É possível encontrar dificuldades específicas em um grupo, independente do desempenho individual de cada um dos alunos.

Os dados gerados pela *API do Weka* foram apresentados e discutidos com os docentes da disciplina de lógica de programação. Os docentes foram questionados a respeito da utilização de um ambiente gamificado para o ensino e aprendizagem de conceitos básicos de lógica de programação e as respostas aparecem abaixo.

A figura 46 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação amplia a motivação dos alunos?”. Neste caso, a maioria dos professores

afirma que a utilização do ambiente gamificado aumentou a motivação dos alunos para o aprendizado.

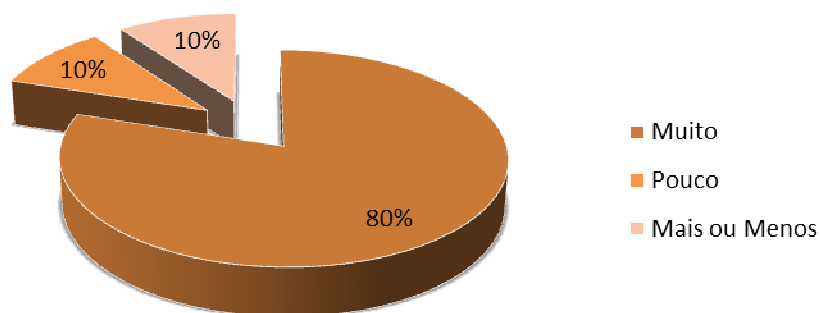
**Figura 46: Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação amplia a motivação dos alunos?**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 47 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação melhora o desempenho do aluno na disciplina?”. A maioria dos professores afirma que a utilização do ambiente gamificado melhorou o desempenho de alguns alunos na disciplina de lógica de programação.

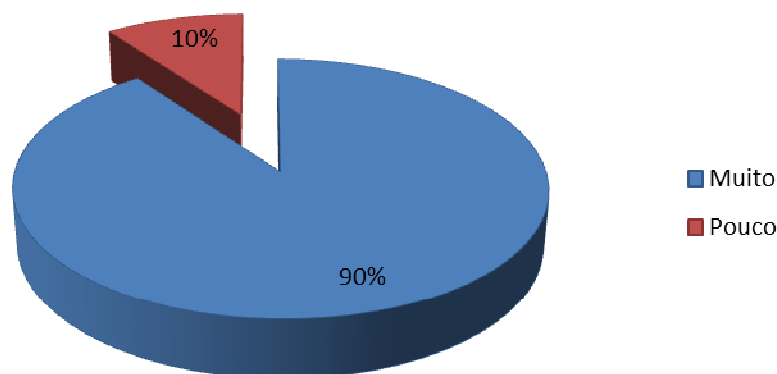
**Figura 47: Em sua opinião a utilização de um ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação melhora o desempenho do aluno na disciplina?**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

A figura 48 ilustra graficamente os resultados obtidos para a pergunta “Em sua opinião os dados fornecidos pela utilização do ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação contribui no planejamento da disciplina de lógica de programação?”. Neste caso, a maioria dos professores afirma que a utilização do ambiente gamificado aumentou a motivação dos alunos para o aprendizado.

**Figura 48: Em sua opinião os dados fornecidos pela utilização do ambiente gamificado como ferramenta de ensino e aprendizagem de lógica de programação contribui no planejamento da disciplina de lógica de programação?**



**Fonte: Elaborado pelo Autor, Canoas: 2016**

Observa-se pelas respostas dos professores que a utilização do ambiente gamificado no ensino e aprendizagem de lógica contribuiu para a efetividade no ensino dos conceitos básicos da lógica de programação.

No item que segue, expõem-se os resultados dos testes e uso do sistema.

### 6.3 Análise dos resultados de testes de uso do sistema

Com a aplicação das atividades de testes junto aos alunos, foi possível observar que o ambiente gamificado foi parcialmente bem sucedido no objetivo do estímulo ao aprendizado da lógica de programação através do uso da linguagem Lua aos alunos de cursos de nível técnicos relacionados a TI.

