

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MARCIO DAVI MARTINS MACIEL

UMA FERRAMENTA GAMIFICADA DE APOIO AO ENSINO DE ALGORITMOS

NATAL / RN

2015

MARCIO DAVI MARTINS MACIEL

UMA FERRAMENTA GAMIFICADA DE APOIO AO ENSINO DE ALGORITMOS

Trabalho de Conclusão de Curso apresentado ao Centro Universitário do Rio Grande do Norte como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Orientadora: Prof^ª. MSc. Joseane Alves Pinheiro

NATAL / RN

2015

Catalogação na Publicação – Biblioteca do UNI-RN
Setor de Processos Técnicos

Maciel, Marcio Davi Martins.

Uma ferramenta gamificada de apoio ao ensino de algoritmos / Marcio Davi Martins Maciel. – Natal, 2015.

XX f.

Orientadora: Profª. MSc. Joseane Alves Pinheiro.

Monografia (Graduação em Sistemas de Informação) – Centro
Universitário do Rio Grande do Norte,

1. Gamificação – Monografia. 2. Algoritmos – Monografia. 3. Ensino –
Monografia. 4. Aprendizado – Monografia. I. Pinheiro, Joseane Alves. II.
Título.

RN/UNI-RN/BC

CDU 004.4

MARCIO DAVI MARTINS MACIEL

UMA FERRAMENTA GAMIFICADA DE APOIO AO ENSINO DE ALGORITMOS

Trabalho de Conclusão de Curso apresentado ao Bacharelado em Sistemas de Informação do Centro Universitário do Rio Grande do Norte - UNIRN como requisito parcial para obtenção do título de bacharel em sistemas de informação.

Aprovada em 08 / 12 / 2015

BANCA EXAMINADORA:

Joseane Alves Pinheiro

Prof. Joseane Alves Pinheiro

Orientadora

Prof. Luiz Gonzaga Damasceno

Membro

Maíra de Faria Barros Medeiros

Prof. Maíra de Faria Barros Medeiros

Membro

Dedico este trabalho aos meus pais Francisco Cláudio Maciel e Maria Lucy Martins Maciel pela paciência e compreensão, à minha tia Mundinha Camurça pelo consolo, à minha amiga e orientadora Joseane Alves Pinheiro pela dedicação e ao meu amor Irandir Dantas pelo apoio.

AGRADECIMENTO

Em primeiro lugar, aos meus pais Francisco Cláudio Maciel e Maria Lucy Martins Maciel, devo todo o meu amor, respeito e gratidão por nunca desistirem de mim e pela paciência durante este longo período, há muito esperado.

A minha tia 'Mundinha' Camurça, devo minha vida, por me salvar da escuridão em um momento de grande dificuldade pessoal com suas palavras de ternura e consolo.

À minha orientadora, professora e amiga, Joseane Alves Pinheiro, agradeço pela paciência, pela dedicação, pelas brincadeiras e pelas ideias criativas sempre me incentivando a melhorar cada vez mais.

Aos amigos, que tive o prazer de conhecer e compartilhar vitórias e derrotas ao longo deste período turbulento, Irandir Dantas, Dora Bortone e Paula Vasconcelos, Fábio Villas Boas e a todos que gostaria de citar mas que por um motivo ou outro não estão nesta lista.

Aos meus colegas de turma que sempre me ajudaram quando precisei, que dividiram comigo este período de provação e que agora compartilham da mesma vitória.

Quero ainda agradecer aos portais online *Google*, *Wikipedia* e *Stackoverflow*, pois sem sua base de conhecimento, eu não teria superado alguns dos problemas técnicos em tempo hábil ao longo deste curso.

Por fim, alcancei esta conquista através do conjunto formado principalmente pelo meu próprio esforço, pela ajuda e pelo apoio de amigos, colegas e familiares, então, embora não me considere ateu, gostaria de agradecer a Deus por nada.

RESUMO

A disciplina de algoritmos é considerada base para o ensino de programação em cursos voltados para a área de Tecnologia. Ela envolve conceitos da matemática e da lógica, com o objetivo de desenvolver no aluno a capacidade de análise e resolução de problemas de forma descritiva e sequencial, ou seja, na forma de algoritmos. Porém ela exige também a capacidade de abstração de dados, ou seja, a capacidade de visualizar objetos do mundo real em representações na forma conjuntos de dados relevantes a um determinado problema, manipuláveis por um computador. Para pessoas que nunca tiveram contato com a área, essa abstração geralmente se torna uma barreira para o aprendizado, sendo uma das causas para as altas taxas de evasão, reprovação e desmotivação na disciplina. Assim, este trabalho objetiva fornecer um embasamento teórico sobre temas abordados na disciplina de algoritmos I, trazer o conceito de gamificação, isto é, a aplicação de conceitos do mundo dos jogos de computador em ambientes diferentes dos jogos, com o objetivo de motivar os alunos tornando o aprendizado mais fácil e interessante e propor uma ferramenta gamificada para apoio ao ensino de algoritmos.

Palavras-Chave: Gamificação. Algoritmos. Ensino. Aprendizado.

ABSTRACT

The discipline of algorithms is often thought as the basis for the teaching of programming in courses focusing on information technology. It involves concepts of mathematics and logic, in order to help students to develop or enhance the ability to *analyze and solve problems in a descriptive and sequential way, that is, as algorithms*. However, it also requires the skill of data abstraction, which can be thought as the ability to view and represent real-world objects as sets of relevant data to a particular problem, manipulable by a computer. For those who have never had contact with the area, this abstraction often becomes a barrier to learning, being one of the causes for the high evasion rates, failure and demotivation surrounding the discipline. This paper aims to provide a theoretical foundation on topics covered in the discipline of algorithms I, raising the concept of gamification, that is, the application of concepts from the world of games in other environments or situations, as a possible solution for the problem of motivating students by making learning easier and more interesting and finally to propose a gamified computational tool to support the teaching of algorithms.

Keywords: Gamification. Algorithms. Teaching. Learning.

1. INTRODUÇÃO	9
1.1. JUSTIFICATIVA	10
1.2 OBJETIVOS	10
1.2.1 Gerais:	10
1.2.2 Específicos	10
2 REFERENCIAL TEÓRICO.....	11
2.1 ALGORITMOS	11
2.1.1 Conteúdo da disciplina	13
2.1.2 Composição e representação.....	15
2.1.3 Abordagem tradicional em sala	24
2.2 GAMIFICAÇÃO	26
2.2.1 O que é?	26
2.2.2 Para que serve?	27
2.2.3 Como funciona?	27
2.2.4 Exemplos de sistemas gamificados	29
2.2.5 Quais os elementos de um projeto de gamificação?	30
2.3. Gamificação aplicada ao ensino	34
3 METODOLOGIA	36
4 RESULTADOS.....	36
4.1 A PESQUISA.....	36
4.1.1 Ferramentas e estratégias de ensino	37
4.2 FERRAMENTA PROPOSTA.....	38
4.2.1 Definição e escopo.....	38
4.2.2 Modelagem.....	41
5 CONCLUSÃO	55

6 REFERÊNCIAS BIBLIOGRÁFICAS	56
6.1 LEVANTAMENTO BIBLIOGRÁFICO PRELIMINAR.....	56
6.2 REFERÊNCIAS UTILIZADAS	60

1. INTRODUÇÃO

Este trabalho visa apresentar um panorama geral sobre a disciplina de algoritmos e as estratégias comuns utilizadas no seu ensino, propondo uma ferramenta computacional de apoio aos professores no ensino dessa disciplina, mostrando a modelagem inicial da ferramenta.

Com a motivação do aluno em foco, este trabalho pretende dar ênfase à aplicação de conceitos de gamificação, como agentes motivadores tanto no âmbito teórico da sala de aula quanto prático em laboratórios, enfatizando a importância do uso de ferramentas computacionais no apoio ao ensino.

Devido à natureza dos conteúdos vistos na disciplina, e pelo fato de poder ser aplicada em qualquer área do conhecimento, a gamificação requer a capacidade de abstração de partes do mundo real para representações virtuais. Em outras palavras, transformar objetos e ideias concretas ou abstratas em algo relevante e manipulável por um computador. Por exemplo, como representar um conjunto de mesas para o problema de organização física das mesas em uma sala de aula? Quais as informações pertinentes? O que é uma mesa para um computador? Será que a estrutura molecular da madeira e dos metais afetaria o resultado do problema? Podemos pensar em abstração como sendo a aplicação de um 'filtro' sobre um objeto, ideia ou conceito, extraíndo dele apenas informações relevantes ao que se deseja representar ou resolver. Alunos que nunca tiveram contato com o computador ou que nunca o utilizaram para resolução de problemas, encontram mais dificuldade em desenvolver as habilidades e as competências necessárias para obter sucesso nestas disciplinas e, muitas vezes acabam se desmotivando e desistindo no meio do caminho.

Além disso, existe uma alta taxa de evasão e de reprovação na disciplina. Assim, manter o aluno focado e motivado se torna o verdadeiro desafio do professor. Será possível utilizar uma abordagem diferente para tornar as aulas mais interessantes e motivadoras, aumentando assim o grau de aprendizado e reduzindo as taxas de evasão e reprovação?

1.1. Justificativa

Além da dificuldade em abstração de dados, existe muita resistência à ideia de mesclar a abstração com a lógica relacional, construindo uma linguagem semi-formal capaz de descrever problemas complexos a partir de proposições atômicas mais simples. Além disso existem fatores inerentes ao próprio sistema de aulas com grupos de alunos que por vezes dificultam a absorção deste conteúdo. Diferentes alunos, interpretam a mesma informação de maneira diferente. Alunos com formações diferentes podem ter mais dificuldade em aspectos distintos. O professor também sente dificuldade no gerenciamento de turmas muito grandes bem como em avaliar individualmente o grau de aprendizado de cada aluno. A didática aplicada em sala de aula muitas vezes torna as aulas tediosas e cansativas. E a lista não para por aí.

Para auxiliar o professor a lecionar a disciplina em sala de aula, não só para os alunos da área de informática, mas também de outras áreas, várias ferramentas computacionais têm sido adotadas nos últimos anos, desde simples apresentações de slides até o uso de ferramentas online distribuídas para promover competição e colaboração entre alunos. Mais recentemente, técnicas de gamificação vêm ganhando força à medida em que seus resultados se mostram cada vez mais significativos no que diz respeito a manter o usuário interessado em um produto ou serviço.

1.2 Objetivos

A seguir, descrevemos os objetivos gerais e específicos esperados.

1.2.1 Gerais:

Utilizar conceitos de jogos aplicando-os no cenário da disciplina de algoritmos, visando manter o aluno interessado, focado e motivado, melhorando a taxa de aprendizado e, conseqüentemente, reduzindo as taxas de evasão e reprovação da mesma, enquanto torna as aulas mais práticas, dinâmicas e atraentes.

1.2.2 Específicos

- Apresentar as estratégias mais comuns utilizadas nas aulas da disciplina de algoritmos;

- Analisar ferramentas computacionais ou não utilizadas na disciplina de algoritmos;
- Analisar as estratégias de gamificação usadas em jogos computacionais;
- Elaborar a estrutura de uma ferramenta computacional de apoio ao ensino;
- Modelar uma ferramenta de apoio ao ensino das disciplinas de algoritmos com uso da gamificação, ou seja, utilizando conceitos vindos dos jogos, como noções de personagens, roteiro, mapas de progresso, competição e cooperação, quadro de pontuação entre outros. Inicialmente a ferramenta abordará temas vistos na disciplina de Algoritmos I, como variáveis e seus tipos, estruturas de controle e repetição, entrada e saída de dados, análise de problemas e como aplicar estes conceitos ordenadamente de forma a montar um algoritmo para resolvê-los. A ferramenta será executada em plataforma online, sendo dividida em dois módulos: o de administração de conteúdo e montagem de lições; e o de interação com o usuário, a partir de então tratado como estudante/jogador.

2 REFERENCIAL TEÓRICO

O referencial teórico está dividido em 3 partes: o conteúdo relacionado à disciplina de algoritmos I; o processo de Gamificação e; a aplicação da gamificação no ensino da disciplina de algoritmos.

2.1 Algoritmos

A seguir é apresentado uma breve explicação sobre conceitos e definições relacionadas aos algoritmos. Não faz parte do escopo deste trabalho entrar em detalhes sobre o tema, apenas apresentar as principais noções necessárias para seu entendimento e funcionamento.

