

**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia Software**

**Ambiente de aprendizagem Gamificado para ensino de
Algoritmos.**

**Autor: Wilker Mesquita de Assunção
Orientador: Prof. Dr. Sérgio Antônio Andrade Freitas**

**Brasília, DF
2016**



WILKER MESQUITA DE ASSUNÇÃO

Ambiente de aprendizagem Gamificado para ensino de Algoritmos.

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Orientador: Doutor Sérgio Antônio Andrade de Freitas

**Brasília, DF
2016**

CIP – Catalogação Internacional da Publicação*

Assunção, Wilker Mesquita.

Ambiente de aprendizagem Gamificado para ensino de Algoritmos./ Wilker Mesquita de Assunção (em ordem normal). Brasília: UnB, 2016. 103 p.: il. 29,5 cm.

Monografia (Graduação) – Universidade de Brasília
Faculdade do Gama, Brasília, 2016. Orientação: Sérgio Antônio
Andrade de Freitas.

1. Algoritmos. 2. Ambiente Virtual. 3. Gamificação I. Freitas,
Sérgio Antônio Andrade de. II. Doutor.

CDU Classificação

**REGULAMENTO E NORMA PARA REDAÇÃO DE RELATÓRIOS DE PROJETOS
DE GRADUAÇÃO FACULDADE DO GAMA - FGA**

Wilker Mesquita de Assunção

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software da Faculdade UnB Gama - FGA, da Universidade de Brasília, em (___/___/___) apresentada e aprovada pela banca examinadora abaixo assinada:

Prof. Doutor: Sérgio Antônio Andrade de Freitas, UnB/ FGA
Orientador

Prof. Doutor: Wander Cleber Pereira da Silva, UnB/ FGA
Membro Convidado

Prof.^a. Doutora: Carla Silva Rocha Aguiar, UnB/ FGA
Membro Convidado

Brasília, DF
2016

Esse trabalho é dedicado às pessoas que buscam, em pequenas atitudes, tornar o mundo melhor.

AGRADECIMENTOS

À minha mãe Silvia Regina, por todo seu amor, dedicação e em especial por sempre estar ao meu lado.

Ao meu pai Weber, por todo seu amor, dedicação e em especial por me mostrar que o estudo é uma das coisas mais importantes da vida.

A Sabrina Lunara, por todo seu amor, dedicação e em especial por me apoiar, alegrar e sonhar comigo durante toda a minha graduação.

Ao professor Sérgio, por todas as experiências e conhecimentos compartilhados em sua ótima orientação.

À professora Milene que me apoiou e me aconselhou em momentos de dificuldade.

A todos os amigos do curso que de alguma forma tornaram minha jornada mais leve e prazerosa.

Aprender sem pensar é trabalho perdido;
pensar sem aprender é perigoso.
Confúcio.

RESUMO

A utilização de softwares vem se tornando cada vez mais populares na sociedade como um todo, o mesmo pode ser dito para ambientes de aprendizagem virtual. Os métodos de ensino e aprendizagem vêm evoluindo juntamente com os avanços da tecnologia, novas técnicas surgem com o intuito de tirar cada vez mais proveito desses avanços com o fim de melhorar o ensino e aprendizagem. Este trabalho traz a adaptação de um ambiente virtual de aprendizado para ensino de algoritmos, utilizando técnicas de Gamificação proposta por Yukai Chou juntamente com os métodos pedagógicos propostos por Salman Khan. O ambiente de aprendizado virtual Moodle foi estruturado com um curso de algoritmos inteiramente gamificado, utilizando como suporte os softwares Funifier e Scratch. O curso online tem duração de 1 mês, dividido em tarefas semanais que ensinam contextos básicos de algoritmos computacionais, trazendo elementos consolidados no campo da gamificação, a exemplo dos pontos, medalhas e classificações.

Palavras-chave: Ambiente virtual de ensino. Gamificação. Algoritmos. Ensino e aprendizagem, Engenharia de software.

ABSTRACT

Using software has become even more popular in the general society, we can say the same to the virtual learning environments. The teaching methods are evolving along with the technological advances, new techniques arises with the intention of taking more and more advantage of these advances in order to improve the learning and teaching. This work brings an adaptation of a virtual learning environment to teach algorithms, using techniques of gamification proposed by Yukai Chou alongside the pedagogics methods proposed by Salman Khan. The Moodle virtual learning environment was structured with the course complete gamified, using as support the Funifier and Scratch Softwares. The online course has the duration of 1 month, divided in weekly tasks that teach basic concepts of computational algorithm, bringing consolidated elements in the field of gamification, for example points, badges and leaderboards.

Keywords: Virtual learning environment. Gamification. Algorithm. Teaching and Learning. Software Engineering.

LISTA DE ILUSTRAÇÕES

Figura 1 - Framework Ocatalysis	19
Figura 2 - Lados esquerdo e direito do cérebro	24
Figura 3 - Gamificação <i>White-Hat</i> e gamificação <i>Black-Hat</i>	25
Figura 4 - Algoritmo simplificado de um dia de um estudante	36
Figura 5 - Grupo de operadores simples.	37
Figura 6 - Bloco de código em <i>Scratch</i>	40
Figura 7 - Ambiente de programação do <i>Scratch</i>	41
Figura 8 - Interface de atividade do Code Studio.	43
Figura 9 - Atividade de HTML Básico no Codecademy.	44
Figura 10 - Tela de atividade da <i>Khan Academy</i>	46
Figura 11 - Página inicial do Moodle.	47
Figura 12 - Tela de Login Funifier.	48
Figura 23 - Distribuição das técnicas.	51
Figura 24 - Nível 2 Ocatalysis.	52
Figura 13 - Modelo visual de integração.	60
Figura 14 - Criando uma Gamificação Funifier.	63
Figura 15 - Menu Funifier.	64
Figura 16 - Criação de Ações Funifier.	65
Figura 17 - Criação de níveis Funifier.	66
Figura 18 - Criação e Medalhas Funifier.	67
Figura 19 - Menu Integration do Funifier.	68
Figura 20 - Instalador de Plug-ins Moodle.	68
Figura 21 - Plugin instalado no Browser.	71
Figura 22 - Plugin Inline do Funifier.	72
Figura 25- Fluxo Básico do ambiente.	73
Figura 26 - Tela de login.	74
Figura 27 - Curso da gamificação.	75
Figura 28 - Primeira vista do curso.	75
Figura 29 - Lista de atividades	76
Figura 30 - Atividade "Conhecendo as medalhas".	76
Figura 31 - Atividade "[VÍDEO] Introdução a algoritmos".	77
Figura 32 - Desafio "Teste do Iniciante".	78
Figura 33 - Atividade Scratch.	78
Figura 34 - Ambiente de criação de algoritmos no Scratch	79
Figura 35 - Escolha de caminhos.	80
Figura 36 - Widget de Classificações.	80
Figura 37 - Seção de atividades condicionais.	81
Figura 38 - Seção EXTRA motivacional.	82