A interface gráfica do jogo, que foi concebida para manter um mesmo padrão de posição, cores e funcionalidade de botões em todas as telas, demonstrou-se efetiva, conforme detalhamento da figura 27, uma vez que todos os alunos participantes da atividade de testes classificaram-na como regular ou bom.

No quesito mecânica de jogo baseada em "apontar e clicar" a mesma também alcançou seus objetivos, uma vez que todos os alunos consideraram que a jogabilidade funcionou no mínimo regularmente bem. Em termos de diversão, a mecânica de jogo obteve sucesso parcial, uma vez que 63,6% dos alunos consideraram-na divertida ou muito divertida, enquanto que para 18% a mesma mostrou-se chata ou muito chata.

Em termos do ensino/aprendizagem, o Tri-Lua atingiu resultados satisfatórios, pois ao serem questionados sobre o quanto o jogo contribuiu para o aprendizado da lógica de programação através do uso da linguagem Lua os alunos, em geral, consideraram que o jogo foi sim proveitoso, como será apresentado a seguir, em detalhamento por cada unidade.

A unidade de manipulação de variáveis foi considerada de grande colaboração para o aprendizado da lógica de programação através do uso da linguagem Lua na opinião de 54,5% dos alunos, que consideraram a unidade boa ou muito boa, já para 27,3% alunos, esta unidade foi apenas regular.

A unidade de utilização de operadores matemáticos, por sua vez, foi aprovada por unanimidade, uma vez que 100% dos alunos consideraram-na proveitosa ou muito proveitosa para seu aprendizado da lógica de programação através do uso da linguagem Lua.

A unidade onde os alunos tiveram a oportunidade de aprender a manipulação de vetores e matrizes mostrou-se mais complicada para os mesmos, uma vez que esta unidade foi considerada nada proveitosa para 9,1% do alunos. Em contrapartida 18,2% consideraram esta unidade regular e 72,7% apontaram que esta unidade foi muito proveitosa para o entendimento de como ocorre a manipulação de vetores e matrizes em lógica de programação através do uso da na linguagem Lua.

Outra unidade aprovada por unanimidade pelos alunos foi a unidade de cláusulas condicionais, uma vez que 54,5% dos alunos relataram ter compreendido muito bem a utilização da cláusula *if* em lógica de programação através do uso da linguagem Lua, enquanto 36,4% consideraram ter entendido apenas bem.

A avaliação dos alunos diante da unidade de estruturas de repetição foi dividida entre as três estruturas apresentadas no jogo: *for*, *repeat-until* e *while*. A estrutura *for* foi bem compreendida pela grande maioria dos alunos, pois 81,8% dos alunos relataram ter conseguido compreender bem ou muito bem o funcionamento desta estrutura em lógica de programação através do uso da linguagem Lua, enquanto que apenas 9,1% relataram não ter conseguido compreender o funcionamento da estrutura.

Os níveis referentes a estrutura *repeat-until* demonstraram uma certa efetividade, sendo considerados parcialmente proveitosos para 9,1% dos alunos enquanto que para 81,8% aproveitamento mostrou-se bom ou muito bom.

E por fim os níveis referentes à estrutura *while* obtiveram determinado sucesso, uma vez que 81,9% dos alunos consideraram este nível proveitoso ou muito proveitoso em seu aprendizado, enquanto que apenas 18,1% consideraram que os níveis pouco agregaram a sua compreensão da utilização da estrutura em lógica de programação através do uso da linguagem Lua.

O objetivo “motivacional” implementado em segundo plano ao aprendizado da lógica de programação através do uso da linguagem Lua acabou-se por mostra-se inconclusivo, uma vez que não foi possível observar uma tendência sobre a motivação gerada pelo ambiente. Os resultados obtidos no teste apontam que 90,9% dos alunos sentiram-se motivados a seguirem aprendendo lógica de programação através do uso da linguagem Lua após os testes com o ambiente, número igual ao de alunos que mencionaram terem sentindo-se motivados ou muito motivados em buscarem mais conhecimento sobre lógica de programação.