Existem muitas definições para o termo algoritmo. Informalmente pode ser feita uma analogia com receitas de bolo, um conjunto de instruções ordenadas que realizam alguma tarefa. Na área de informática pode ser vista como qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída. Portanto, “um

algoritmo é uma sequência de passos computacionais que transforma uma entrada em uma saída" (CORMEN et al., 2012).

Um algoritmo é uma ferramenta aplicada na resolução de problemas bem definidos. O enunciado do problema define uma relação entre dados de entrada e saída e um algoritmo vai descrever um procedimento computacional específico para que esse relacionamento seja alcançado.

Na prática, algoritmos podem ser aplicados na resolução de problemas de praticamente qualquer área. Alguns exemplos são citados a seguir.

O projeto Genoma Humano visa identificar todos os 100.000 genes do DNA humano, determinar as sequências dos 3 bilhões de pares de bases químicas, armazenar essas informações em bancos de dados e desenvolver ferramentas para analisar tais dados.

A internet conecta milhões de pessoas espalhadas pelo mundo. A quantidade de informações que trafega pela rede é gigantesca. Para gerenciar e manipular tanta informação, são aplicados algoritmos inteligentes para resolução de diversos problemas, desde a localização de rotas pelas quais os dados viajarão (problema de roteamento) até o uso de mecanismos eficientes de pesquisa para encontrar páginas específicas.

Na indústria, por exemplo, frequentemente é preciso alocar recursos limitados de maneira eficiente para obter o melhor resultado. Uma empresa petrolífera pode desejar saber onde posicionar seus poços para maximizar o lucro esperado.

Um candidato a presidência da república pode precisar de relatórios indicando os melhores locais para investir em publicidade de campanha para aumentar suas chances de vencer uma eleição. Um viajante pode querer definir a melhor rota para visitar determinados pontos turísticos gastando a menor quantia ou o menor tempo possível.

As possibilidades são ilimitadas e frequentemente determinado problema admite mais de um algoritmo para resolvê-lo. Porém, embora boa parte destes problemas possa ser resolvida eficientemente por algoritmos, existe também um conjunto de problemas para os quais não se conhece ainda uma solução eficiente, conhecidos como problemas NP-completos (CORMEN et al., 2012).

2.1.1 Conteúdo da disciplina

Os conteúdos relacionados aos algoritmos são muito variados tanto em natureza (classe de problemas) quanto em complexidade (dificuldade e associação com outros conteúdos ou disciplinas). Por isso, em muitos cursos o tema é dividido em várias disciplinas. O escopo deste trabalho está baseado nos conteúdos iniciais vistos na disciplina de Algoritmos I, incluindo entrada e saída de dados, variáveis e tipos de dados, estruturas de repetição e controle.

Mas antes de iniciar uma discussão sobre o assunto, algumas definições são necessárias para que o aluno possa formar sua base. A seguir as listamos:

DEFINIÇÕES:

- SOBRE O COMPUTADOR:
 - CPU ou UCP (Unidade Central de Processamento): é o conjunto de dispositivos eletrônicos responsável pelas operações de processamento e cálculos lógicos e matemáticos.
 - MEMÓRIA: são dispositivos responsáveis pelo armazenamento de dados. Os tipos principais de memória são a memória RAM (*Random Access Memory*) ou principal que é volátil, ou seja, perde o conteúdo armazenado ao se cortar a energia e a memória secundária que é usada para armazenar grandes quantidades de dados e é não-volátil, mantendo seu conteúdo armazenado mesmo sem energia.
 - DISPOSITIVOS DE ENTRADA E SAÍDA: são dispositivos que o usuário pode usar para passar informações ao computador (entrada) como teclados e mouses ou receber informações do mesmo (saída) como monitores ou impressoras.
- SOBRE UM ALGORITMO:
 - TIPOS PRIMITIVOS: cada dado em um computador deve ser armazenado de acordo com o tipo de informação que ele representa. O uso de tipos adequados aos dados permite a otimização de recursos disponíveis e a aceleração do processamento. Os principais tipos primitivos são: **inteiro**, números pertencentes ao conjunto dos Inteiros,

sem parte fracionária; **real**, números pertencentes ao conjunto dos Reais, também chamados de ponto flutuante; **caractere** que são valores pertencentes ao conjunto composto por números de 0 a 9, letras de A a Z e caracteres especiais como '!', '@', '#', '%' entre outros, também chamado de conjunto alfanumérico e; **lógico**, que representa apenas dois valores, verdadeiro ou falso, também podendo ser interpretado de outras formas como ligado e desligado, alto e baixo, aberto e fechado.

- **CONSTANTES E VARIÁVEIS:** existem basicamente duas categorias de dados: **constantes** e **variáveis**. Constantes são dados que não mudam de valor ao longo da execução dos algoritmos, enquanto variáveis podem ser alteradas de acordo com sua execução. Vemos frequentemente estes dois conceitos na matemática. Por exemplo, no cálculo da área da circunferência ($A = \pi r^2$), temos π como uma constante, cujo valor é fixo em aproximadamente 3,1416... e r como uma variável, representando o raio da circunferência, que pode assumir diferentes valores.
- **IDENTIFICADORES:** são os nomes que damos a um dado, para que ele possa ser identificado e manipulado pelo computador. Enquanto na matemática, é comum termos identificadores para constantes e variáveis formados por uma única letra, como X, Y ou A, em algoritmos, geralmente, damos nomes significativos que facilitem o entendimento tanto pelo leitor quanto pelo escritor do algoritmo. O conteúdo do identificador é o próprio valor do dado e seu nome (identificador) é usado para acessar o dado e realizar operações sobre ele. Existem algumas regras para nomeação de identificadores válidos:
 - Devem ser iniciados por caractere alfabético ou '_' (sublinhado ou *underline*);
 - Pode ser seguido por um ou mais caracteres alfanuméricos
 - Não pode conter caracteres especiais nem espaços, exceto o *underline*.

- Não pode ser usado palavras reservadas, ou seja, palavras específicas da linguagem de programação em uso, como comandos, tipos de variáveis, etc.

2.1.2 Composição e representação

Um algoritmo é composto em essência por três partes: **entrada**, **processamento e saída**. Além disso, pode ser representado de várias formas, entre as mais comuns, tem-se o fluxograma, o pseudo-código e as linguagens de programação. Nas figuras número 1 e 2 respectivamente, podemos ver a representação em fluxograma e pseudocódigo:

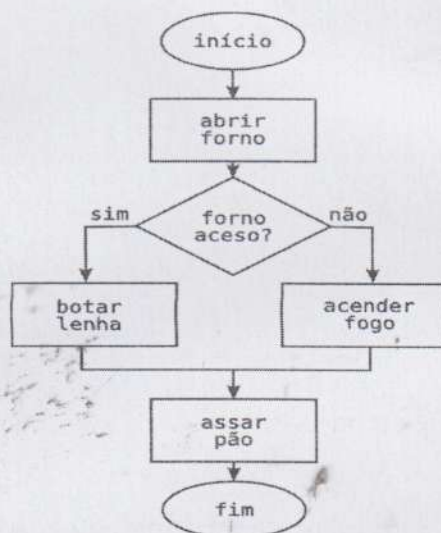


Figura 1 - Exemplo de algoritmo representado por um fluxograma

```

1: ir até a parada
2: enquanto ônibus não chega faça
3:     esperar ônibus
4: fim-enquanto
5: subir no ônibus
6: pegar passagem
7: se não há passagem então
8:     pegar dinheiro
9: fim-se
10: pagar o cobrador
11: troco ← dinheiro - passagem
12: enquanto banco não está vazio faça
13:     ir para o próximo
14: fim-enquanto
15: sentar
16: ...

```

Figura 2 - Exemplo de algoritmo em pseudo-código para pegar ônibus

COMANDOS E OPERAÇÕES:

1. DECLARAÇÕES (OU DEFINIÇÕES): são instruções que definem variáveis ou constantes, geralmente também associando um tipo de dado e um identificador a elas, para que possam ser processadas pelo computador. Identificadores associados a um tipo de dados, ficam restritos a armazenar dados daquele tipo apenas. Sendo assim, tentativas de armazenar dados de outro tipo geralmente resultam em erros.

Um exemplo de declaração de variável em pseudo-código:

var nome1, nome2, nome3 : inteiro

A instrução acima define três variáveis do tipo inteiro com os identificadores 'nome1', 'nome2' e 'nome3'.

Já um exemplo de declaração de constante seria:

constante pi = 3.141592

Ao declararmos constantes ou variáveis, estamos solicitando ao computador que reserve uma parte da memória para armazenar dados do tipo especificado. Durante a execução de um algoritmo, um programa armazena por padrão as informações na memória principal (RAM). Naturalmente a quantidade de memória disponível é finita,

embora atualmente ainda seja grande o bastante para armazenar grandes volumes de dados.

2. ATRIBUIÇÕES: após a declaração de um identificador, podemos começar a manipulação de seus dados. A primeira forma de manipulação é a atribuição, que consiste em informar qual o dado que será associado àquele identificador. Uma vez informado o valor do dado, este é armazenado na posição de memória previamente reservada pela declaração do identificador. Exemplos de atribuições seriam:

$\langle \text{identificador_da_variavel} \rangle \leftarrow \langle \text{valor do mesmo tipo da variável} \rangle$

$\langle \text{id_da_variavel} \rangle \leftarrow (\langle \text{operações que resultam em um valor do mesmo tipo} \rangle)$

3. EXPRESSÕES: Expressões são operações executadas sobre um ou mais valores para determinar uma relação ou um resultado após seu processamento.

- a. ARITMÉTICAS: são expressões que executam conjuntos de operações aritméticas cujos operadores são aritméticos e os operandos são numéricos. As operações aritméticas principais são a adição, a subtração, a multiplicação, a divisão, a potenciação, a divisão inteira e o resto (ou módulo).

Operação	Operador	Exemplo
Adição	+	2+3, 9+x
Subtração	-	x-1, f(x)-2
Multiplicação	*	2*1, x*y, 2*g(x)
Divisão	/	1/x, a/z, 2*f(x)
Potenciação	**	10**x, 2**3
Resto (Módulo)	mod	10 mod 2, 120 mod 10
Divisão Inteira	div	10 div 2, 120 div 10

Figura 3 - tabela de operadores aritméticos principais

- b. LÓGICAS: são aquelas cujo valor resultante assume apenas dois valores possíveis: verdadeiro ou falso. Podem ser compostas por operadores

lógicos, relacionais, identificadores de variáveis ou constantes de tipo lógico, ou ainda por resultados de expressões aritméticas.

- i. **Operadores relacionais:** são aqueles que comparam 2 (dois) valores do mesmo tipo afim de estabelecer uma relação entre eles. Seu valor de retorno indica se a relação é verdadeira ou falsa. Por exemplo:

$(20 < 30)$ e $(10 < 2)$

são duas expressões lógicas válidas, embora seus resultados sejam verdadeiro na primeira e falso na segunda. Os operadores relacionais são : maior ($>$) e maior ou igual ($>=$), menor ($<$) e menor ou igual ($<=$), igual ($=$ ou $==$) e diferente ($<>$ ou $!=$).

- ii. **Operadores lógicos:** também chamados de conectivos lógicos, são usados para representar situações lógicas ou unir duas expressões simples em uma composta. Podem ser binários, operando sobre duas expressões ou unários, operando sobre apenas uma. Os operadores lógicos mais comuns são o AND ($\wedge$, e lógico ou conjunção, binário), OR ($\vee$, ou lógico ou disjunção, binário) e o NOT ($\neg$, negação lógica, unário). Um exemplo de expressão lógica seria: $(1 < 3) \vee (10 <> 10)$, cujo resultado é verdadeiro.

Quando uma expressão composta precisa ser avaliada em um algoritmo, o analisador dá prioridade a certos operadores, o que é chamado de **precedência de operadores**. As sub-expressões contendo estes operadores são avaliadas primeiro e substituídas pelo seu resultado. A próxima sub-expressão na ordem é avaliada e substituída, e assim por diante até que a expressão tenha sido totalmente analisada e seu resultado seja um só valor.

4. **ENTRADA E SAÍDA DE DADOS:** representam as operações de interação entre usuário e máquina, por onde o usuário informa valores (entrada) e visualiza resultados (saída).