SUMÁRIO

1. INTRODUÇÃO	14
1.1 JUSTIFICATIVA	15
1.1.1 Questão de pesquisa	15
1.2 OBJETIVOS	16
1.2.1 Objetivo geral	16
1.2.2 Objetivos específicos	16
1.3 METODOLOGIA	17
1.3.1 Seleção das ferramentas	17
1.3.2 Adaptação de ferramentas	17
1.3.3 Verificação dos resultados	18
1.3.4 Organização do trabalho	18
2. REFERENCIAL TEÓRICO	19
2.1 GAMIFICAÇÃO	19
2.1.1 Framework Octalysis	19
2.1.3 Gamificação dos lados cerebrais esquerdo e direito	24
2.1.3 Gamificação <i>Black-Hat</i> e <i>White-Hat</i>	25
2.2 ASPECTOS PEDAGÓGICOS DA FERRAMENTA	25
2.2.1 A sala de aula atual	25
2.2.2 O conteúdo das aulas e a aprendizagem	26
2.2.3 Avaliação do modelo de ensino e aprendizagem	27
2.2.4 Ambiente virtual de ensino e aprendizagem	29
2.3 ENGENHARIA DE SOFTWARE	30
2.3.1 Processos de Software	30
2.3.2 Princípios fundamentais	31
2.3.3 Engenharia de Requisitos	31
2.3.3 Conceitos de Design	32
2.3.4 Conceitos de Qualidade	32
3. APRENDIZAGEM DE ALGORITMOS E FERRAMENTAS TECNOLÓGICAS	33
3.1 APRENDIZADO DE ALGORITMOS	33
3.1.1 Algoritmos	33
3.1.2 Ensino de Algoritmos	36
3.2 FERRAMENTAL TECNOLÓGICO	38

3.2.1 Scratch.....	38
3.2.2 Code Studio	40
3.2.3 Codecademy	41
3.2.4 Khan Academy.....	42
3.2.5 Moodle	44
3.2.6 Funifer.....	45
3.3 CONSIDERAÇÕES FINAIS SOBRE O FERRAMENTAL TECNOLÓGICO	46
4. AMBIENTE DE APRENDIZADO GAMIFICADO	47
4.1 PROJETO DE GAMIFICAÇÃO.....	47
4.1.1 Descoberta.....	49
4.1.2 Entrada	50
4.1.3 Dia-a-dia	51
4.1.4 Fim de jogo	53
4.1.5 Técnicas utilizadas na gamificação implementadas.....	55
4.2 DINÂMICA DAS FERRAMENTAS.....	56
4.2.1 Funcionamento do Moodle.....	57
4.2.2 Funcionamento do Scratch	57
4.2.3 Funcionamento do Funifier	58
4.3 CONFIGURAÇÃO DE AMBIENTE	58
4.3.1 Instalação Moodle gamificado.....	58
4.3.2 Configuração Funifier.....	59
4.3.3 Integração entre Funifier e Moodle	63
4.3.4 Integração entre Funifier e Scratch	66
4.3.5 Integração entre Moodle e Scratch	68
5. FLUXO BÁSICO DO AMBIENTE DE APRENDIZADO GAMIFICADO	68
5.1 FLUXOGRAMA DO CAMINHO EFETUADO.....	69
5.2 DETALHAMENTO DAS PRINCIPAIS ATIVIDADES DO FLUXO	70
5.2.1 Login	70
5.2.2 Inscrição no curso.....	71
5.2.3 Primeiras atividades.....	72
5.2.4 Desafios	73
5.2.5 Atividades do Scratch	74
5.2.6 Escolha de classe	75
5.2.7 Fim do curso	77

6. CONCLUSÃO	79
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	80
APENDICE A – Apanhado sobre Síndrome de Down	82
A.1 Aspectos biológicos	82
A.2 Aspectos do aprendizado	82
A.3 Aspectos educacionais	84
APENDICE B – Instalação do Moodle.....	87
APÊNDICE C – Lista de técnicas Octalysis	89

1. INTRODUÇÃO

A educação à distância vem ganhando força na sociedade a medida que a população se familiariza com as novas tecnologias, colocando assim mais confiança nos métodos virtuais de ensino. Com a crescente dos ambientes virtuais de ensino os mesmo vêm evoluindo em qualidade e quantidade. Novas instituições e até mesmo indivíduos vêm desprendendo seus esforços nesta nova forma de ensino e aprendizagem. Um exemplo claro é Salman Khan(2013), que por meio de vídeo aulas e, posteriormente com um ambiente completo de ensino, deu um importante passo à nova forma de ensino e aprendizado virtual.

Algoritmos computacionais são elementos centrais na formação dos profissionais de Tecnologia da Informação, e por consequência das novas tecnologias que vêm surgindo. Desenvolver raciocínio lógico através do aprendizado de algoritmos pode ser proveitoso para a população em geral, mesmo levando em consideração que o assunto seja aplicado majoritariamente em soluções de Tecnologia da informação.

Um dos grandes desafios do ensino é manter a motivação dos estudantes, mas como manter o aluno interessado em um conteúdo? Yukai Chou (2014) traz algumas respostas, mesmo que o foco de Chou não seja motivar especificamente os estudantes, com seu Framework chamado Octalysis, Chou apresenta técnicas que têm como objetivo motivar pessoas em suas atividades cotidianas, trazendo aspectos psicológicos observados em sua grande maioria em jogos para dentro de outras atividades.