Outros resultados obtidos através do questionário foram as sugestões e comentários gerais, questão esta que era opcional e foi respondida por um pequeno número de alunos. Dentre as observações realizadas, destacam-se críticas à unidade de tutoriais, a qual alguns alunos acharam bastante confusa, mas relataram ter conseguido esclarecer as dúvidas surgidas nos tutoriais ao decorrer das demais unidades de jogo. Tal *feedback* possibilitou a observação de fatores que poderiam ter sido melhor trabalhados na unidade tutorial, como por exemplo, o uso de um vocabulário menos técnico.

Também pode-se observar pontos a serem melhorados na interação do agente animado com o usuário, uma vez que mediante a observação das ações dos alunos durante o teste pode-se notar que alguns alunos demonstram muito impaciência ao ler com atenção todas as informações que eram apresentadas pelo agente. Uma ação que poderia ser tomada para

corrigir este problema seria retrabalhar a atuação do agente animado, tornando suas interações mais sucintas.

Após a conclusão das oficinas, os dados gerados através do uso do ambiente gamificado subsidiaram a utilização das técnicas de mineração de dados educacionais e assim foi possível apresentar aos professores que ministram a disciplina de lógica de programação um panorama da realidade dos alunos.

A partir da aplicação das técnicas de *Simple K-mean* e *Apriori* junto aos dados minerados foram extraídas informações relevantes sobre o desempenho individual dos alunos e, bem como, do grupo de alunos que participou da oficina.

Estes dados, após a apreciação dos professores, indicam que a utilização de um ambiente gamificado, a coleta de dados dessa interação e a sua posterior mineração podem ser ferramentas úteis no ensino e aprendizagem de lógica de programação.



## 7 CONSIDERAÇÕES FINAIS

Através deste trabalho apresentou-se o projeto Tri-Lua, objetivado pelo desenvolvimento de um ambiente gamificado que se propôs inicialmente a ser capaz de apoiar o ensino da lógica de programação através dos conceitos básicos da linguagem de programação Lua a alunos iniciantes em cursos técnicos na área da Tecnologia da Informação.

O projeto teve início com o levantamento de requisitos para o ambiente, onde através de uma profunda revisão bibliográfica sobre o assunto, foram abordados temas como o aprendizado de lógica de programação e aprendizagem através de atividades lúdicas de gamificação. Na sequência, foi realizada uma revisão de trabalhos relacionados, apresentando apenas alguns jogos de ensino de linguagens de programação atualmente disponíveis no mercado, destacando aqueles que de alguma maneira relacionam-se com a lógica de programação.

Após o levantamento dos requisitos iniciais, ocorreu a implementação do ambiente através do uso de tecnologias web. Uma vez finalizado, o ambiente foi utilizado em avaliações com uma turma de alunos do primeiro ano do Curso Técnico em Desenvolvimento de Sistemas integrado ao ensino médio do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - IFRS - Campus Canoas.

Dentre as possíveis melhorias observadas durante o período de testes, destacam-se potenciais reformulações que poderiam ser aplicadas na unidade de tutorial do jogo, bem como na interação do agente animado. Porém, há de ser observado que os pontos negativos apontados nos testes não impediram o ambiente de atingir seu objetivo, uma vez que no âmbito geral, os alunos relataram que vivenciaram uma experiência gamificada extremamente agradável e que conseguiram absorver o conteúdo proposto através do ambiente.

Na sequência das oficinas realizadas com os 20 alunos que compunham o grupo de controle na avaliação realizada, foram coletados os dados brutos gerados pela interação dos alunos com o ambiente gamificado. Estes dados brutos foram minerados através das técnicas de mineração de dados educacionais fornecendo informações sobre o desempenho dos alunos nos conceitos básicos de lógica de programação. Estes dados também deram origem a um relatório que permitiu aos professores conhecer melhor a realidade dos alunos possibilitando a adequação de seu planejamento e avaliações às necessidades reais dos alunos.