- a. **Saída:** permite informar dados ao usuário através de dispositivos de saída, por padrão, a tela (monitor). Outro exemplo de dispositivos de saída são as impressoras. Em pseudo-código, usa-se o comando **escreva** (ou *write*), seguido pela informação que se deseja informar, opcionalmente entre parênteses. Por exemplo, o comando "escreva ('olá pessoal!')", irá mostrar no dispositivo de saída padrão (monitor) o texto "olá pessoal!" (sem as aspas). Também pode-se imprimir valores de variáveis e constantes, através de seus identificadores. Por exemplo, se tivermos a declaração "var nome" e a atribuição "nome ← 'Marcio Davi'", poderíamos imprimir o conteúdo da variável 'nome' através do comando "escreva (nome)". É importante ressaltar que: para imprimirmos o conteúdo de um identificador, ele deve vir sem as aspas, caso contrário será tratado como um texto comum. Enquanto a instrução anterior imprime "Marcio Davi", a instrução "escreva ('nome')" irá imprimir na tela o texto "nome". O mesmo tratamento é dado para impressão do resultado de expressões aritméticas e lógicas.
- b. **Entrada:** permite, ao usuário, fornecer dados para o algoritmo através de um dispositivo de entrada, por padrão, o teclado. Outros dispositivos de entrada incluem o mouse, o scanner e os leitores biométricos. Em pseudo-código, geralmente se usa o comando **leia** (ou *read*). O programador deve saber previamente qual a variável que armazenará o valor informado pelo usuário, ou seja, deve-se declarar variáveis de tipo adequado para o armazenamento dos valores antes de tentar ler um dado da entrada. Frequentemente os comandos 'escreva' e 'leia' são usados em conjunto, para informar o usuário sobre o dado que ele precisa fornecer e então ler da entrada este valor, armazenando em variável.

5. ESTRUTURAS DE CONTROLE:

Também chamadas de condicionais, são estruturas que permitem controlar e alterar o fluxo de execução de um algoritmo, ou seja, escolher conjuntos de instruções a executar dependendo de alguma condição pré-definida, diferentemente do fluxo ordenado e sequencial padrão. Por exemplo, imagine que desejamos saber se um aluno foi aprovado por média (maior ou igual a 7), através do valor da variável 'nota_final' e imprimir para o usuário o resultado "Aprovado" ou "Reprovado" dependendo da nota do aluno. Para este caso, precisamos testar o valor da variável e **se** a nota for maior ou igual a 7, imprimir "Aprovado", **senão** imprimir "Reprovado".

As estruturas de controle mais comuns são o "**se-então-senão**" e o "**escolha-caso**".

SE-ENTÃO-SENÃO: permite testar uma expressão lógica inicial (condição) e, caso seu resultado seja verdadeiro, executa o bloco de instruções ENTÃO. Opcionalmente, pode-se acrescentar um bloco SENÃO para instruções que serão executadas caso a condição inicial tenha resultado falso. Vale também ressaltar que dentro de cada bloco de instruções pode haver outras estruturas tanto de controle quanto de repetição (mais a frente), caracterizando **estruturas encadeadas**. O formato geral da estrutura se-então-senão é o seguinte:

```
se <expressão lógica> então
    <bloco de instruções caso condição seja verdade>
senão
    <bloco de instruções caso condição seja falsa>
fim-se
```

No exemplo acima da nota do aluno, poderíamos escrever em pseudocódigo o seguinte:

```
se nota_final >= 7 então
    escreva ( "Aprovado!" )
senão
    escreva ( "Reprovado!" )
fim-se
```

ESCOLHA-CASO: ou caso-seja, permite verificar uma variável por várias condições, executando um bloco de instruções para cada um e, opcionalmente definindo ainda um bloco de instruções padrão, caso nenhuma das condições seja avaliada como verdadeira. Seu formato em pseudocódigo é da forma:

```

escolha <variável>
    caso <valor1>:
        <bloco 1>
    caso <valor2>:
        <bloco 2>
    caso <valor3>:
        <bloco 3>
    padrão:
        <bloco padrão>
fim-escolha

```

6. ESTRUTURAS DE REPETIÇÃO:

Enquanto estruturas de controle determinam que fluxo a execução do algoritmo deve seguir, as estruturas de repetição fornecem uma maneira de repetir blocos de instruções específicas um determinado número de vezes (laço ou *loop*). Esse número pode ser um valor absoluto ou uma expressão aritmética ou mesmo lógica, dependendo da estrutura utilizada. Tal número ou expressão é chamado **condição de controle**, ou **condição de parada**, que é testada em cada ciclo de repetição até que seja determinado quando encerrar o *loop*.

As principais estruturas de repetição são: **enquanto-faça**, **faça-enquanto** e **para-até-faça**.

ENQUANTO-FAÇA: Nessa estrutura, antes de entrar no *loop*, a condição de parada é testada. Enquanto a condição for avaliada como verdadeiro, o *loop* se repete

e antes de cada repetição a condição é re-testada. Quando a condição resultar em falso, o *loop* se encerra e o fluxo do algoritmo retorna para a instrução seguinte ao ciclo.

Geralmente o mecanismo que altera o resultado da condição de parada está incluído dentro do bloco executado dentro da estrutura ou é dependente de alguma variável externa fornecida pelo usuário ou calculada em tempo de execução. Esta estrutura é útil quando não sabemos quantas vezes o bloco de instruções deve ser repetida.

O formato da estrutura enquanto-faça em pseudocódigo segue o modelo:

```
enquanto <expressão lógica> faça
    <bloco de comandos>
<fim-enquanto>
```

FAÇA-ENQUANTO: segue um raciocínio semelhante ao enquanto-faça, mas sua principal diferença é que a condição de parada é testada depois da execução do bloco de repetição. Isso garante que o bloco seja executado pelo menos uma vez, antes de encerrar a repetição e continuar o fluxo normal do algoritmo. Seu formato em pseudocódigo segue o modelo:

```
faça
    <bloco de comandos>
enquanto <expressão lógica>
```

PARA-ATE-FAÇA: nessa estrutura, define-se um mecanismo para regular o número de repetições usando um intervalo de um valor 'início' até um determinado valor 'fim', definindo também o **incremento** do valor de controle a cada repetição. Segue o seguinte modelo em pseudocódigo:

```
para <variavel> de <valor_inicial> até <valor_final> passo <incremento> faça
    <bloco de comandos>
<fim-para>
```

Por exemplo:

```

var cont : inteiro
para cont = 2 até 10 passo 2 faça
    escreva ( <cont> + " é um número par")
<fim-para>

```

O trecho acima declara uma variável 'cont' e um laço onde 'cont' receberá os valores 2, 4, 6, 8 e 10, escrevendo na tela a saída:

```

"2 é um número par"
"4 é um número par"
"6 é um número par"
"8 é um número par"
"10 é um número par"

```

CONTADORES E ACUMULADORES: as vezes é preciso armazenar valores que representam o número de ocorrências de determinado valor, evento, ação etc., ou armazenar somatórios ou produtórios de determinados valores em um laço.

Contadores são variáveis especiais usadas para guardar quantidades de ocorrências. Geralmente são inicializados com o valor 0 ou 1 e incrementados em 1 cada vez ocorre uma nova ocorrência. Já os **acumuladores** são usados para armazenar somas ou produtos acumulativos. No caso de somas, o acumulador geralmente é inicializado com 0 e em cada ocorrência é incrementado por um valor qualquer determinado pelo problema em questão. Já no caso de produtos, o acumulador geralmente é inicializado com 1 mas segue o mesmo raciocínio.

Exemplo algoritmo com contador:

```

var numero_de_alunos ← 25
var num_aprovados ← 0
var nota : real
var cont : inteiro
para cont de 0 ate numero_de_alunos passo 1 faça
    escreva ("nota final do aluno " + cont )
    leia ( nota )

```

```

    se nota > 7 entao
        num_aprovados ← num_aprovamos + 1
    fim-se
fim-para
escreva ( "O total de aprovados é de: " + num_aprovados)

```

Exemplo de algoritmo com acumulador:

```

var n, i, fat : inteiro
escreva("digite um numero inteiro para cálculo de seu fatorial: ")
leia (n)
fat ← 1
para i de 1 ate n passo 1 faça
    fat ← fat * i
fim-para
escreva ("O fatorial de " + n + " é: " + fat )

```

2.1.3 Abordagem tradicional em sala

Embora a realidade atual venha mudando com o auxílio de recursos áudio visuais cada vez mais presentes em sala, a metodologia tradicional de ensino ainda é amplamente empregada na disciplina de ALP (Algoritmos e Linguagens de Programação). São aulas basicamente expositivas onde o professor apresenta um conteúdo em sala e então realiza exercícios práticos com os alunos.

"O método tradicional de ensino da disciplina de ALP começa com a apresentação da estrutura do computador e uma breve explicação de seu funcionamento. Em seguida através de montagem de fluxogramas apresenta-se ao aluno a estrutura de programas de computadores. Geralmente utiliza-se uma linguagem genérica e programas alternativos como o VISUALG. Através de exercícios, o professor apresenta ao aluno as diversas formas e estruturas que um programa possui. Para finalizar, são estudados os conceitos básicos de uma linguagem de programação (sintaxe, semântica, análise léxica e tipos de dados) escolhida pela instituição e através de exercícios, aumentando gradualmente o nível de dificuldade o aluno seria capaz de criar seus próprios programas. O método consiste basicamente de exercícios de repetição utilizando-se a seguinte estratégia: quanto mais exercícios e repetição melhor o aprendizado" (INÁCIO et al., 2014)

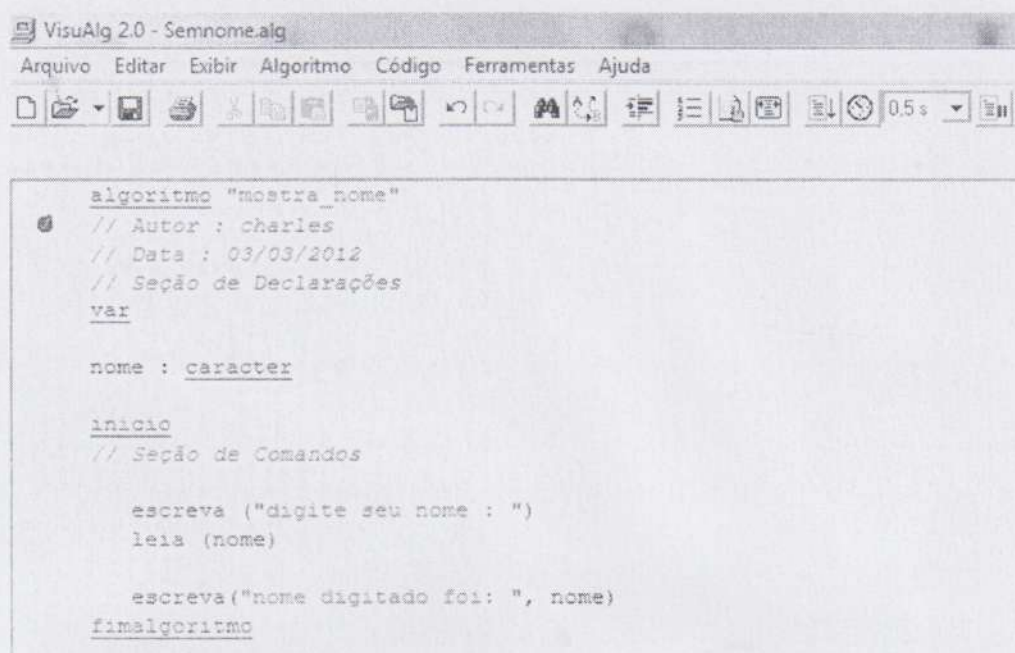


Figura 4 - Exemplo de algoritmo feito na ferramenta VisualG

Na figura 4, vemos um exemplo de algoritmo feito na ferramenta VisualG que já oferece ao aluno algumas facilidades como a visualização do estado atual de cada variável ao longo da execução do algoritmo e o uso de português estruturado (portugol), tornando-se assim um primeiro facilitador no processo de ensino.

Este modelo geralmente torna as aulas maçantes e cansativas, além de, com certa frequência, se tornarem tediosas, especialmente para alunos que nunca tiveram contato com o computador. Para estes alunos, é mais difícil desenvolver a capacidade de **abstração de dados e raciocínio lógico**, indispensáveis para o entendimento da disciplina.

O resultado disso é o baixo rendimento do aluno com consequente desmotivação, geralmente causando um efeito bola de neve, acumulando frustrações e culminando com a evasão ou reprovação da disciplina.

Há uma unanimidade em afirmar que é preciso ultrapassar as barreiras pedagógicas, inovar, deixar de ser o professor apenas "repassador" de conteúdos. (Sombrio, Haeming e Ulbricht, 2015)

2.2 Gamificação

Como parte da solução proposta, abordamos o conceito de gamificação, descrevendo a seguir o seu funcionamento.