Aliando aspectos de ensino propostos por Salman Khan à ferramenta motivacional desenvolvida por Yukai Chou, este trabalho traz a adaptação de um ambiente virtual de ensino com elementos de gamificação. Apresentando o conteúdo de algoritmos computacionais em um ambiente virtual gamificado, almejando assim auxiliar o aprendizado dos estudantes tendo como diferencial a motivação na confecção das atividades e postura ativa nos exercícios dentro do ambiente.

1.1 JUSTIFICATIVA

Adaptar uma ferramenta computacional com o intuito de ensinar algoritmos computacionais, utilizando técnicas de gamificação propostas por Yukai Chou por meio de um processo definido pode abrir precedentes para novas experiências. Como uma ferramenta genérica a mesma pode evoluir para contextos mais específicos e beneficiar usuário com necessidades mais específicas. Partido de uma experiência bem-sucedida outras podem surgir e, aumentar assim, o benefício de se utilizar gamificação e ambientes de aprendizado virtual, integrando duas técnicas relativamente novas no contexto de ensino e aprendizagem.

1.1.1 Questão de pesquisa

É possível adaptar um ambiente virtual de aprendizado gamificado, utilizando um processo pautado nas práticas de engenharia de software?

1.2 OBJETIVOS

A presente seção descreve os objetivos do trabalho, tanto no âmbito geral como no específico.

1.2.1 Objetivo geral

Adaptar um ambiente de aprendizado virtual para o ensino de algoritmos computacionais, com o auxílio de técnicas de gamificação.

1.2.2 Objetivos específicos

- Selecionar Ferramental tecnológico.
- Confeccionar o projeto de gamificação;
- Construir o curso de algoritmos;
- Testar o funcionamento do ambiente;

1.3 METODOLOGIA

O desenvolvimento do trabalho se deu em iterações semanais ou quinzenais, de acordo com o planejamento, como prevê o Processo Unificado, com ressalvas as características do projeto, o qual trata de um processo de adaptação de software e não de desenvolvimento. O ambiente foi adaptado e integrado em um servidor e seus elementos foram testados a fim de verificar o funcionamento dos elementos presentes no sistema.

1.3.1 Seleção das ferramentas

As ferramentas foram selecionadas com base em critérios técnicos específicos para cada uma das três ferramentas que compõem o ambiente.

O Moodle foi escolhido por ser uma ferramenta de criação de ambientes virtuais Open Source bem consolidada. Seus níveis de aceitação e confiabilidade por parte do público são altos e sua comunidade é bem estabelecida. Levando ainda em conta a escolha do Moodle, foram avaliadas suas funcionalidades e sua relevância como ambiente de aprendizado.

O Funifier foi escolhido para ser a ferramenta gamificadora do ambiente virtual, a ferramenta foi escolhida por estar alinhada com a proposta de gamificação de Yukai Chou e por oferecer suporte à integração com o Moodle, além de possuir um bom nível de maturidade no que tange a gamificação em si e a facilidade de retirada de métricas relacionadas ao comportamento dos jogadores.

O Scratch foi escolhido por ser uma forma paralela de se aprender algoritmos computacionais na prática. Abordando a construção de algoritmos de uma forma mais lúdica o Scratch integrado ao ambiente de aprendizado visa ampliar a motivação dos estudantes enquanto estão aprendendo sobre o conteúdo exposto. Colocando em contra partida outros ambientes que atuam da mesma maneira que o Scratch o mesmo se sobressai para a escolha deste ambiente virtual pois é uma ferramenta com maior gama de ações e também mais pura, no sentido de que não envolve outros contextos na criação de algoritmos a não ser o próprio uso de blocos lógicos.

1.3.2 Adaptação de ferramentas

O processo de instalação do Moodle se deu com base na documentação disponível no site moodle.org, site da organização desenvolvedora e mantenedora do

projeto, os detalhes desta adaptação se encontram na seção 4. O Funifier recebeu os dados do projeto de gamificação, dados esses retirados da implementação do framework Octalysis (Chou, 2014) no contexto do projeto, detalhes desta implementação se encontram no capítulo 4. O Scratch não foi adaptado em sua estrutura principal por restrições técnicas, o método utilizado para integra-lo ao ambiente virtual gamificado mapeia apenas os elementos HTML das páginas Scratch, os detalhes técnicos dessa integração também estão presentes no capítulo 04.

1.3.3 Verificação dos resultados

Os resultados de comportamento esperados para o ambiente foram definidos na etapa de projeto de gamificação seção 4.1. Resultados esses que foram definidos com base na literatura, mais especificamente utilizando o autor Yukai Chou (2014). Os testes consistiam na verificação do comportamento esperado em face da ação executada dentro do sistema, o teste era tido como satisfatório ao se observar nos testes os resultados esperados.

1.3.4 Organização do trabalho

O trabalho foi organizado em 6 capítulos, sendo eles: Introdução, Referencial teórico, Aprendizagem de algoritmo, Ambiente de aprendizado gamificado, Fluxo básico do ambiente e conclusão.

A introdução apresenta um panorama e os objetivos do trabalho ao leitor. O referencial teórico trata do levantamento de estudos que compõem os principais assuntos tratados no trabalho, sendo eles: Gamificação, Aspectos pedagógicos da ferramenta e Engenharia de Software. O capítulo 3 trata da definição dos conceitos de algoritmos e como eles são ensinados a pessoas sem familiaridade com o assunto, trata também de uma revisão do ferramental tecnológico presente no contexto de ensino e aprendizagem online. O capítulo 4 apresenta as ações desenvolvidas para a adaptação do ambiente de aprendizado virtual, desde a instalação e configuração individual dos softwares até a integração dos mesmos. O capítulo 5 trata do fluxo básico da aplicação, no qual algumas das atividades presentes dentro do ambiente estão descritas.

2. REFERENCIAL TEÓRICO

Esta seção engloba todo o referencial teórico que serviu como base para a elaboração do trabalho de conclusão de curso, as seções discorrem sobre Gamificação, aspectos pedagógicos e Engenharia e Software.

2.1 GAMIFICAÇÃO

Segundo Yu-kai Chou (2014, p.8), Gamificação é o ofício de derivar a diversão e os elementos achados tipicamente nos jogos e aplica-los nas atividades produtivas do mundo real de forma cuidadosa. Yu-kai Chou chama a Gamificação de *Design* focado em humanos, o que faz oposição ao que é normalmente executado na sociedade, um sistema focado em funções, que é pensando para se executar tarefas de maneira rápida, o trabalhador faz esse trabalho porque ele precisa e não porque ele quer. A razão para se chamar essa disciplina de gamificação é porque a indústria de jogos foi a primeira a aperfeiçoar o *Design* focado em humanos.