As avaliações realizadas pelos docentes envolvidos na pesquisa demonstraram que tanto a utilização do ambiente gamificado pelos alunos quanto o resultado disponibilizado pela mineração de dados educacionais constituíram-se importantes ferramentas para o ensino e aprendizagem de conceitos básicos de lógica de programação.

O objetivo geral do trabalho consistiu em analisar a potencialidade do uso das técnicas de gamificação aplicadas a uma ferramenta de apoio ao ensino e aprendizagem da lógica de programação em cursos técnicos integrados ao ensino médio e as vantagens advindas da sua integração com a Mineração de Dados Educacionais, como elemento de apoio à atividade do professor. Por sua vez, os objetivos específicos consistiram em (a) construir um modelo computacional do ambiente proposto, utilizando o conceito de gamificação e mineração de dados educacionais; (b) construir um protótipo do ambiente, utilizando o conceito de gamificação e mineração de dados educacionais; (c) avaliar o protótipo do ambiente

desenvolvido; e (d) contribuir na discussão da utilização de ambientes educacionais/digitais pedagogicamente atrativos ao aluno.

Observa-se que os objetivos deste trabalho foram atingidos na medida em que os objetivos específicos foram realizados. Foi construído o modelo do ambiente proposto utilizando-se o conceito de gamificação e de mineração de dados educacionais. O protótipo do ambiente foi implementado e em suas utilizações foram coletados dados brutos para a posterior aplicação das técnicas de mineração de dados educacionais. O ambiente implementado foi utilizado e avaliado pelos alunos que o utilizaram, bem como pelos professores participantes da pesquisa.

A construção e a utilização do ambiente gamificado contribuiu para a discussão sobre a relevância da implementação de ambientes educacionais/digitais pedagogicamente atrativos ao aluno, em especial no apoio ao ensino e aprendizagem de conceitos básicos de lógica de programação.

Pelo exposto ao longo do texto, verifica-se que a utilização de ambientes gamificados para o ensino e aprendizagem da lógica de programação demonstra potencialidade na medida em que amplia a motivação dos alunos para aprender conceitos básicos da disciplina de Lógica de Programação já que explora o lúdico através das dinâmicas e mecânicas dos jogos em ambientes de não jogos. Essa característica dos ambientes gamificados demonstrou-se eficaz no ensino e aprendizagem da lógica de programação tendo sido bem recebida pelos alunos e tendo colaborado para o aumento do desempenho dos alunos do grupo de controle nas avaliações da disciplina de Lógica.

Quanto aos professores, os dados gerados pela utilização do ambiente gamificado pelos alunos e que foram obtidos através das técnicas de mineração de dados educacionais mostraram-se de grande valia para o planejamento das atividades dos docentes que ministram a disciplina de Lógica de Programação.

Considera-se, apesar do contexto das avaliações realizadas não ser extensivo e contar com um número reduzido de alunos e professores, que a utilização de ambientes gamificados para o ensino e aprendizagem pode ser considerada como uma alternativa com bons indícios de ser promissora na busca por maior motivação e melhores resultados junto aos alunos de lógica de programação e que estes resultados podem ser aplicáveis a outras disciplinas do ensino médio ou ensino médio integrado.

## 7.1 Contribuições realizadas

Destacamos aqui algumas das contribuições deste trabalho, efetivadas até o momento.

Esta proposta de trabalho foi melhorada e transformada em um projeto de ensino que vem sendo aplicada amplamente a todos os níveis de educação atendidos pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS – Campus Canoas através do projeto *Gamificando o ensino da lógica de programação* as séries iniciais dos cursos superiores e de ensino médio técnico que possuem em sua grade curricular a disciplina de lógica de programação.