2.2.1 O que é?

Gamificação (*gamification*) corresponde ao uso de mecanismos de jogos orientados ao objetivo de resolver problemas práticos ou de despertar engajamento entre um público específico (VIANNA et al., 2013). Em outras palavras, é o uso de técnicas e conceitos de jogos aplicado fora do contexto dos jogos ou em ambientes corporativos.

De forma mais abrangente podemos citar Karl Kapp apud Alves, 2015: "Gamificação é a utilização de mecânica, estética e pensamento baseados em games para engajar pessoas, motivar a ação, promover a aprendizagem e resolver problemas."

Segundo Alves 2015, o termo *gamification* (gamificação) surge no formato que conhecemos hoje em 2003, atribuído a Nick Pelling, programador de computadores e inventor nascido na Inglaterra, na década de 60. Ele funda uma consultoria chamada "Conunda" com o objetivo de promover o *Gamification* de produtos de consumo.

Segundo Vianna (2013), nos últimos anos, profissionais da área de desenvolvimento de jogos têm aplicado estes conceitos em áreas como a de saúde, a de educação, a de política, a de esportes, entre outras.

O termo foi popularizado em 2010 a partir de uma apresentação feita pela *game-designer* Jane McGonigal, autora do livro "A realidade em jogo: Por que os games nos tornam melhores e como eles podem mudar o mundo" (Mcgonigal, 2012). Segundo Jane, se somássemos todas as horas gastas pelos jogadores do jogo World of Warcraft, teríamos um total de 5.93 bilhões de anos gastos na resolução de problemas de um mundo virtual. Tal número deixa no ar a reflexão sobre que tipos de resultados poderíamos obter se o mesmo esforço fosse feito para resolver problemas do mundo real, como a erradicação da pobreza ou a cura de uma doença.

2.2.2 Para que serve?

O uso da gamificação tem se tornado cada vez mais comum no âmbito corporativo, devido à alta competitividade do mercado e a necessidade de se ter diferenciais para atrair usuários transformando-os em consumidores, além de manter os consumidores ativos e fieis ao produto ou serviço oferecido.

É uma forma de fugir da abordagem tradicional para resolução de um problema, motivar pessoas, mudar comportamentos, agilizar processos e tornar as tarefas mais agradáveis e até divertidas.

2.2.3 Como funciona?

Como os jogos podem ajudar o entendimento ou mesmo a adaptação ao mundo a nossa volta? Qual seria a contribuição dos jogos a uma determinada área, dado que estes promovem desenvolvimento cognitivo, afetivo e social?

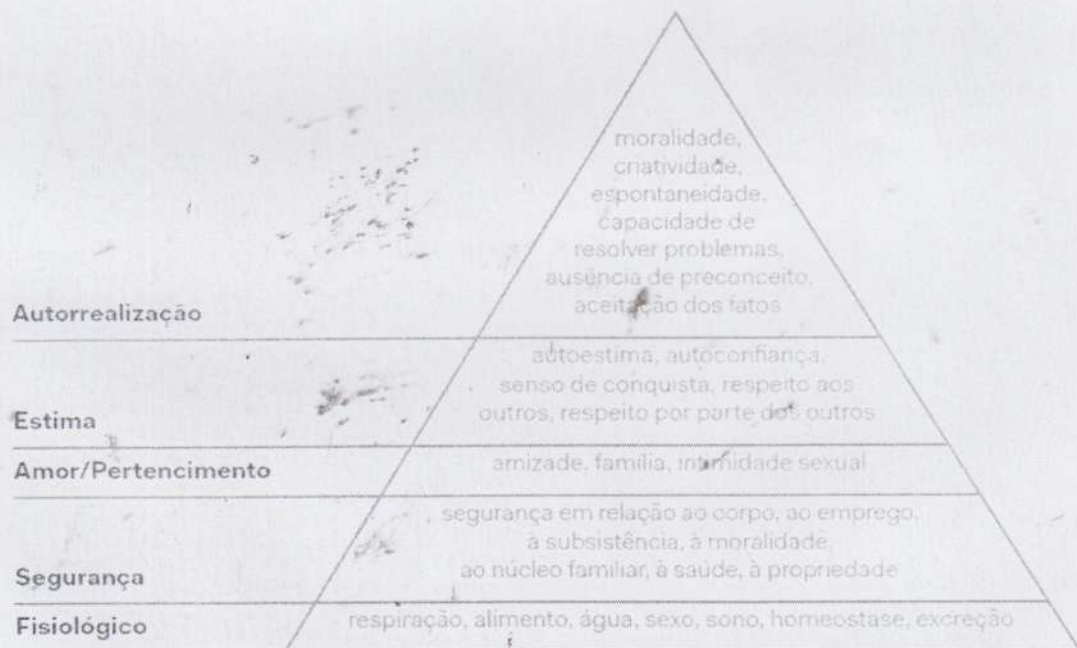


Figura 5 - Teoria da Hierarquia das Necessidades

Em 1943, Abraham Maslow propôs a célebre Teoria da Hierarquia das Necessidades, que pode ser vista na figura 5. Segundo ela, as necessidades humanas podem ser agrupadas em uma pirâmide, sugerindo que a autorrealização individual só

é bem sucedida quando a satisfação de suas necessidades ocorre em ordem crescente, das mais básicas (na base) para as mais complexas (no cume). Na base da pirâmide estão as demandas fisiológicas, como alimentar-se e respirar. Em seguida, viriam necessidades relacionadas à segurança, ao amor, à estima e, finalmente a realização pessoal, como dispor-se de um senso moral, exercer a criatividade e alcançar espontaneidade nas relações.

De acordo com essa teoria, a ação de jogar estaria localizada no cume da pirâmide, visto que representa uma atividade desejável, mas não essencial à vida. O problema é que quanto mais complexos os indivíduos, mais complexos se tornam os mecanismos necessários para se atingir a satisfação. Nesse contexto, os jogos preenchem eficientemente nossa necessidade por buscar cumprir ou conquistar objetivos.

Em geral, a aplicação do processo de gamificação envolve a criação ou a adaptação da experiência do usuário em relação a determinado produto, serviço ou processo, com o objetivo de despertar emoções positivas. À exceção dos chamados Jogos Sérios, o processo de gamificação não significa produzir um jogo, mas sim utilizar suas características mais eficientes (estéticas, mecânicas e dinâmicas) para simular os benefícios que costumam ser alcançados com eles.

Antes de se iniciar um projeto de gamificação como abordagem para um problema, considera-se uma boa prática que sejam respondidas 3 (três) perguntas essenciais para aumentar as chances de sucesso do projeto:

1. Por que um jogo irá beneficiar os usuários do serviço?
2. Quais os objetivos de negócio por trás desse jogo?
3. Como os mecanismos dos jogos farão com que os usuários atinjam os objetivos de negócio?

E afinal, por que os jogos nos atraem tão intensamente? Porque eles nos causam satisfação e consequente **motivação**. A dedicação dos jogadores e o consequente prazer ao se alcançar determinado objetivo são, portanto, elementos chave para causar uma emoção positiva. Nesse contexto, pode-se definir duas formas de motivação: intrínseca e extrínseca.

A **motivação intrínseca** é aquela na qual o usuário busca uma atividade por vontade própria, seja por achar interessante, desafiadora ou prazerosa. Ele está em busca de entretenimento, satisfazer sua curiosidade, ou mesmo obter domínio sobre algo.

A **motivação extrínseca** por outro lado, é aquela onde o usuário busca completar a atividade visando uma recompensa externa, seja ela um bem material ou mesmo psicológico, como o reconhecimento pelo sucesso.

2.2.4 Exemplos de sistemas gamificados

EDUCAÇÃO: DUOLINGO (www.duolingo.com)

O Duolingo é uma plataforma online para aprendizado de idiomas gratuito que usa um serviço de tradução baseado em *crowdsourcing*, ou seja, traduzido pelo próprio público. Ao estudar um idioma, o aluno ajuda a traduzir sites e documentos online. Iniciantes começam com frases mais simples e conforme evoluem, vão recebendo frases e textos mais complexos. Além disso, podem dar notas para traduções dos colegas, fornecendo *feedback* aos tradutores.

Ainda durante sua evolução, os estudantes recebem pontuações ao completar lições específicas, algumas delas sendo obrigatórias para continuar o progresso. Perdem pontos, ao dar respostas erradas. Outras características incluem árvores de progresso, visualização da situação atual e geral, troca de 'corações' por desbloqueio de lições bônus entre outras.

O modelo é bastante eficiente se considerarmos que os usuários estão trabalhando sem remuneração e, mesmo assim, parecem não se importar com isso.

TECNOLOGIA / SOCIAL: GAMES WITH PURPOSE (www.gameswithpurpose.org)

Utilizando *crowdsourcing* e gamificação, o objetivo é simples: treinar computadores para fornecer respostas mais assertivas a pesquisas. Em um dos games disponíveis, chamado Esp Game, duas pessoas são apresentadas à mesma imagem em suas telas e solicitadas que forneçam o maior número possível de palavras para descrevê-la. Quando a descrição fornecida por dois jogadores coincide, ambos

pontuam e o jogo passa de fase. As descrições combinadas tornam-se *tags* que ajudam a categorizar imagens, facilitando a busca em pesquisas futuras.

2.2.5 Quais os elementos de um projeto de gamificação?

Se desconsiderarmos as diferentes abordagens e detalhes específicos em diferentes jogos, todos eles compartilham 4 (quatro) características básicas: meta, regras, sistema de feedback e participação voluntária (VIANNA et al., 2013 apud McGonigal, 2012)

A **meta** é o motivo pelo qual os jogadores realizam uma atividade. Diferentemente de objetivo, a meta transcende a ideia de conclusão de uma tarefa. Significa dizer que a meta pode não ser alcançada, servindo apenas como um propósito que o jogador persegue constantemente.

As **regras** formam um conjunto de elementos que condicionam a execução do jogo, buscando promover um equilíbrio entre o desafio da resolução de um problema, sem facilitar demais a ponto de desestimular o jogador. Elas ajustam a complexidade da atividade a ser realizada pelo jogador, estimulando o pensamento estratégico. Têm a função de definir como o jogador se comportará ou como organizará suas ações para cumprir os objetivos do jogo.

O **sistema de feedback** é responsável por informar aos jogadores a sua situação com relação aos diferentes aspectos que regulam a interatividade das ações realizadas. Serve também para estimular a motivação, mantendo os jogadores constantemente informados sobre suas ações e progresso atingido ou sobre a meta estabelecida.

A **participação voluntária** significa a aceitação da meta, suas condições propostas e modelo de *feedback*. Desta forma, pode-se dizer que só existe jogo quando o jogador está disposto a se relacionar com esses elementos da maneira como foram propostos. Por isso é o principal objetivo de um projeto de gamificação.

"Demais aspectos como interatividade, suporte gráfico, narrativa, recompensas, competitividade, ambientes virtuais ou o conceito de vitória, dentre outros, são características comuns a muitos jogos, mas não definidoras. Trata-se de artifícios orientados a constituir uma relação mais próxima com as quatro características apresentadas anteriormente, ou seja, maneiras de consolidar e fortalecer os elementos expostos." (VIANNA et al., 2013)

Além das características comuns aos jogos, diversos elementos podem ser aplicados a um projeto de gamificação, entre eles: interatividade, suporte gráfico, noção de personagens, narrativa, recompensas e punições, missões e obstáculos, cooperação e competitividade, o conceito de vitória, e outros. São elementos que quando aplicados corretamente, buscam aproximar a relação entre o usuário e as quat.o características anteriores.

Diversos fatores influenciam fortemente a geração da motivação, tão visada.

IDADE, GERAÇÃO E GÊNERO são os três fatores mais influentes na criação de um jogo realmente envolvente, pois com eles podemos fazer suposições sobre o usuário, como tempo estimado de dedicação, nível de interesse por determinado tema entre outras. Isto permite a criação e ambientação de uma narrativa.

NÍVEIS DE HABILIDADE diferentes são desejáveis para ajustar a complexidade de uma tarefa aos interesses do jogador. Geralmente, em um jogo pode haver diferentes níveis de complexidade, seja pela aplicação de desafios distintos ou pela aplicação de regras ou restrições que tornam o mesmo desafio mais ou menos complicado de se resolver. O jogo só se torna motivador quando o jogador sente que suas habilidades estão adequadas para a execução das missões que ele deve cumprir. O uso de níveis, tutoriais, treinamento entre outros, são estratégias amplamente empregadas para avaliar e nivelar as habilidades do jogador para que ele se sinta apto e motivado a interagir com o jogo.