Ainda segundo Yu-kai Chou (2014, p. 21), quando se está desenhando a estrutura de Gamificação de um projeto, não se deve pensar em o que os outros projetos têm que os fazem bem-sucedidos, em contrapartida o que fará um projeto de Gamificação bem-sucedido é o fato de se focar, primeiramente, na motivação de seus usuários. Ao invés de começar um projeto selecionando as ferramentas, mecânicas e elementos que se irá usar, o que deve ser feito é pensar no que o seu usuário deve sentir; espera que eles se sintam orgulhosos ou até mesmo inspirados? A partir desse momento deve-se pensar em quais elementos deverão ser inseridos no projeto para que ele alcance os resultados esperados, sempre lembrando que os elementos dos jogos são um meio de se alcançar os objetivos e não os objetivos em si.

2.1.1 Framework Octalysis

O Octalysis é um framework de Gamificação desenvolvido por Yu-kai Chou. O Octalysis serve como base para a aplicação das técnicas de Gamificação e descreve a forma como se deve iniciar, executar e continuar este processo de aplicação da disciplina de maneira a encontrar resultados satisfatórios nos campos de motivação do usuário ao se completar uma ação dentro do contexto do projeto.

Chou, através de vários anos de experimentos e ajustes chegou à conclusão de que tudo que fazemos é baseado em um ou mais núcleos centrais, dentre os 8 existentes, pertencentes ao Octalysis. Se nenhum dos 8 tipos de núcleos centrais está

por trás de uma ação desejada, então não há motivação em se completar aquela ação. A seguir os oito núcleos centrais do Octalysis serão apresentados.

Nem todas as técnicas foram empregadas no contexto do projeto. A figura (1) mostra o octógono que representa os núcleos centrais do framework.

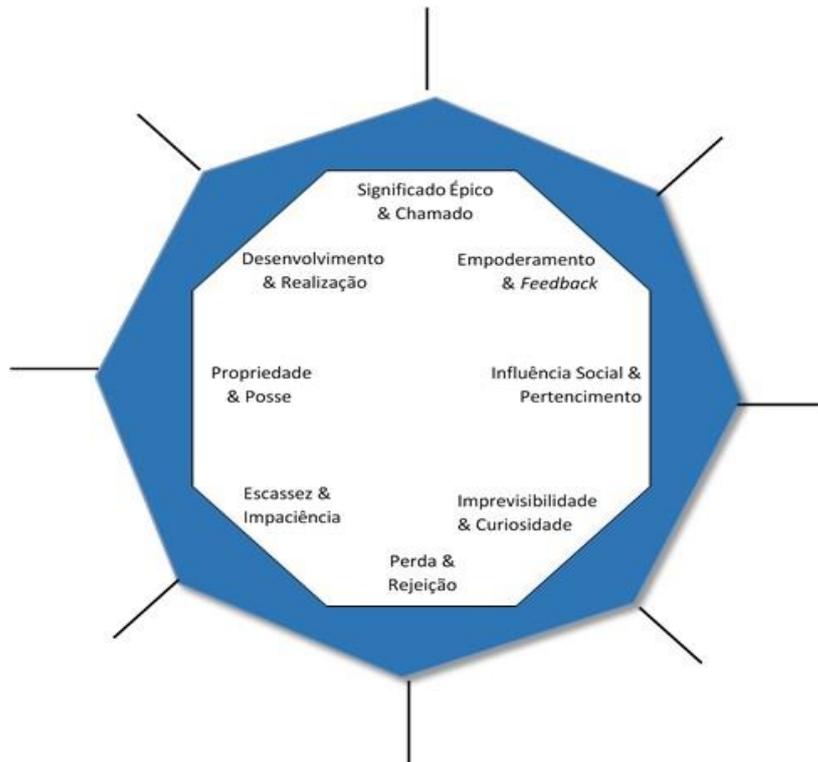


Figura 1. Framework Octalysis

2.1.1.1 Significado épico e chamado

Significado épico e chamado é o núcleo central que está em jogo quando uma pessoa acredita que está realizando algo superior a eles mesmos e/ou são escolhidos para executar aquela ação. Esse núcleo também está na noção que as pessoas têm da “Sorte de iniciante”, que faz com que acreditem que eles possuem uma espécie de prêmio que os outros não possuem (Chou, 2014). A seguir algumas técnicas relacionadas ao Significado épico e chamado serão listadas:

- Narrativa;
- Sorte de iniciante;
- Criança predestinada;
- Almoço grátis;
- Elitismo;

- Significado superior;
- Herói da humanidade.

2.1.1.2 Desenvolvimento e realização

Desenvolvimento e realização é o núcleo responsável por conduzir progressos, desenvolver habilidades, alcançar maestrias e eventualmente superar desafios. O ponto chave deste núcleo é a palavra “desafio”, sem desafio não há sentido em se distribuir medalhas, troféus (CHOU, 2014). A seguir algumas técnicas relacionadas ao Desenvolvimento e realização serão listadas:

- Pontos;
- Medalhas (Símbolos de conquista/realização);
- Barra de progresso;
- Ganhar prêmios;
- Tutorial passo-a-passo;
- Lutas contra chefes;
- Lista de desafios.

2.1.1.3 Empoderamento da criatividade e *Feedback*

Empoderamento da criatividade e *feedback* é expressado quando o usuário está engajado em um processo criativo em que ele repetidamente descobre novas coisas e combinações. As pessoas não precisam apenas de um modo para expressar sua criatividade, elas também precisam receber um *feedback* de como ela está caminhando. Isso explica o porquê de se fazer arte ou brincar com lego é tão divertido (CHOU, 2014). A seguir algumas técnicas relacionadas ao Empoderamento da criatividade e *Feedback* serão listadas:

- Desbloqueio de marcos;
- Controle em tempo real;
- Combinações em cadeia;
- *Feedback* instantâneo;
- *Boosters*;
- Autonomia voluntária;
- Percepção de escolhas.