O trabalho realizado permitiu a escrita de artigo descrevendo a concepção inicial e fundamentação, que foi submetido e aprovado em evento de Uso de Jogos em Educação: ECGBL 2016 - 10th European Conference on Games Based Learning, Paisley, Scotland. Tri-Lua: Using gamification as Support Learning Programming Language.

Foi submetido e aceito artigo contendo os resultados do ambiente implementado e das avaliações realizadas, com título “Tri-lua – O Ensino da Lógica de Programação em uma

Proposta Lúdica Gamificada com Mineração de Dados”, na XXI Conferencia Internacional de Informática Educativa, TISE2016.

## **7.2 Trabalhos futuros**

Após a conclusão do Tri-Lua foram abertas algumas hipóteses para trabalhos futuros.

Em um primeiro momento, considera-se que poderia ser aproveitado o feedback proporcionado pelos testes com usuários para otimizar o potencial de ensino que o jogo possui atualmente. Para isso, poderiam ser feitas melhorias na unidade de tutoriais e na interação do agente animado com o usuário, além do acréscimo de mais atividades sobre os conteúdos que proporcionaram maior dificuldade para os jogadores, caso das unidades de tabelas e estruturas de repetição.

Outras possibilidades vislumbradas ao final do projeto, seriam a adição de um número maior de níveis ao jogo, de modo que fosse possível ensinar aos alunos conceitos mais complexos sobre lógica de programação através do uso da linguagem Lua.

Também foi observada a possibilidade da adição de um novo modo de jogo que fosse posterior ao modo de jogo atualmente disponível, aonde o aluno pudesse interagir escrevendo os algoritmos nativos na própria linguagem Lua, ao invés de "gerá-lo" por meio de atividades lúdicas.

## 8 REFERÊNCIAS

- AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. **Mining association rules between sets of items in large databases**. Proceedings of the 1993 ACM SIGMOD international conference on Management of data. **Anais...1993**
- ALMEIDA, E. S. DE et al. AMBAP: Um AMBiente de Apoio ao aprendizado de Programação. **Anais do 6o Congresso de Iniciação Científica – ASSER – UNICEN**, p. 91, 2002.
- ALVES, F. **Gamification: Como criar experiências de aprendizagem engajadoras. Um guia completo: do conceito à prática**. [s.l.] DVS Editora, [s.d.].
- ATKIN, C. **Education and Minorities**. [s.l.] A&C Black, 2012.
- BAKER, R. S. J. D.; YACEF, K. The State of Educational Data Mining in 2009 : A Review and Future Visions. **Journal of Educational Data Mining**, v. 1, n. 1, p. 3–16, 2009.
- BARBOSA, E. et al. Usando Google Maps para Educação Ambiental na Bacia do Capibaribe. **XIII JORNADA DE ENSINO, PESQUISA E EXTENSÃO – JEPEX**, 2013.
- BODENAVE, J. D.; PEREIRA, A. M. **Estratégias de ensino-aprendizagem (Parte do Livro)**. [s.l: s.n.].
- BRUSILOVSKY, P.; PEYLO, C. Adaptive and Intelligent Web-based Educational Systems ADAPTIVE AND INTELLIGENT TECHNOLOGIES FOR WEB-BASED EDUCATIONAL SYSTEMS. **International Journal of Artificial Intelligence in Education**, v. 13, p. 156–169, 2003.
- C. RAPKIEWICZ, G. FALKENBACH, L. S. E. A. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **RENOTE - Revista Novas Tecnologias na Educação**, v. 4, n. 2002, p. 1–10, 2006.
- CAMILO, C.; SILVA, J. Mineração de Dados: Conceitos, tarefas, métodos e ferramentas. **Universidade Federal de Goiás (UFG)**, p. 29, 2009.
- CASTRO, T. H. C. DE et al. **Utilizando programação funcional em disciplinas introdutórias de computação**. Anais do XXII Congresso da Sociedade Brasileira de Computação e X Workshop de Educação em Informática. **Anais...2002** Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2002/0015.pdf>>
- CAVALCANTE, G. **Tipagem Dinâmica**. Disponível em: <[https://gabrielacavalcante.gitbooks.io/python/content/tipagem\\_dinamica.html](https://gabrielacavalcante.gitbooks.io/python/content/tipagem_dinamica.html)>. Acesso em: 24 maio. 2016.
- CELES, W.; FIGUEIREDO, L. H. DE; IERUSALIMSCHY, R. **A Linguagem Lua e suas Aplicações em Jogos**.
- CL, A.; GIAMBIAGI, F. **O BNDES em um Brasil em Transição**. [s.l: s.n.].
- COSTA, E. et al. **Mineração de Dados Educacionais: Conceitos, Técnicas, Ferramentas e Aplicações**. Jornada de Atualização em Informática na Educação - JAIE 2012. **Anais...2012**
- CRESPO, L. C.; GIACOMINI, R. **As atividades lúdicas no Ensino de Química: uma revisão da revista Química Nova na Escola e das Reuniões Anuais da Sociedade Brasileira de Química**. Encontro Nacional de Pesquisa em Educação em Ciências. **Anais...2011**
- CRUZ, G. **A matemática na programação e desenvolvimento de software**. Disponível em:

<<http://cienciaetecnologias.com/matematica-na-programacao-desenvolvimento-software/>>. Acesso em: 24 maio. 2016.

- DECI, E. L.; RYAN, R. M. **Handbook of Self-determination Research**. [s.l.: s.n.].
- DETERDING, S. et al. From game design elements to gamefulness. **Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11**, p. 9–11, 2011.
- DINELLO, R. A. **Os Jogos e as Ludotecas**. Santa Maria: Organização Mundial para Educação Pré-Escolar - Santa Maria - RS, 2004.
- DUDA, R. Elaboração de aplicativos para Android com uso do App Inventor: uma experiência no Instituto Federal do Paraná – Câmpus Irati Introdução Tecnologia e Ensino de Matemática. v. 8, p. 115–128, 2014.
- FADEL, L. M. et al. **Gamificação na Educação**. São Paulo: Pimenta Comunicação E Projetos Culturais Ltda - ME, 2014.
- FALKEMBACH, G. A. M. et al. **Aprendizagem de algoritmos: uso da estratégia ascendente de resolução de problemas**. 8º Taller Internacional de Software Educativo. **Anais...2003** Disponível em: <[http://www.tise.cl/2010/archivos/tise2003/papers/aprendizagem\\_de\\_algoritmos.pdf](http://www.tise.cl/2010/archivos/tise2003/papers/aprendizagem_de_algoritmos.pdf)>
- FERREIRA, C.; GONZAGA, F.; SANTOS, R. Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração. **Computer**, p. 981–990, 2010.
- GARCÍA, E. et al. A collaborative educational association rule mining tool. **Internet and Higher Education**, v. 14, n. 2, p. 77–88, 2011.
- GLOVER, I. Play As You Learn: Gamification as a Technique for Motivating Learners. p. 1999–2008, 2003.
- GOIRI, I. **Gamificación y aprendizaje basado en el juego: ¿en qué se diferencian?** Disponível em: <<http://www.net-learning.com.ar/blog/infografias/gamificacion-y-aprendizaje-basado-en-el-juego-en-que-se-diferencian.html>>. Acesso em: 23 maio. 2016.
- GOMES, A.; MENDES, A. J. N. **Learning to program - difficulties and solutions**. ICEE - International Conference on Engineering Education. **Anais...Coimbra: International Conference on Engineering Education, 2007** Disponível em: <<http://icee2007.dei.uc.pt/proceedings/papers/411.pdf>>. Acesso em: 22 maio. 2016
- HOC, J. M. **Psychologie cognitive de la planification**. [s.l.] Livre papier, 1987.
- HUNICKE, R.; LEBLANC, M.; ZUBEK, R. MDA: A Formal Approach to Game Design and Game Research. **Workshop on Challenges in Game AI**, p. 1–4, 2004.
- IERUSALIMSCHY, R. **Programming in Lua**. Rio de Janeiro: Paperback, 2013.
- IERUSALIMSCHY, R.; MOURA, A. L. DE. **LabLua - Programming Language Research**. Disponível em: <<http://www.lua.inf.puc-rio.br/portugues.html>>. Acesso em: 22 maio. 2016.
- KAMPFF, A. J. C. et al. Identificação de Perfis de Evasão e Mau Desempenho para Geração de Alertas num Contexto de Educação a Distância. **Revista Latinoamericana de Tecnología Educativa**, v. 13, n. 2, p. 1–16, 2014.
- KAPP, K. M. **The Gamification of Learning and Instruction, Game-based Methods and Strategies for Training and Education**. San Francisco: Pfeiffer, 2012.
- KASTRUP, V. O funcionamento da atenção no trabalho do cartógrafo. **Psicologia & Sociedade**, v. 19, n. 1, p. 15–22, 2007.