JOGADORES são os protagonistas do jogo. Qualquer jogo ou projeto gamificado parte do princípio de que existe dentro de cada um de nós um jogador, ávido por enfrentar desafios e resolver problemas. A partir do trabalho de Bartle (1996), percebe-se que os mais variados perfis de jogadores podem ser agrupados em quatro grupos principais segundo a figura abaixo.



Figura 6 - Grupos de Jogadores

Segundo o esquema mostrado na figura 6, os **predadores** são jogadores que entram em uma competição para derrotar adversários, não importando a recompensa, a intensão é ser o melhor. São mais agressivos e suas ações voltadas para assegurar a liderança. Os **realizadores** ou **conquistadores** jogam pela sensação da vitória, mesmo que o objetivo não seja tão valioso, pois sua motivação está na realização de todas as atividades apresentadas. Geralmente não se importam com interações sociais, mas são competitivos. Os **exploradores** por sua vez, estão interessados em desvendar todos os segredos do jogo, sendo bastante curiosos, buscando sempre adquirir habilidades específicas à resolução de desafios. Para estes jogadores, o mais importante é o trajeto, não a conquista em si. Finalmente, os **socializadores** são aqueles que vêem nos jogos a oportunidade de interação social. Mais importante que os próprios objetivos do jogo é o potencial em estimular e adquirir vínculos sociais de seu interesse. Os socializadores geralmente preferem jogos cooperativos e trabalho em conjunto. Cerca de 80% dos jogadores se enquadram neste grupo, segundo pesquisa feita pela Gamer DNA (gamerdna.com/quizzes/bartle-test-of-gamer--psychology).

RECOMPENSAS são o principal motivador dos jogadores a persistir no jogo até o fim ou até continuar jogando mesmo que não alcancem o seu fim. Para se desenvolver um bom sistema de recompensas, é preciso um modelo de *feedback* frequente, que na prática torna-se um meio eficiente para assegurar o engajamento no decorrer da execução de uma atividade. É preciso também que haja clareza e equilíbrio entre os valores das recompensas e o grau de dificuldade necessário para conquistá-

las. Existem vários formatos diferentes de recompensa, entre eles pode-se destacar o *status*, acesso, influência, brindes e *giftings*.

O **status** é encontrado na forma de *rankings* de pontuações com os melhores colocados, distribuição de *badges* (ou troféus) pela realização de uma dada tarefa, ou até por indicação dos próprios jogadores. Através do **acesso**, é permitido ou negado o acesso a conteúdos específicos, informações secretas, habilidades entre outros, configurando um mecanismo importante para manter os jogadores comprometidos com seus objetivos. A **influência** pode ser direta ou indireta. Será direta quando o jogador está no controle do jogo ativamente. Será indireta quando pode ser percebida voluntária ou involuntariamente com intervenções de um jogador afetando as decisões ou destino de outros jogadores. Os **brindes** são a forma mais simples de recompensa, na forma de benefícios como itens, pontos, vidas extras, entre outros dados pelas regras do jogo ao jogador. Já os **giftings** são benefícios trocados entre jogadores, promovendo maior interação social e a criação de comunidades mais fortes engajadas. A satisfação do jogador, com relação a estes dois últimos está fortemente conectada à totalidade da coleção ou no caráter aleatório da sua distribuição

MONITORAMENTO E MÉTRICAS são mecanismos para avaliar o desempenho e grau de satisfação e engajamento do jogador no decorrer do jogo. Como em qualquer implementação de sistema ou tecnologia, é fundamental que se tenha formas de avaliar resultados parciais o quanto antes, para que se possa ajustar o processo ainda em seu início. Três mecanismos principais são indispensáveis na avaliação do grau de sucesso alcançado por um projeto: adequação de missões, monitoramento da motivação e criação de métricas geradas pelo jogo.

Com respeito à **adequação de missões**, o modelo SMART (*Specific, Measurable, Achievable, Realistic and Time Bound*) proposto por Mario Herger (2013) diz que as chances de um jogador aderir a dada missão depende de o quão específica, mensurável, alcançável, realística e solucionável dentro de um prazo a missão for. Para o **monitoramento da motivação**, é recomendável o planejamento com antecedência da fadiga dos jogadores com relação aos objetivos estipulados. A ideia é tentar prever possíveis percursos com pontos chaves e inserir elementos inéditos ou diferenciados nestes pontos, afim interferir no curso da narrativa. Um exemplo comum é o uso de um

conjunto inicial de ferramentas ou habilidades disponíveis a todos no início da atividade e outro diferenciado, geralmente melhor ou mais poderoso, disponível apenas para quem atingiu certo ponto ou objetivo no jogo. Finalmente, algumas **métricas** comumente usadas são o grau de engajamento despertado, tempo dedicado à atividade e o ROI (*Return of Investment*).

2.3. Gamificação aplicada ao ensino

Para Brian J. Arnold (2014) o ponto de partida para a gamificação na educação está nos cursos online. No contexto do ensino superior, quando estudantes são solicitados a usar raciocínio crítico para resolver problemas, simulações baseadas em jogos podem ser aproveitadas em qualquer disciplina para reforçar conceitos de aplicações do mundo real. Em Madrid, por exemplo, estudantes aprendem a complexidade de políticas econômicas globais através do jogo chamado "10 Downing Street" (go.nmc.org/street) onde assumem o papel do primeiro ministro britânico e trabalham com cinco personalidades políticas para chegar a um acordo que afetará o bem estar da economia nacional. Em times de seis, entram em debates para determinar a opção política mais viável e então a colocam em prática, após uma eleição geral. Cenários como este mostram o potencial dos jogos para simular problemas reais, estimulando no estudante o desenvolvimento de um raciocínio de mais alto nível, enquanto exercita habilidades pertinentes a sua área de estudo.

Uma característica que algumas universidades de jogos vêm adotando é o sistema de *badges* ou troféus, mencionado anteriormente, como uma forma do estudante acumular documentação sobre suas habilidades, conquistas, qualidades e interesses em um formato visual público. Retornando aos tipos de jogadores, o sistema de *badges* mostra-se especialmente interessante para os do tipo conquistadores e socializadores. Enquanto o conquistador recebe uma posição de distinção visível que ele valoriza pessoalmente, o socializador vê no troféu um símbolo visual que ele pode exibir a seus parceiros e até usar como motor de novas interações sociais, discutindo sobre sua aquisição ou entrando em um grupo de jogadores na mesma posição.

Por outro lado, a natureza imediatista do *feedback* para um estudante/jogador é um dos mais importantes argumentos para o uso da gamificação no ensino. Enquanto

a vida fornece às pessoas um *feedback* inconsistente e errático para nossas ações, no ambiente educacional, este se torna mais focado, embora não necessariamente mais rápido. Um estudante presencial, pode ter que esperar até uma semana pelo resultado de um trabalho. Já um estudante online, geralmente espera em torno de 24-48 horas. Porém em um ambiente educacional gamificado, o *feedback* é quase sempre imediato, pois este é projetado para permitir ao jogador a alteração de sua abordagem buscando sempre melhores resultados.

Freifeld (2011) cita o site www.khanacademy.org de Salman Khan como um exemplo bem sucedido de aplicação da estrutura de games no ensino e sugere que também podemos nos beneficiar dos princípios listados abaixo aplicados também no ambiente de trabalho.

1. **Foco no Objetivo:** os objetivos são específicos, mensuráveis e alcançáveis através do aprendizado, fornecendo clareza quanto as expectativas e objetivos;
2. **Obtenção de feedback:** aprendizes visitam tutoriais, dão suas respostas e recebem *feedback* instantâneo se estão corretos ou não, dando a chance de seguir adiante com o curso ou tentar novamente resolver o problema buscando pistas para descobrir o 'erro';
3. **Coleta de pontos:** ao completar uma lição, estudantes recebem uma pontuação, podem aumentar seu nível e receber troféus, dando ao usuário a sensação de orgulho com cada conquista atingida;
4. **Alcançando níveis:** o site posta um quadro de liderança mostrando a proficiência individual e os diferentes tipos de conquistas. Estudantes recebem troféus à medida que progredem nos diferentes estágios de dificuldade de um assunto ou pela velocidade e eficiência em cada nível;
5. **Ritmo próprio:** cada estudante recebe os assuntos selecionados em seu próprio ritmo, tentando resolver problemas, recebendo *feedbacks*, assistindo vídeos conforme necessário, ganhando pontos e aumentando de nível. Professores assumem o papel de mentores ou técnicos e são necessários apenas quando o estudante se vê sem saída ou com uma dificuldade maior em alguma área.

6. **Métricas de aprendizado:** enquanto estudantes podem acompanhar seu próprio desempenho, professores também podem observar estatísticas vitais sobre os alunos, tais como tempo gasto em exercícios e vídeos, troféus conquistados, entre outros. Estes mecanismos fornecem resultados em tempo real com detalhamento de aprendizado granular.

"A academia de Salman Khan já entregou mais de 70 milhões de lições a pessoas de todas as idades por todo o mundo" (Freifeld (2011)).

3 METODOLOGIA

Primeiramente foi feita uma pesquisa exploratória para definição de objetivos e busca por mais informações sobre o assunto abordado.

Foi realizado coleta de materiais bibliográficos sobre os temas algoritmos, gamificação e gamificação aplicada ao ensino através de pesquisas em bibliotecas e portais de periódicos, em especial o portal ProQuest e o Google Scholar.

Os dados coletados foram estudados, filtrados por relevância ao tema escolhido e sintetizados ao longo do capítulo 2 (embasamento teórico). Após o estudo bibliográfico, foi feito um modelo inicial descritivo para uma ferramenta gamificada para apoio ao ensino de algoritmos, e a modelagem inicial de sua arquitetura, que serão discutidos mais a frente nos resultados.

O escopo da pesquisa não cobre uma varredura por ferramentas similares nem visa analisar e comparar diferentes técnicas ou estratégias aplicadas ao ensino, mas sim fornecer um panorama geral sobre o cenário atual do ensino e propor uma solução entre muitas possibilidades para a problemática discutida na introdução deste trabalho.

4 RESULTADOS

A seguir, apresentamos a pesquisa inicial e a ferramenta proposta.

4.1 A Pesquisa

Para produção dos resultados, foi feita uma breve pesquisa sobre estratégias e ferramentas de ensino, discutidas abaixo.

4.1.1 Ferramentas e estratégias de ensino

Segundo pesquisa feita por Santos e Costa (2006), os problemas encontrados na compreensão de conceitos abstratos da programação têm origem ainda na formação básica do estudante, sendo, por isso, uma das linhas de pesquisa na metodologia educacional. Ainda segundo os autores, a falta de compreensão do raciocínio lógico está entre as principais causas para o alto índice de reprovação e desistência das disciplinas, o que em parte está associado à dificuldade dos professores em administrar as atividades do grande número de alunos sob sua supervisão.

A partir do desafio da inovação no ensino, diversas mudanças têm sido consideradas nos trabalhos mais recentes sobre o ensino de algoritmos. Em geral, nota-se a presença de 3 abordagens principais buscando solucionar os problemas encontrados por professores e alunos:

- Abordagem focada em **ferramentas**: são trabalhos que apresentam alguma ferramenta computacional com o objetivo de auxiliar o processo de ensino e aprendizagem da disciplina.
- Abordagem focada em **estratégias**: trabalhos que discutem estratégias pedagógicas e metodologias de ensino ou a avaliação de competências e habilidades dos alunos.
- Abordagem focada em **ferramentas e estratégias**: são os trabalhos que unem os dois anteriores, geralmente trazendo uma análise sobre alguma estratégia suportada por uma ferramenta computacional.

Em geral, dois pontos em comum entre várias destas pesquisas dizem respeito à importância do uso de ferramentas computacionais de apoio e o trabalho em equipe como facilitador do aprendizado.

Algumas ferramentas relacionadas são apresentadas a seguir:

- ASTRAL (*Animation os Data Structures and Algorithms*): desenvolvido na UNICAMP entre 1995-1997, inicialmente para Mac e portado para Windows, é um ambiente de programação para produção de animações dos algoritmos e estruturas de dados. A ideia é permitir o aluno visualizar

graficamente, com poucas instruções, as estruturas de dados e seu funcionamento durante a execução do algoritmo.