2.1.1.4 Propriedade e Posse

Propriedade e posse é quando o usuário se sente motivado por ter ou controlar algo. Quando uma pessoa sente ser a proprietária de algo ela naturalmente começa

a desejar o crescimento e melhoramento do que ela possui. Esse núcleo também está presente no fato de uma pessoa sentir que é proprietário de um processo, projeto ou organização (CHOU, 2014). A seguir técnicas sobre Propriedade e Posse serão listadas:

- Bens virtuais;
- Construir a partir do zero;
- Coleções;
- *Avatares*;
- Curva de aprendizado;
- Proteção;
- Monitoramento.

2.1.1.5 Influência social e Pertencimento

Influência social e pertencimento incorporam todos os elementos sociais que motivam as pessoas, incluindo: tutoria, aceitação social, *feedback* social, companheirismo e até mesmo competição e inveja. A exemplo de quando você vê um amigo alcançando um objetivo, você se torna mais focado em alcançar o mesmo *status* (CHOU, 2014). A seguir técnicas sobre Influência social e Pertencimento serão listadas:

- Amizades;
- Reconhecimento social;
- Atividades em grupo;
- Economia do “Obrigado”;
- Orientações.

2.1.1.6 Escassez e Impaciência

Escassez e impaciência é o núcleo direcionado que explica o porquê de uma pessoa desejar alguma coisa apenas porque essa coisa é extremamente rara, exclusiva ou não está disponível no momento. Muitos jogos exploram isso, com “pausas de tortura”, na qual você deve voltar após um período de tempo para conseguir alguma coisa. O fato de uma pessoa não poder conseguir a coisa no exato momento faz com que ela passe o intervalo de tempo pensando sobre o seu objeto de desejo (CHOU, 2014). A seguir técnicas sobre Escassez e Impaciência serão listadas:

- Dinâmica de nomeação;
- Intervalos fixos;

- *Feedback* paciente;
- Fossos;
- Aceleradores;
- Contagem regressiva.

2.1.1.7 Imprevisibilidade e Curiosidade

Imprevisibilidade e curiosidade é o núcleo relacionado ao envolvimento do usuário em uma atividade justamente por ele não saber o que irá acontecer no momento futuro. Quando seu cérebro não consegue reconhecer padrões e prever o que está por vir, ele usa toda a prioridade e atenção ao inesperado. Este é o principal núcleo por trás do vício por apostas e também uma das explicações para pessoas assistirem novelas e lerem livros (CHOU, 2014). A seguir técnicas relacionadas à Imprevisibilidade e Curiosidade serão listadas:

- Escolha brilhante;
- Mini desafios;
- *Easter eggs*;
- Recompensas aleatórias;
- Travessuras;
- Maravilha óbvia;
- Recompensas repentinas.

2.1.1.8 Perda e Rejeição

Perda e Rejeição vem de um pressuposto óbvio, a necessidade ou capacidade de evitar algo ruim ou a perda de algo bom. Em uma escala menor, pode ser caracterizado por evitar a perda de um trabalho anterior ou mudar o comportamento de alguém. Alguns acham que caso eles não aproveitem a oportunidade agora perderão essa chance para sempre, as promoções por tempo limitado são um bom exemplo disso (CHOU, 2014). A seguir técnicas relacionadas à Perda e Rejeição serão listadas:

- Perda de progresso;
- Preguiça de Status Quo;
- Carta escarlate;
- Sepulturas visuais.

2.1.3 Gamificação dos lados cerebrais esquerdo e direito

Yu-kai Chou explica ainda que além dos 8 núcleos presentes no Octalysis há também outras naturezas que permeiam esse *framework*. Algumas fazem o usuário se sentir poderoso, porém não criam urgência, outras o fazem sentir urgência e obsessão, e até mesmo dependência, mas não o fazem se sentir bem. Alguns são focados em fatores extrínsecos de longo prazo ou curto prazo. A organização do octalysis está de tal maneira que os núcleos focados em criatividade, dinâmicas sociais e auto expressão estejam localizadas no quadrante direito, enquanto os núcleos focados em lógica, pensamento analítico e propriedades estão localizados no lado esquerdo. Vale ressaltar que o autor não faz nenhuma afirmação biológica sobre os lados cerebrais e que a nomenclatura é simbólica e representa diferentes funções do cérebro. A figura (2) demonstra essa divisão no framework.



Figura 2. Lados esquerdo e direito do cérebro

Os núcleos focados no quadrante esquerdo, presentes no *framework* Octalysis, representam fatores extrínsecos, algo que você quer obter, seja ele um bem ou um objetivo, já os núcleos focados no quadrado direito estão relacionados a fatores intrínsecos, a atividade em si é recompensadora. A diferenciação é importante, muitas empresas focam em fatores extrínsecos, como premiações por tarefas

cumpridas, porém muitos estudos têm mostrado que motivações intrínsecas rivalizam com as motivações extrínsecas (Chou, 2014).

2.1.3 Gamificação *Black-Hat* e *White-Hat*

Yu-kai Chou organizou o Octalysis de maneira a separar as motivações mais positivas acima do octógono e as motivações mais negativas abaixo. As motivações positivas são chamadas de Gamificação *White-Hat*, enquanto as negativas são chamadas de Gamificação *Black-Hat*. Empresas de jogos podem fazer sucesso com jogos baseados em Gamificação *Black-Hat*, no qual os usuários vão se sentir viciados e retidos, porém quando eles encontrarem a possibilidade de se livrarem do jogo, eles irão, isso se dá pela razão de que os usuários não se sentem no controle da situação enquanto jogam. Vale a ressalva de que não é porque as técnicas de Gamificação são chamadas de *Black-Hat* que elas sejam negativas, há sim a possibilidade de executá-las de maneira proveitosa, produtiva e que retorne resultados saudáveis (Chou, 2014). A figura (3) demonstra essa visão no *framework*.