- LADLEY, A. P. **Gamification , Education and Behavioural Economics Gamification of Life**, 2011.
- LISBOA, F. G. DA S. **Tipagem de Variáveis**. Disponível em: <<http://www.fgsl.eti.br/blog/?p=118>>. Acesso em: 24 maio. 2016.
- LOPES, S. **A Web Mobile: Programe para um mundo de muitos dispositivos**. 1ª Edição ed. São Paulo: Casa do Código, 2013.
- MACKENZIE, U. P.; ENGENHARIA, E. DE. **Alice : Uso Do Software No Processo Educacional Junto**. 2008.
- MANSSOUR, I. H. **Linguagem de Programação C++**. Disponível em: <<http://www.inf.pucrs.br/manssour/LinguagemC++/EstruturaControle.pdf>>. Acesso em: 24 maio. 2016.
- MARINACCI, J. **Point, Click, and Drool!** Disponível em: <<https://community.oracle.com/blogs/joshy/2003/12/01/point-click-and-drool>>. Acesso em: 22 maio. 2016.
- MCGONIGAL, J. **Reality Is Broken: Why Games Make Us Better and How They Can Change the World**. [s.l.] Penguin Publishing Group, 2011.
- MENDES, A. J. N. **Software educativo para apoio à aprendizagem de programação**. Disponível em: <[http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla\\_mendes.htm](http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm)>. Acesso em: 22 maio. 2016.
- MOSTOW, J. et al. **An Educational Data Mining Tool to Browse Tutor-Student Interactions : Time Will Tell!** Proceedings of the Workshop on Educational Data Mining (2005). **Anais...2005**
- MOTIL, J.; EPSTEIN, D. **JJ : a Language Designed for Beginners**. Disponível em: <<http://www.csun.edu/~jmotil/BeginLanguageJr.pdf>>. Acesso em: 22 jun. 2016.
- NG, E.; BEREITER, C. Three Levels of Goal Orientation in Learning on JSTOR. **The Journal of the Learning Sciences**, v. 1, n. 3/4, p. 243–271, 1991.
- PAULO, R.; JÚNIOR, M.; BONIATI, B. B. **LogicBlocks : Uma Ferramenta para o Ensino de Lógica de Programação**. Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação. **Anais...Frederico Westphalen: 2015**
- PINHO, M. S. **Matrizes e Vetores**. Disponível em: <<http://www.inf.pucrs.br/~pinho/Laproi/Vetores/Vetores.htm>>. Acesso em: 24 maio. 2016.
- RAABE, A. L. A.; SILVA, J. M. C. DA S. **Um Ambiente para Atendimento as Dificuldades de Aprendizagem de Algoritmos**. XXV Congresso da Sociedade Brasileira de Computação. **Anais...São Leopoldo: 2005**
- RAABE, A.; ZANCHETT, G.; VAHLICK, A. Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com à Programação. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, v. 4, n. Cbie, p. 1485, 2015.
- REATEGUI, E. B. et al. Agentes Pedagógicos Animados. **Renote Revista Novas Tecnologias na Educação**, v. 4, n. 2, p. 1–10, 2006.
- ROCHA, P. S. et al. Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. **RENOTE - Revista Novas Tecnologias na Educação**, v. 8, n. 3, p. 1–11, 2010.
- ROMERO, C. et al. Data mining algorithms to classify students. **Educational Data Mining**

2008, p. 8–17, 2008.