- TED (Tutorial de Estrutura de Dados): desenvolvido na Universidade do Vale do Itajaí em 2004, consiste de um tutorial desenvolvido como software educacional, permitindo maior interação entre professor e aluno. O tutorial se divide nas várias estruturas de dados apresentadas ao longo da disciplina, permitindo que o aluno visualize um conteúdo individual sem perder a visão do todo, mostrando os algoritmos e implementações lógicas.
- TBC-AED (Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados): é um software que trata alguns tópicos básicos de programação, cobrindo-os de forma simples e direta, além de exibir um processo gráfico passo a passo destes, facilitando a visualização e entendimento dos conteúdos.

4.2 Ferramenta Proposta

Passamos então para a ferramenta em si, mostrando o procedimento utilizado para descrevê-la, sua arquitetura inicial e casos de uso cobertos pelo primeiro módulo, *CONTENT CORE*, definido adiante.

4.2.1 Definição e escopo

Alves (2015), sugere um processo de 6 etapas a serem consideradas antes de iniciarmos um projeto de gamificação: definir, arquitetar, aprender, transferir, assegurar e analisar. Como o foco deste trabalho está nas fases iniciais, de modelagem da ferramenta, a seguir descrevemos as etapas **definir**, **arquitetar**, **aprender** e **transferir**, deixando assegurar e analisar para etapas posteriores, após considerações sobre características relacionadas com a implementação da ferramenta.

4.2.1.1 Etapa 1: Definir

Objetivos: ensinar o conteúdo da disciplina de algoritmos I: organização de algoritmos, *representações de algoritmos, variáveis e tipos de dados primitivos, comandos de entrada e saída, operações aritméticas e relacionais, estruturas de decisão e de repetição.*

Problema: manter o aluno focado e motivado a continuar o processo de aprendizagem. Avaliar o rendimento do aluno e adaptar a estratégia de ensino.

Público-alvo: inicialmente, estudantes da disciplina de algoritmos I, com acesso a um laboratório de informática com computadores para rodar a aplicação proposta.

Perfil do estudante: usuários confortáveis com o uso de computadores e da internet em seu cotidiano.

Processo de ensino proposto:

1. Aplicar técnicas de gamificação, inicialmente:
 - a. Visualização de quadro geral e pontual;
 - b. Sub-divisão de tarefas, quebrando cada tópico dos objetivos em tarefas menores, detalhando para o estudante seu funcionamento e aplicação;
 - c. Sistema de missões com recompensas/penalidades;
 - d. Missões bônus e desafios (e obstáculos);
 - e. *Checkpoints* a serem atingidos;
 - f. Pontuações, troféus e *rankings*.
2. Apresentar inicialmente uma árvore/mapa geral contendo o caminho a ser trilhado pelo estudante ao longo do processo, permitindo a visualização de tudo que será coberto ao longo do caminho, bem como *checkpoints*, avaliações, testes práticos entre outros;
3. Quebrar cada tópico definido nos objetivos em missões, detalhar para o usuário/estudante e aplicar 2 (dois) métodos de questionamento:
 - a. Questionário associativo: realização de analogias ao conteúdo em questão para entendimento do mesmo;
 - b. Questionário técnico: realização de exercícios de fixação práticos para pontuar e avaliar o aprendizado do estudante.

4. Aplicar sistema de pontuação e troféus para cada missão ou grupo de missões cumpridas e relatório de acompanhamento parcial;
5. Aplicar sistema de recompensa/penalidades por missão cumprida;
6. Permitir repetir missões para melhorar resultados.

4.2.1.2 Etapa 2: Arquitetar

Com o intuito de manter a ferramenta compatível com expansões, não só o conteúdo da disciplina como também as regras de avaliação podem ser parametrizadas e armazenadas no banco de dados, permitindo algum nível de meta-programação. Assim, os conceitos pertinentes e aplicáveis ao nosso tema podem ser agrupados e transformados em entidades do nosso domínio de aplicação, de acordo com as palavras-chaves abaixo.

- Panorama: mapa, carreira/campanha, situação atual, situação geral;
- Assuntos: conteúdo, teoria, descrição, detalhamento;
- Lições: missão, regra, analogia, associação;
- Atividades: exercício, desafio, problema;
- Questionário: pergunta, resposta, resultado, *feedback*;
- Gerenciamento: avaliação, relatório;
- Pontuação: *ranking*, troféu, recompensa, penalidade, bônus;
- Interatividade: obstáculo, ajuda, dica;
- Equipe: competição, colaboração.

4.2.1.3 Etapa 3: Aprender

O modelo de ensino proposto consiste em duas fases: associação e fixação.

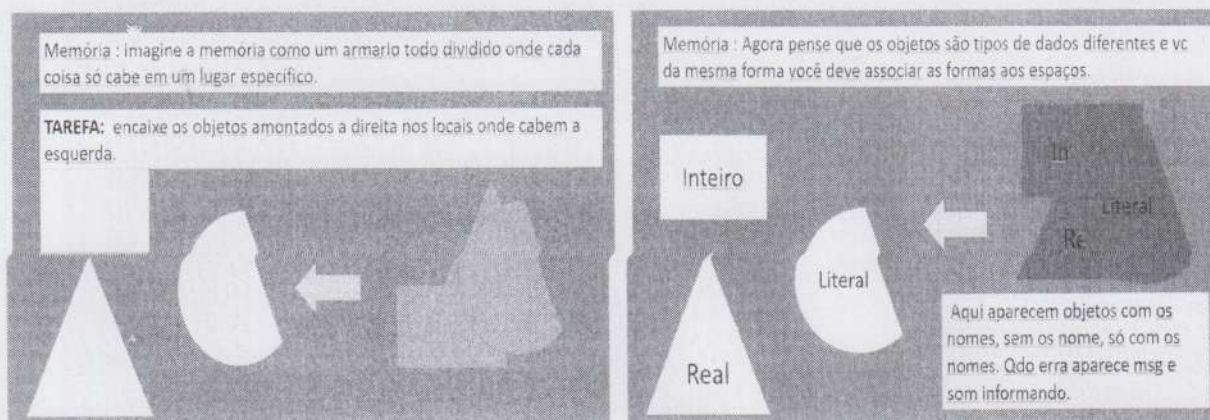


Figura 7 - Fase de Associação no modelo de ensino proposto

A fase de associação, descrita na figura 7, se dá por meio de analogias entre o conteúdo que buscamos ensinar e outros domínios já conhecidos pelo aluno, através do uso de comparações. No exemplo acima, estamos fazendo analogia entre tipos de dados e formas geométricas para encaixar.

A fase de fixação ocorre com a aplicação do conteúdo associado anteriormente em questionários simples envolvendo o tema abordado para aplicar e fixar o conhecimento, conforme exemplificado na figura 8, abaixo.

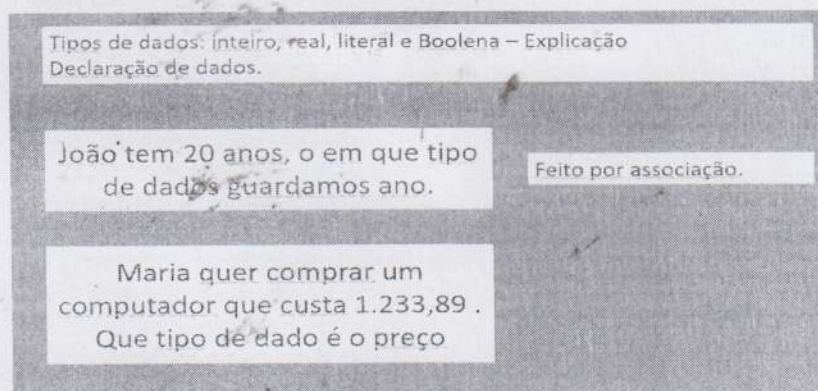


Figura 8 - Fase de Fixação no modelo de ensino proposto

4.2.1.4 Etapa 4: Transferir

Os exercícios de fixação são aplicados a problemas reais ou desafios, cadastrados no banco de problemas, a partir do módulo *CONTENT CORE*, que será

discutido adiante. Dessa forma permitimos que o estudante pratique o conhecimento abordado na lição e visualize como o mesmo pode ser aplicado no mundo real.

4.2.2 Modelagem

A modelagem inicial definiu a arquitetura base, dos componentes principais da ferramenta e os casos de uso de um de seus módulos.

4.2.2.1 Arquitetura Base

A ferramenta será composta por dois módulos independentes, mas que trabalharão em conjunto para desempenhar seu papel conforme os objetivos pretendidos. O diagrama de componentes da figura 9 mostra esta arquitetura. Em seguida, detalhamos cada componente.

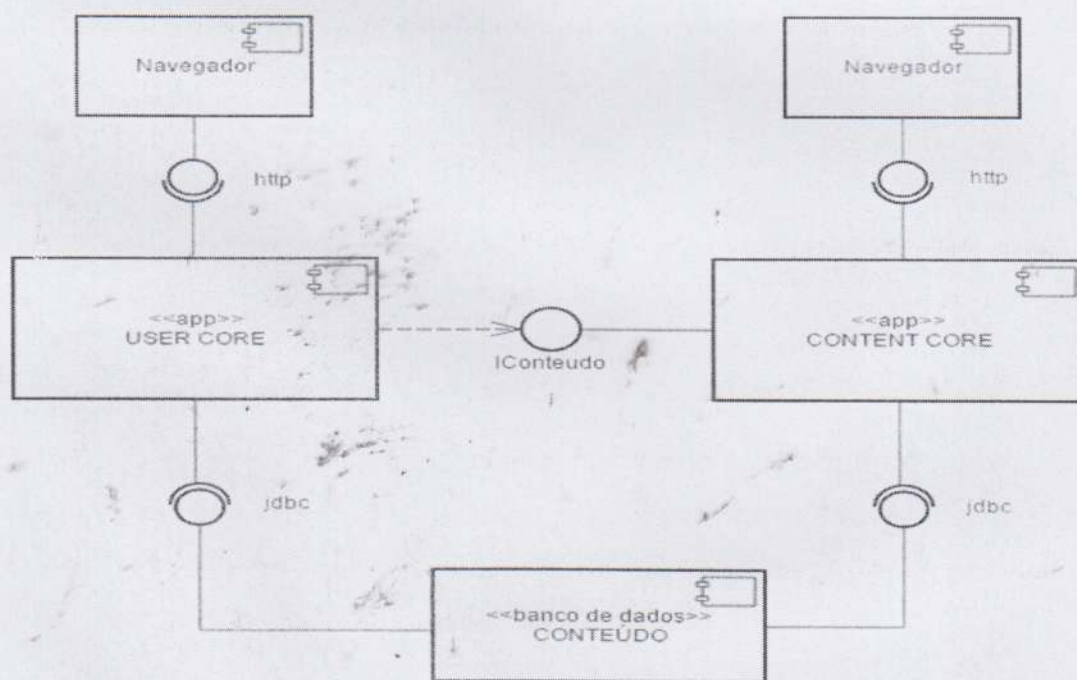


Figura 9 - Diagrama de Componentes da Ferramenta

1. **CONTENT CORE**: módulo de 'back-end' da ferramenta, onde professores / administradores podem gerenciar o conteúdo disponível para exibição ao usuário/estudante;
 - a. **Gerenciador de Conteúdo**: fornece uma camada de abstração com o banco de dados, permitindo cadastro de diferentes formas de conteúdo, a serem usados nas oficinas e na interface do usuário;
 - b. **CONTEÚDO**: base de dados contendo conteúdos explorados por tema, problemas relacionados, perguntas, respostas, desafios, regras, associações, analogias, figuras/animações, dicas, descrições;
 - c. **Oficina de lições/missões**: onde o professor pode montar as lições/missões de acordo com os conteúdos cadastrados. Pode definir as regras das missões, as pontuações, agrupar associações e analogias, escolher ajudas etc.;
 - d. **Oficina de trilhas/cursos**: onde o professor pode selecionar conjuntos de missões para compor uma trilha (ou mapa) que será usado como mecanismo de ensino de um determinado conteúdo. Permite agrupar conteúdos e temas, em capítulos a serem seguidos pelo estudante/jogador.
2. **USER CORE** (*front-end*): responsável por controlar a interação do estudante/jogador com o sistema gamificado, composto por controladores e serviços que manipulam o conteúdo e o apresenta ao usuário;
 - a. **Gerenciador de Usuários**: controla o acesso e monitora as atividades do usuário
 - i. **Controladores e Serviços**: gerenciam as interações do usuário com o sistema, armazenando seu progresso e estado, além calcular resultados de acordo com as configurações das progressões e missões, devolvendo dados na forma de *feedbacks*.
 - b. **Visor do Usuário**: a tela como será apresentada ao estudante/jogador
 - i. **Missão**: mostra o detalhamento da missão atual, tópico abordado, objetivo, dicas etc;

- ii. **Situação atual e geral:** mostra dados da situação geral do usuário em relação à progressão atual, dados do usuário em relação à missão atual, tempo gasto se for necessário etc;
- iii. **Console:** onde o usuário informará suas respostas para cada missão/objetivo.