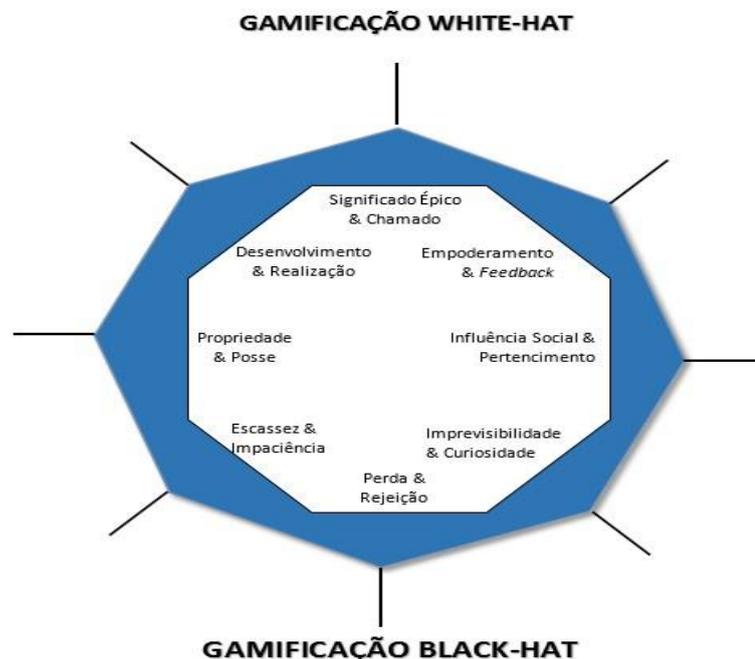


Figura 3. Gamificação *White-Hat* e gamificação *Black-Hat*.

2.2 ASPECTOS PEDAGÓGICOS DA FERRAMENTA

2.2.1 A sala de aula atual

Khan (2013) explica que as salas de aula atuais são caracterizadas por professores em posições privilegiadas, em frente à uma quantidade considerável de estudantes que assistem passivamente o desempenho do professor. Os professores, por um meio expositivo, apresentam aos estudantes os conceitos que os mesmos devem aprender a fim de obterem a certificação de que merecem avançar para o próximo nível de ensino e por fim conseguir um diploma, da mesma forma, os alunos encaram o professor como um ser que está lá para transmitir seus ensinamentos e que esses devem a todo custo serem memorizados para aprovação nos testes de desempenho. O modelo atual de sala de aula foi baseado em um modelo implantado pela primeira vez na Prússia, no século XVIII (KHAN, 2013).

A educação pública e compulsória financiada por impostos foi vista como uma ferramenta política, ao menos tanto quanto pedagógica, e não se tentava disfarçar isso. A intenção não era produzir pensadores independentes, mas extrair cidadãos leais e complacentes, que aprenderiam o valor de se submeter à autoridade dos pais, dos professores, da Igreja e, em última instância, do rei (KHAN, 2013, p.81).

2.2.2 O conteúdo das aulas e a aprendizagem

Segundo Khan (2013) o conteúdo é geralmente dividido em pequenas partes, que são transmitidas em intervalos fixos de tempo, no final de cada etapa há uma prova para verificação dos conceitos ensinados. Avaliando a situação em que a educação escolar se encontra, falhando em aspectos como motivação dos estudantes, fixação de conteúdo, igualdade de possibilidades e defasagem na base de conhecimento, pode-se dizer que já está em tempo de se repensar os métodos de ensino e aprendizagem de forma a trazer maiores benefícios tanto para os estudantes, quanto para os professores. “O aluno com bom histórico que de repente deixa de compreender um tópico mais complexo por causa do alicerce queijo suíço sente-se como se desse com a cara na parede, e isso é muito comum” (Khan, 2013, p.90).

Desde de que se adentra dentro de uma sala de aula, o estudante é condicionado a escutar e absorver passivamente o conteúdo dentro de sala de aula até o momento em que se depara com uma prova, o estudante não é incentivado a produzir conhecimento, mas absorvê-lo a fim de cumprir suas tarefas.

Uma das grandes dificuldades dos estudantes frutos desse modelo de ensino é relacionar o que estudou em sala de aula. “A falha em relacionar tópicos do currículo

escolar com sua eventual aplicação no mundo real é uma das deficiências centrais do nosso fragmentado modelo de sala de aula” (Khan, 2013, p.93).

2.2.2.1 Como a educação acontece

Segundo Kandel (2007), citado por Khan (2013, p.51) “a aprendizagem é, na verdade, nada mais nada menos que uma série de alterações que ocorrem nas células nervosas que compõem nosso cérebro“. Khan (2013) explica que o cérebro se altera de forma a registrar o conhecimento, e que por mais que esse conhecimento não seja permanente o mesmo não é perdido completamente, explicando o fato de ser mais fácil aprender um conceito pela segunda vez. Segundo Kandel (2007), citado por Khan (2013): “para que uma memória persista, a informação precisa ser processada de maneira profunda e meticulosa. Isso se consegue ao prestar atenção à informação e, em seguida, a associá-la significativa e sistematicamente a algum conhecimento”.

Nós aprendemos, antes de tudo, decidindo aprender, assumindo um compromisso com a aprendizagem, que, por sua vez, gera concentração. A concentração não se refere unicamente à tarefa imediata a cumprir, mas a todas as inúmeras associações que a cercam. Todos os processos são ativos e profundamente pessoais. (Khan, 2013, p.50)

Khan (2013), propõe que haja de forma majoritária, conexões entre as disciplinas aprendidas e condena o sistema atual de ensino que fragmenta e isola pedaços específicos de assuntos, para facilitar que os mesmos se encaixem no cronograma corrido das salas de aula. Khan (2013, p.55) afirma: “São as conexões entre conceitos, ou falta de conexões, que separam os estudantes que decoram uma fórmula para a prova, só para esquecê-la, daquelas que internalizam os conceitos”.

2.2.3 Avaliação do modelo de ensino e aprendizagem

Segundo Khan (2013), não é difícil avaliar o modelo atual de ensino e aprendizagem, essa não é uma tarefa tão árdua, as dificuldades estão claras para qualquer um dos lados, o desafio, na verdade, seria superar essas barreiras. A educação como conhecemos hoje já é datada de alguns séculos atrás e pode estar ultrapassada, a mesma surgiu do modelo Prussiano de aprendizagem, que tinha como objetivo transmitir conceitos básicos para uma grande escala de pessoas, de maneira padronizada (KHAN, 2013), porém esse modelo de ensino pode não ser o mais adequado.