- SÁNCHEZ, J.-N. G. **Dificuldades de aprendizagem e intervenção psicopedagógica**. [s.l.] Artmed, 2004.
- SÁPIRAS, F. S.; VECCHIA, R. D.; MALTEMPI, M. V. Utilização do Scratch em sala de aula. **Educação Matemática Pesquisa**, v. 17, p. 973–988, 2015.
- SCHLEMMER, E. Gamificação em espaços de convivência híbridos e multimodais: design e cognição em discussão. **Revista da FAEEDBA-Educação e Contemporaneidade**, v. 23, n. 42, p. 73–89, 2014.
- SCHLEMMER, E. et al. Framework para gamificação. **PPGEDU / Unisinos - Seminário de Educação Digital**, 2015.
- SEBESTA, R. W. **Conceitos de linguagens de programação**. 9ª ed. São Paulo: Bookman, 2011.
- SELIVON, M.; BEZ, J. L.; TONIN, N. A. URI Online Judge Academic: Integração e Consolidação da Ferramenta no Processo de Ensino/Aprendizagem. **Anais do 23º Workshop sobre Educação em Computação (WEI 2015)**, n. CSBC, 2015.
- SHEN, R. et al. Data Mining and Case-Based Reasoning for Distance Learning. **International Journal of Distance Education Technologies**, v. 1, n. 3, p. 46–58, jan. 2003.
- SIoux, BLEND, E. **Pesquisa Game Brasil**. Disponível em: <<http://www.pesquisagamebrasil.com.br/>>. Acesso em: 29 maio. 2016.
- SOARES, N. **Crise? Não no mercado de Tecnologia da Informação**. Disponível em: <<http://educacao.estadao.com.br/noticias/geral,crise-nao-no-mercado-de-tecnologia-da-informacao,10000023666>>. Acesso em: 3 jun. 2016.
- TEECH. **Por que tantos estudantes largam o curso de Computação?** Disponível em: <<http://teech.com.br/noticia/1297/por-que-tantos-estudantes-largam-o-curso-de-computacao.html>>. Acesso em: 22 maio. 2016.
- TILLMANN, N. et al. Code hunt: Gamifying teaching and learning of computer science at scale. **Proceedings of the first ACM conference on Learning @ scale conference - L@S '14**, p. 221–222, 2014.
- TIOBE. **The Long Tail of Programming Languages**. Disponível em: <[http://www.tiobe.com/tiobe\\_index?page=index](http://www.tiobe.com/tiobe_index?page=index)>. Acesso em: 12 jun. 2016.
- VALENTIM, H. Um Estudo Sobre o Ensino-Aprendizagem de Lógica de Programação. **VII Enpec - Encontro Nacional de Pesquisa em Educação em Ciências**, 2009.
- WERBACH, K.; HUNTER, D. **For the Win: How Game Thinking Can Revolutionize Your Business**. [s.l.] Wharton Digital Press, 2012.
- ZICHERMANN, G.; CUNNINGHAM, C. **Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps**. [s.l.] “O’Reilly Media, Inc.”, 2011.
- ZORRILLA, M. E. et al. **Web usage mining project for improving Web-based learning sites**. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). **Anais...2005** Disponível em: <[http://link.springer.com/10.1007/11556985\\_26](http://link.springer.com/10.1007/11556985_26)>. Acesso em: 12 jun. 2016