4.2.2.2 Descrição de Casos de Uso do Módulo de Conteúdo

O diagrama de casos de uso mostrado na figura 10, abaixo, traz a descrição geral do módulo *CONTENT CORE*, descrito anteriormente. Através do diagrama, vemos como o professor poderá usar a ferramenta para criar e gerenciar conteúdo 'gamificado' para seus alunos, além de acompanhar o desempenho dos estudantes.

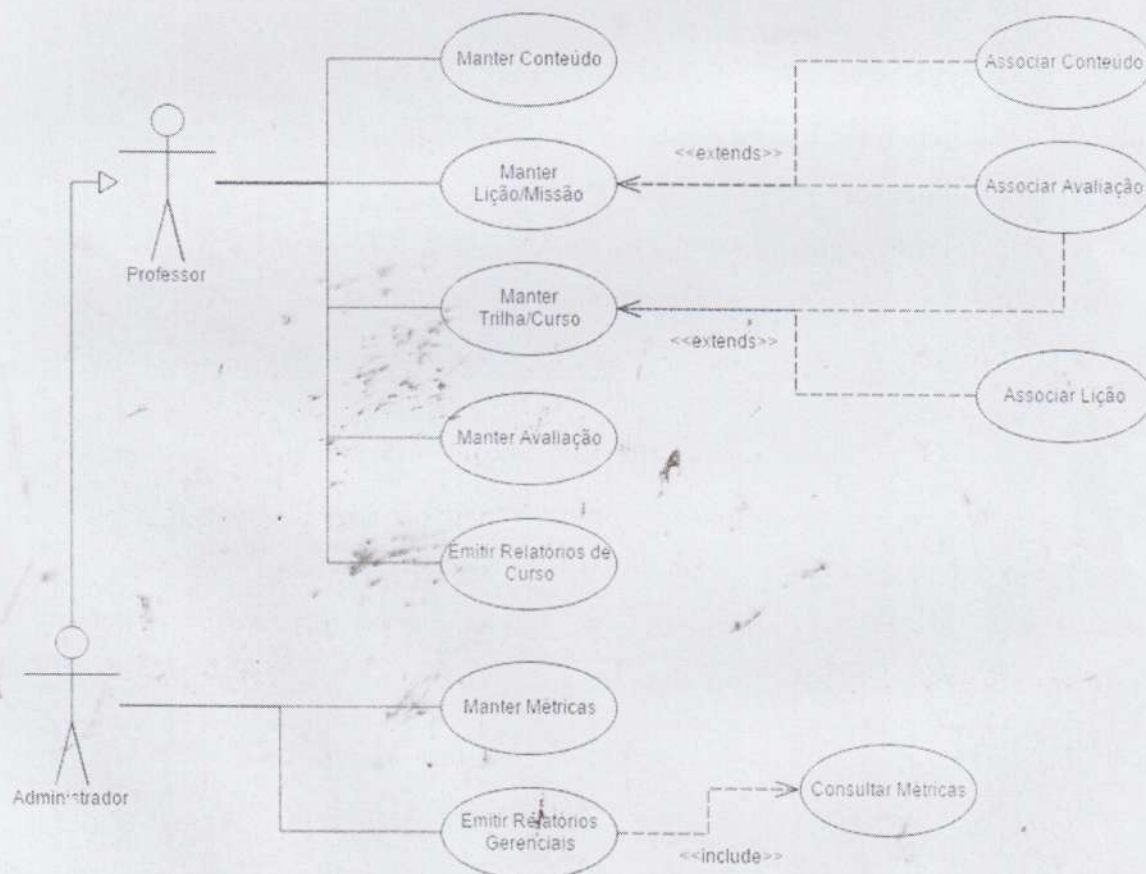


Figura 10 - Diagrama de Casos de Usos do Módulo de Conteúdo

Abaixo segue a descrição de cada caso de uso.

Nome do caso de uso	Manter Conteúdo
Sumário	O usuário cadastra, edita visualiza e remove conteúdos diversos na base de dados da aplicação.
Ator primário	Professor
Atores secundários	-
Precondições	O professor está previamente cadastrado no sistema.
Fluxo Principal	1. O professor acessa o sistema; 2. O sistema exibe opções de conteúdo disponíveis; 3. O professor seleciona o tipo de conteúdo que deseja gerenciar; 4. O professor escolhe operação que deseja executar; 5. O sistema apresenta informações e formulários, quando houver necessidade, para cada operação; 6. O professor preenche os formulários ou visualiza dados. a. Caso tenha sido preenchido um formulário, o sistema valida-o e processa a operação devolvendo o resultado para o usuário.
Fluxo Alternativo	-
Pós-condições	A operação selecionada para o tipo de conteúdo selecionado foi executada corretamente.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda.
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Manter Lição/Missão
Sumário	O professor seleciona diferentes tipos de conteúdos cadastrados para montar uma lição que será uma missão no

	módulo USER CORE, descrito anteriormente.
Ator primário	Professor
Atores secundários	-
Precondições	Os conteúdos já estão cadastrados no sistema
Fluxo Principal	1. Professor seleciona a operação desejada; 2. Sistema exibe tipos de conteúdo atualmente cadastrados; 3. Professor seleciona tipos de conteúdo que deseja incluir na lição; 4. Sistema exibe lista de conteúdos cadastrados para cada tipo selecionado; 5. Se professor selecionou cadastrar ou editar lição, deverá selecionar quais conteúdos de cada tipo serão incluídos na lição; a. Professor organiza a lição: i. Ordenação de sequência de conteúdos a serem exibidos ou executados; 6. Sistema processa operação selecionada pelo professor.
Fluxo Alternativo	A partir de (5), o professor pode opcionalmente: i. definir associações entre conteúdos, descrito no caso de uso "Associar Conteúdo"; ii. definir um método de avaliação, descrito no caso de uso "Associar Avaliação".
Pós-condições	A operação selecionada para a lição selecionada foi executada corretamente.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Manter Trilha/Curso
Sumário	O professor seleciona diferentes lições cadastradas para montar uma trilha que será um curso no módulo USER CORE, descrito anteriormente.
Ator primário	Professor
Atores secundários	-
Precondições	As lições já estão cadastradas no sistema
Fluxo Principal	1. Professor seleciona operação desejada; 2. Para cadastro e edição, sistema apresenta lições cadastradas; 3. Professor seleciona lições que deseja incluir na trilha, ou remover, no caso de edição; 4. Professor organiza a trilha, definindo a ordem das lições; 5. Sistema processa operação selecionada pelo professor.
Fluxo Alternativo	A partir de (4), o professor pode opcionalmente: iii. definir associações entre lições, descrito no caso de uso "Associar Lição"; iv. definir um método de avaliação, descrito no caso de uso "Associar Avaliação".
Pós-condições	A operação selecionada para a trilha selecionado foi executada corretamente.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel

Data	04/12/2015
-------------	------------

Nome do caso de uso	Manter Avaliação
Sumário	O professor define um mecanismo de avaliação, composto por regras associando entrada, processamento e saída de dados, que poderá ser empregado em lições e trilhas.
Ator primário	Professor
Atores secundários	-
Precondições	As regras já estão cadastradas no sistema.
Fluxo Principal	1. Professor seleciona operação desejada; 2. Sistema exibe formulários para operação selecionada; 3. Professor preenche formulários para operação desejada; 4. Sistema processa operação; 5. Sistema devolve resultado para o professor.
Fluxo Alternativo	-
Pós-condições	A operação selecionada para a trilha selecionado foi executada corretamente.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Associar Conteúdo
Sumário	O professor realiza associação entre dois ou mais conteúdos, opcionalmente definindo também regras de associação, para tornar uma lição mais ou menos complexa.
Ator primário	Professor
Atores secundários	Módulo CONTENT CORE
Precondições	Os conteúdos e regras já estão cadastradas no sistema.
Fluxo Principal	1. Sistema exibe conteúdos previamente selecionados; 2. Professor seleciona 2 ou mais conteúdos para fazer uma associação; 3. Opcionalmente professor pode selecionar também regras para associação; 4. Sistema processa operação solicitada; 5. Sistema devolve resultado para usuário.
Fluxo Alternativo	-
Pós-condições	A associação de conteúdos selecionados foi executada corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda.
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Associar Avaliação
----------------------------	--------------------

Sumário	O professor faz associação entre um ou mais conteúdos, e uma avaliação, opcionalmente definindo também regras de associação, para tornar uma lição mais ou menos complexa.
Ator primário	Professor
Atores secundários	Módulo CONTENT CORE
Precondições	Os conteúdos, avaliações e regras já estão cadastrados no sistema.
Fluxo Principal	1. Sistema exibe conteúdos previamente selecionados 2. Professor seleciona um ou mais conteúdos para associação; 3. Sistema exibe avaliações disponíveis para aplicação; 4. Professor seleciona uma avaliação para a associação; 5. Opcionalmente professor pode selecionar regras para aplicar à associação; 6. Sistema processa operação.
Fluxo Alternativo	-
Pós-condições	A associação entre conteúdos selecionados e avaliação foi executada corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Associar Lição
Sumário	O professor faz associação entre uma ou mais lições, opcionalmente definindo também regras de associação, para tornar uma trilha mais ou menos complexa.
Ator primário	Professor
Atores secundários	Módulo CONTENT CORE
Precondições	As lições e regras já estão cadastrados no sistema.
Fluxo Principal	1. Sistema exibe lições previamente selecionados; 2. Professor seleciona 2 ou mais lições para fazer uma associação; 3. Opcionalmente professor pode selecionar também regras para associação; 4. Sistema processa operação solicitada; 5. Sistema devolve resultado para usuário.
Fluxo Alternativo	-
Pós-condições	A associação entre lições selecionadas foi executada corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Emitir Relatórios de Curso
Sumário	O professor seleciona filtros para emissão de relatórios de

	acompanhamento junto aos estudantes em suas trilhas cadastradas.
Ator primário	Professor
Atores secundários	-
Precondições	Existem trilhas cadastradas pelo professor no sistema.
Fluxo Principal	1. Sistema exibe trilhas disponíveis para o professor; 2. Professor seleciona trilha para emissão de relatório; 3. Professor seleciona filtros para a trilha selecionada; 4. Sistema processa operação solicitada; 5. Sistema devolve resultado para usuário.
Fluxo Alternativo	-
Pós-condições	O relatório da trilha com os filtros selecionados é emitido corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	Para cada tipo de conteúdo disponível para gerenciamento, serão apresentados formulários dinâmicos, com campos obrigatórios ou não, influenciando os filtros para relatórios.
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Manter Métricas
Sumário	O administrador define e gerencia métricas para avaliação estatística da ferramenta ao longo do uso.
Ator primário	Administrador

Atores secundários	-
Precondições	Usuário administrador deve estar cadastrado previamente.
Fluxo Principal	1. O administrador acessa o sistema; 2. O sistema exibe operações disponíveis para métricas; 3. O administrador seleciona operação desejada; 4. O sistema apresenta informações e formulários, quando houver necessidade, para cada operação; 5. O administrador preenche os formulários ou visualiza dados. a. Caso tenha sido preenchido um formulário, o sistema valida o mesmo e processa a operação devolvendo o resultado para o usuário.
Fluxo Alternativo	-
Pós-condições	A operação selecionada para métrica é executada corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	-
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Emitir Relatórios Gerenciais
Suário	O administrador seleciona filtros de relatórios gerenciais usando as métricas disponíveis no sistema e emite um relatório com estatísticas usando os filtros selecionados.