Hoje, com avanços tecnológicos e mudanças culturais existem novas possibilidades que podem ser exploradas, novas metodologias pensadas e desenvolvidas. Khan (2013) propõe uma quebra profunda de paradigmas quando o assunto é educação, não apenas especulando e teorizando, mas colocando em práticas conceitos que vêm se provando efetivos no que tange o aprendizado de diversas crianças ao redor do mundo. Salman Khan, o fundador da *Khan Academy*, vai na contramão das escolas tradicionais e propõe o auto aprendizado, proatividade e maior responsabilidade dos alunos com seu ensino, sua ideia começou com aulas particulares, evoluiu para vídeos didáticos e hoje conta com uma plataforma completa de recursos virtuais para aprendizagem de diversos assuntos (KHAN, 2013).

Khan (2013) acredita que o modelo de aprendizado baseado no tempo, com pequenas parcelas de conteúdo separadas e repassadas e com seu principal aspecto centralizado em avaliações está completamente ultrapassado, e que não atende mais as demandas do mundo moderno em que a sociedade globalizada está inserida. Um mundo dinâmico e que evolui a largos passos, com inovações e quebras de paradigmas em todas as áreas do conhecimento.

2.2.3.1 Provas e avaliações

Segundo Khan (2013, p. 94) “os testes pouco ou nada dizem sobre o potencial do aluno em aprender a matéria. Na melhor das hipóteses, fornecem uma fotografia instantânea de onde ele está num determinado momento no tempo”. Khan (2013) critica ainda o fato de uma pontuação na prova não refletir se o estudante entende ou não um conteúdo com profundidade ou se o conteúdo está de fato na memória de longo prazo.

Muitos estudantes tem a capacidade de decorarem o conteúdo apenas pelo tempo necessário para tirar uma boa nota na avaliação. As notas ruins podem ainda refletir momentos de descuido ou falta de sorte, por abordar exatamente a parte do conteúdo que está defasado, por parte dos estudantes. Khan (2013) afirma que os testes e provas podem ser ferramentas valiosas e que provas bem planejadas podem ser usadas como evidências de que alguém efetivamente conhece bem um tópico de uma matéria.

Khan (2013) propõe que para avaliar o aprendizado do estudante não basta apenas uma nota que compreenda 70% do total avaliado. O autor diz ainda que

mesmo que a nota de corte fosse 90% do total avaliado o resultado obtido não seria o ideal e que esses 10% restantes se acumulariam e prejudicariam o estudante em algum momento do futuro. A proposta para a avaliação direta do aprendizado seria uma série de dez questões que abordassem o conteúdo de forma completa, credenciando o aluno apenas se o mesmo fosse capaz de acertar todas as dez questões. As questões estariam em uma base de dados maior e que permitisse uma aleatoriedade das questões sem uma repetição que permitisse a decoraç o por parte do aluno.

2.2.4 Ambiente virtual de ensino e aprendizagem

Na plataforma de aprendizagem de Salman Khan, a *Khan Academy*, os estudantes interessados t m acessos a v deo aulas de no m ximo dez minutos. Khan explica que chegou ao tempo de dez minutos por v deo aula ao acaso e por restri es imposta pelo site YouTube que limitava, na  poca, publica es de v deos   essa faixa de tempo. Khan complementa que te ricos da educa o apontam que o per odo de dura o da aten o dos alunos est  em torno de dez a dezoito minutos. Nas aulas expostas por Salman, os alunos aprendem conceitos b sicos de todas as naturezas, evoluindo para conceitos mais complexos   medida que o estudante se sente preparado e n o quando o tempo de aula acaba e uma prova   aplicada.

Khan (2013) prop e que o foco do ensino n o deve estar no tempo e sim no aprendizado. O aluno deve decidir quando est  de fato conhecendo   fundo o assunto tratado, ou se o mesmo se encontra em um impasse. Caso esteja encontrando dificuldades para entender um certo conceito, ele mesmo pode voltar e rever o v deo quantas vezes quiser, podendo ainda visitar v deos antigos e revisar conceitos mais b sicos que comp em o contexto que ele encontra dificuldade. Nesse contexto proposto por Khan os professores n o est o mais como “entregadores” de conhecimento, mas sim como consultores que auxiliariam os alunos em suas d vidas e impasses, o estudante agora   o respons vel por buscar o conhecimento, constr i-lo com o aux lio do professor. A figura do professor n o   mais a de um ser distante, que recita suas aulas, mas um orientador, um ator mais pr ximo no processo de aprendizagem.

Uma das maiores dificuldades dos professores ao personalizar a instru o   administrar uma grande quantidade de alunos que est o trabalhando em diferentes

matérias, em diferentes anos escolares e em diferentes ritmos. O *software* da Khan Academy é capaz de construir um mapa mental e oferecer conteúdos relacionados ao assunto recém finalizado.

E como o *software* identifica quando um estudante está apto a avançar? A proposta de Khan (2013) de que a avaliação do aprendizado de um assunto por parte do estudante pode ser adquirida com uma sequência correta de dez questões referentes a esse mesmo assunto é aplicada. Segundo Khan (2013, p. 138) “isso pode parecer radical e superidealizado, ou simplesmente difícil demais, mas eu argumentava que era o único padrão simples e digno de respeito tanto da matéria quanto do estudante”. Khan afirma ainda que dez questões foi um número arbitrário, mas insistiu que um número fixo era importante, pois dava aos estudantes um objetivo, que quando alcançado os davam confiança e autoestima para continuarem para desafios maiores de conteúdos mais difíceis.

2.3 ENGENHARIA DE SOFTWARE

Segundo a IEEE a engenharia de software é aplicação de uma abordagem sistemática, disciplinada e quantificada do desenvolvimento, operação, e manutenção de *software*. O estudo das abordagens como são. Fred Brooks (apud Pressman 2010, p. 15) “Einstein argumentou que deve haver uma explicação simplificada para a natureza, porque Deus não é caprichoso ou arbitrário”.

2.3.1 Processos de Software

Segundo Pressman (2010), um processo é um conjunto de atividades, e tarefas que são executadas quando um produto de trabalho está para ser criado. No contexto de engenharia de *software* o processo é uma abordagem adaptável que permite que pessoas façam seu trabalho. A intenção é sempre de entregar *software*.

Pressman (2010) explica que um framework estabelece as fundações básicas para a definição de um processo completo de engenharia de *software*, identificada por um conjunto pequeno de atividades de framework. Um processo genérico é encabeçado por cinco atividades:

- Comunicação: Direcionada para o entendimento das necessidades dos *stakeholders* e para requisitos que ajudem a definir as *features* do *software* e suas funcionalidades.