Ator primário	Administrador
Atores secundários	-
Precondições	Usuário administrador deve estar cadastrado previamente. Conteúdos, Métricas, trilhas e lições devem estar previamente cadastrados para que o relatório retorne resultados relevantes.
Fluxo Principal	1. O administrador acessa o sistema; 2. O sistema exibe opções disponíveis para relatórios gerenciais; 3. O administrador seleciona opções desejados; 4. O sistema consulta métricas disponíveis, conforme descrição do caso de uso "Consultar Métricas"; 5. Sistema oferece filtros de métricas disponíveis conforme resultado do passo 4; 6. Administrador seleciona filtros desejados; 7. Sistema emite relatório ou exibe mensagem informativa sobre o status da solicitação.
Fluxo Alternativo	-
Pós-condições	O relatório gerencial é emitido corretamente ou mensagem de erro informativa é devolvida para usuário.
Regras de negócio	-
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

Nome do caso de uso	Consultar Métricas
Sumário	O administrador busca métricas disponíveis no sistema para avaliação interna e posterior aplicação na emissão de

	relatórios gerenciais.
Ator primário	Administrador
Atores secundários	Módulo CONTENT CORE
Precondições	Usuário administrador deve estar cadastrado previamente. Métricas devem estar previamente cadastradas.
Fluxo Principal	1. Administrador seleciona busca por métricas; 2. Sistema oferece opções de busca; 3. Administrador seleciona opções desejadas; 4. Sistema consulta métricas cadastradas de acordo com opções selecionadas; 5. Sistema retorna métricas encontradas para usuário; 6. Usuário seleciona resultado desejado para conclusão do caso de uso "Emitir Relatórios Gerenciais".
Fluxo Alternativo	-
Pós-condições	O sistema consulta métricas disponíveis de acordo com opções selecionadas e devolve resultado encontrado ou mensagem de erro informativa.
Regras de negócio	-
Requisitos não funcionais	Não definidos ainda
Autor	Marcio Davi Martins Maciel
Data	04/12/2015

5 CONCLUSÃO

Neste trabalho, vimos um pouco sobre a problemática envolvida com a metodologia atual para o ensino das disciplinas de algoritmos e como ela pode dificultar e desencorajar estudantes iniciantes na área de informática, causando evasão,

reprovação ou mesmo desistência de cursos de TI. Em seguida apresentamos um embasamento teórico sobre os principais tópicos comumente ministrados na disciplina de algoritmos I, e então introduzimos o conceito de gamificação como ferramenta de apoio ao ensino propondo sua aplicação na solução do problema da dificuldade no aprendizado e desmotivação, enfrentados pelo aluno.

Neste âmbito, apresentamos a definição de gamificação, os principais conceitos relacionados e seus componentes, funcionamento, considerações a serem feitas e objetivos de sua aplicação. Mostramos o exemplo bem sucedido do uso da gamificação no ensino com a Academia Online de Salman Khan (<https://www.khanacademy.org>), destacando alguns pontos simples do projeto que impactam fortemente no sucesso da aplicação.

Finalmente, propomos uma ferramenta online para ajudar professores a reformular suas estratégias de ensino através do uso da gamificação no dia a dia do aluno, com o objetivo de aumentar tanto seu interesse pela disciplina como seu grau de aprendizado e motivação para continuar os estudos. Apresentamos algumas etapas do processo de gamificação da ferramenta e concluímos com a modelagem inicial da arquitetura da ferramenta e uma descrição do módulo de conteúdo (*CONTENT CORE*), apresentando os casos de uso que descrevem seu funcionamento.

Como trabalhos futuros, a ferramenta será melhor especificada incluindo detalhes específicos das tecnologias usadas em sua implementação, será modelado o módulo de interface do usuário (*USER CORE*) e a integração entre os dois módulos, concluindo o mínimo necessário para sua utilização completa tanto por alunos como por professores. Espera-se com isso, inovar a metodologia de ensino de algoritmos localmente para aumentar o interesse, a motivação e o aprendizado dos alunos da disciplina em questão.

6 REFERÊNCIAS BIBLIOGRÁFICAS

As referências foram classificadas em 2 grupos: o levantamento completo preliminar; e as referências selecionadas para a produção do texto.

6.1 Levantamento bibliográfico preliminar

ALVES JÚNIOR, Vetrúvio . **A gamificação no processo de desenvolvimento e aprendizagem da inovação**. Disponível em: < <http://www.tcc.sc.usp.br>> Acesso em 05 out 2015.

ALVES, Flora. **Gamification: como criar experiências de aprendizagem engajadora: um guia completo: do conceito a prática**. 2 Edição. São Paulo: DVS Editora, 2015.

Arnold, Brian J. **Gamification in Education**. In: Proceedings of American Society of Business and Behavioral Sciences (ASBBS), Volume 21, Number 1, p. 32-39. 2014.

BANFIELD, James, WILKERSON, Brad. **Increasing Student Intrinsic Motivation And Self-Efficacy Through Gamification Pedagogy**. In: Contemporary Issues In Education Research – Fourth Quarter, 2014, Volume 7, Number 4, p. 291-298. 2014.

BARTLE, Richard. "Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs". The Journal of Virtual Environments 1 (1), 1996. Disponível em brandeis.edu/pubs/jove/HTML/v1/bartle.html

BURKE, J.R. **Proofs and algorithms: an introduction to logic and computability theory**. Choice, Middletown, v. 49, n. 1, p. 153, 09 2011.

CHEONG, Christopher; Filippou, Justin; Cheong, France. **Towards the Gamification of Learning: Investigating Student Perceptions of Game Elements**. 2014. In: Journal of Information Systems Education, vol 25 no.3, p. 233-244.

CORMEN, Thomas H. et al. **Algoritmos: Teoria e Prática**. 3. ed. [s.l.]: Campus, 2012. 944 p.

EJSING-DUUN, Stine and KAROFF, Helle Skovbjerg. **Gamification of a Higher Education Course: What's the fun in That?** In: Proceedings of the 8th European Conference on Game Based Learning - ECGBL 2014, p. 92-98. 2014. ISBN: 978-1-910109-55-1.

FABRICATORE, Carlo and López, Ximena. **Using Gameplay Patterns to Gamify Learning Experiences**. In: Proceedings of the 8th European Conference on Game Based Learning. 2014. Academic Conferences and Publishing International Limited, Reading, UK, pp. 110-117. ISBN 978-1-910309-55-1

FIGUEIREDO, Karen da silva. **Proposta de Gamificação de Disciplinas em um Curso de Sistemas de Informação**. 2015. XI Brazilian Symposium on Information System, Goiânia, GO, Maio 26-29, 2015. Disponível em: < <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2015/081.pdf>> Acesso em 05 out. 2015.

FREIFELD Lorri. **MAKING LEARNING FUN**. In: Training Magazine Digital, Vol Nov/Dec 2011, p. 70-71. 2011. Disponível em < <https://trainingmag.com/content/talent-tips-making-learning-fun> > Acesso em 31 out. 2015.

HERGER, Mario, Kumar, Janaki. **Gamification at Work: Designing Engaging Business Software**. The Interaction Design Foundation, 2013.

INÁCIO, Flamarion Assis Jeronimo et al. **O MODELO TRADICIONAL DE ENSINO APLICADO À DISCIPLINA DE ALGORITMOS: ESTUDO DE CASO E PROPOSTA PARA APERFEIÇOAMENTO DO MÉTODO - "É POSSÍVEL DESENVOLVER RACIOCÍNIO LÓGICO UTILIZANDO A TEORIA DAS INTELIGÊNCIAS MÚLTIPLAS?"**. Enciclopédia Biosfera, Goiânia, v. 10, n. 19, p.361-369, nov. 2014.

KIM, Bohyun. **Understanding Gamification**, chapter 2 and 5. ISBN: 978-0-8389-5953-4. Publicado por: American Library Association. 2015.

MANN, C.J.H. **New Edition of Algorithms Book -- Introduction to Algorithms**. Kybernetes, London, v. 32, n. 3, p. 410, 2003.

MARINS, Diego Ribeiro. **Um processo de gamificação baseado na teoria da autodeterminação**. 2013. Tese de Doutorado. Universidade Federal do Rio de Janeiro. Disponível em: < <http://www.cos.ufri.br/uploadfile/1387465246.pdf>> Aceso em 05 out 2015.

MCGONIGAL, Jane. **A realidade em jogo**: porque os games nos tornam melhores e como eles podem mudar o mundo. Rio de Janeiro: Best Seller, 2012.

MITCHELL, Nicholas, DANINO, Nicky and MAY, Lesle. **Motivation and Manipulation**: A Gamification Approach to Influencing Undergraduate Attitudes in Computing. In: Proceedings fo the 7th European Conference on Games Based Learning ECGBL, p. 394-400. 2013.

MOZELIUS, Peter et al. **Let the Students Contruct Their own fun And Knowledge** □ Learning to Program by Building Computer Games. In: Conference: ECGBL 2013, The 7th European Conference on Games Based Learning, At Porto, Portugal, Volume: 7, p. 418-426. 2013.

QUADROS, Gerson Bruno Forgiarini de. **Gamificando os processos de ensino na rede**. In: Anais do Congresso Nacional Universidade, EAD e Software Livre. 2013.. v. 2, n. 3 (2012). Disponível em: <<http://periodicos.letras.ufmg.br/index.php/ueadsl/article/view/3913/0> > Acesso em 05 out 2015.

SANTOS, R. P. dos; COSTA, H. A. X. **Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática**. INFOCOMP: Journal of Computer Science, Lavras, v. 5, n. 1, p. 41-50, Mar. 2006. Disponível em: <<http://repositorio.ufla.br/handle/1/9841>>

SOMBRIO, Graziela de Souza; HAEMING, Waléria Külkamp; ULBRICHT, Vania Ribas. **Aprendizagem Criativa na Educação Utilizando Jogos e Gamificação**. Revista Hipe texto, v. 4, n. 2, p. 37-49, 2015. Disponível em: <[http://www.latec.ufrj.br/revistas/index.php?journal=hipertexto&page=article&op=view&path\[\]=642](http://www.latec.ufrj.br/revistas/index.php?journal=hipertexto&page=article&op=view&path[]=642)> Acesso em 05 out 2015

VIANNA, Ysmar et al. **Gamification, Inc**: Como reinventar empresas a partir de jogos. Rio de Janeiro: Mjv Press, 2013. 116 p.

VICKERY, C. **A concise and practical introduction to programming algorithms in Java**. Choice, Middletown, v. 47, n. 2, p. 346, 10 2009. Sherry Robinson. **TAKING A CHANCE: INTRODUCING UNCERTAINTY INTO LEARNING GAMES**. In: Proceedings of the Academy of Educational Leadership, Volume 17, Number 2, p. 21-26. 2012.

6.2 Referências Utilizadas

ALVES, Flora. **Gamification: como criar experiências de aprendizagem engajadora: um guia completo: do conceito a prática**. 2 Edição. São Paulo: DVS Editora, 2015.

Arnold, Brian J. **Gamification in Education**. In: Proceedings of American Society of Business and Behavioral Sciences (ASBBS), Volume 21, Number 1, p. 32-39. 2014.

CORMEN, Thomas H. et al. **Algoritmos: Teoria e Prática**. 3. ed. [s.l.]: Campus, 2012. 944 p.

FREIFELD Lorri. **MAKING LEARNING FUN**. In: Training Magazine Digital, Vol Nov/Dec 2011, p. 70-71. 2011. Disponível em < <https://trainingmag.com/content/talent-tips-making-learning-fun> > Acesso em 31 out. 2015.

INÁCIO, Flamarion Assis Jeronimo et al. **O MODELO TRADICIONAL DE ENSINO APLICADO À DISCIPLINA DE ALGORITMOS: ESTUDO DE CASO E PROPOSTA PARA APERFEIÇOAMENTO DO MÉTODO - "É POSSÍVEL DESENVOLVER RACIOCÍNIO LÓGICO UTILIZANDO A TEORIA DAS INTELIGÊNCIAS MÚLTIPLAS?"**. Enciclopédia Biosfera, Goiânia, v. 10, n. 19, p.361-369, nov. 2014.

Mcgonigal, Jane. **A realidade em jogo: porque os games nos tornam melhores e como eles podem mudar o mundo**. Rio de Janeiro: Best Seller, 2012.

SANTOS, R. P. dos; COSTA, H. A. X. **Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em**

computação e informática. INFOCOMP: Journal of Computer Science, Lavras, v. 5, n. 1, p. 41-50, Mar. 2006. Disponível em: <<http://repositorio.ufla.br/handle/1/9841>>

SOMBRIO, Graziela de Souza; HAEMING, Waléria Kulkamp; ULBRICHT, Vania Ribas. **Aprendizagem Criativa na Educação Utilizando Jogos e Gamificação.** Revista Hipertexto, v. 4, n. 2, p. 37-49, 2015. Disponível em: <[http://www.latec.ufrj.br/revistas/index.php?journal=hipertexto&page=article&op=view&path\[\]=642](http://www.latec.ufrj.br/revistas/index.php?journal=hipertexto&page=article&op=view&path[]=642)> Acesso em 05 out 2015

VIANNA, Ysmar et al. **Gamification, Inc:** Como reinventar empresas a partir de jogos. Rio de Janeiro: Mjv Press, 2013. 116 p.