- Planejamento: Direcionada à confecção do Plano de projeto de *Software*, que descreve as atividades técnicas que serão executadas, os riscos que serão enfrentados, os recursos que serão necessários, o produto que será entregue e o cronograma de trabalho.
- Modelagem: Direcionada ao maior entendimento por parte do engenheiro de *software* por meio da criação de um modelo.
- Construção: Combina a geração de código (Manual ou automática) e os testes que são necessários para descobrir os erros do *software*.
- Implantação: Direcionada à entrega do *software* ao cliente, que avalia e provê *feedback* sobre a entrega.

2.3.2 Princípios fundamentais

Segundo Pressman (2010) a engenharia de *software* é guiada por um grupo de princípios fundamentais que ajudam na aplicação de um processo de *software* significativo e a execução de métodos efetivos de engenharia de *software*. Pressman afirma: " todo projeto e todo time é único. Isso significa que você deve adaptar seu processo para atender suas necessidades". De acordo com Pressman, os princípios gerais são:

- Prover valor para o usuário final;
- Manter a simplicidade;
- Manter a visão do produto e do projeto;
- Entender que terceiros consomem o que você produz;
- Estar aberto para o futuro;
- Planejar com antecedência o reuso

2.3.3 Engenharia de Requisitos

Pressman afirma que a engenharia de requisitos estabelece uma base sólida para o *Design* e construção. Sem ela, o *software* resultante tem uma grande probabilidade de não atender as necessidades do usuário.

A engenharia de requisitos oferece ferramentas adequadas para a extração das necessidades do usuário, para que o projeto possa trabalhar e entregar o produto

desejado. Informações são coletadas, interpretadas e refinadas ao longo do ciclo de vida. Existem modelos que guiam as atividades da engenharia de requisitos, existem ainda modelos para a escrita desses requisitos, como por exemplo, casos de uso e histórias de usuário. Mesmo que com abordagens diferentes, o objetivo é sempre o mesmo, entender e capturar as necessidades do usuário.

2.3.3 Conceitos de Design

O objetivo do *Design* é produzir um modelo ou representação que exiba firmeza, utilidade e deleite. Evoluindo de acordo com novos métodos, melhores análises e entendimentos mais amplos aparecerem. Segundo Pressman o *Design* é o que quase todo o engenheiro quer fazer, o local onde a criatividade comanda. Pressman afirma que “o *Design* de *software* deve começar com considerando os dados, o fundamento para todos os outros elementos de *Design*. Depois que a fundação é colocada, a arquitetura deve ser derivada. Só então você deve executar as outras tarefas de *Design*”.

2.3.4 Conceitos de Qualidade

Segundo Pressman (2010) o conceito de qualidade não pode ser facilmente definido, mesmo que você saiba o que é um *software* de qualidade quando o vê. Na engenharia de *software* a qualidade diminui os esforços de retrabalho, diminuindo assim os custos e o mais importante, mantendo o *software* no tempo de mercado. Para se alcançar a qualidade de *software* são necessários: processo e práticas de engenharia comprovados, gerência de projeto sólida, controle de qualidade compreensível e a presença de uma infraestrutura de garantia de qualidade.

2.4.4.1 Estratégia de testes

Pressman (2010) afirma que um *software* é testado para se descobrirem os erros que foram cometidos de forma despercebida, enquanto ele era desenvolvido. Para que os testes sejam executados de forma bem-sucedida é necessário que uma estratégia correta seja desenvolvida. As tarefas dessa estratégia devem ser planejadas de forma antecipada e executadas de forma sistemática.

Segundo Pressman (2010), as atividades de teste são elementos de um tópico maior, chamado de Validação e Verificação de *software*. As atividades de testes podem passar desde teste de unidade, testes de integração e testes de usabilidade,

todos eles com seu grau de importância em cada projeto, e por consequência, estrategicamente desenhados para cada situação.

3. APRENDIZAGEM DE ALGORITMOS E FERRAMENTAS TECNOLÓGICAS

Neste capítulo serão apresentados conceitos referentes à definição e aprendizagem de algoritmos, tal como o ferramental tecnológico utilizado. Na subseção 3.1 serão apresentados conceitos sobre algoritmos e na subseção 3.2 as ferramentas que foram estudadas e utilizadas na realização do trabalho. Os *softwares* aqui avaliados não cobrem em toda a totalidade o conjunto de softwares educacionais existentes, os mesmos foram selecionados pelo autor para que sirvam de referência.

3.1 APRENDIZADO DE ALGORITMOS

A seção apresenta os fundamentos necessários para uma introdução ao ensino e aprendizagem de algoritmos, descrevendo o que são algoritmos e seus aspectos básicos, subseção 3.1.1, e também como se dá o ensino de algoritmo em determinados contextos, subseção 3.1.2, tomando como base estudos realizados.

3.1.1 Algoritmos

Um algoritmo pode ser tratado como uma sequência lógica de passos bem definidos e não ambíguos para se cumprir uma tarefa em um tempo finito, como uma “receita de bolo”. Ferreira (FERREIRA, 2010) define algoritmo como: "Cálculos, e processos de cálculos, ou resolução de problemas semelhantes, em que se definem, de forma genérica e sem restrições, regras formais para a obtenção de uma solução de problema". No dia a dia, o ser humano, naturalmente executa algoritmos sem que perceba, seguindo uma ordem lógica de passos até cumprir suas tarefas cotidianas. Mesmo que parece ser algo simples, os algoritmos podem crescer em tamanho e complexidade. Dentro da computação o algoritmo tem um papel de destaque, servindo de insumo para programas implementados nas mais diversas linguagens de programação.

Informalmente, um algoritmo é 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 transformam a entrada na saída. (CORMEN et al. p.3)

Nesta seção não serão explorados todos os conceitos de algoritmos como, por exemplo, algoritmos aplicados em grafos, fluxo máximo, redes de ordenação e árvores. Este trabalho apresenta conceitos básicos como: variáveis, operadores, tipos de dados, laços de repetição e estruturas condicionais. A Figura 4 demonstra de forma simplificada um fluxograma de um algoritmo representando o dia de um estudante.

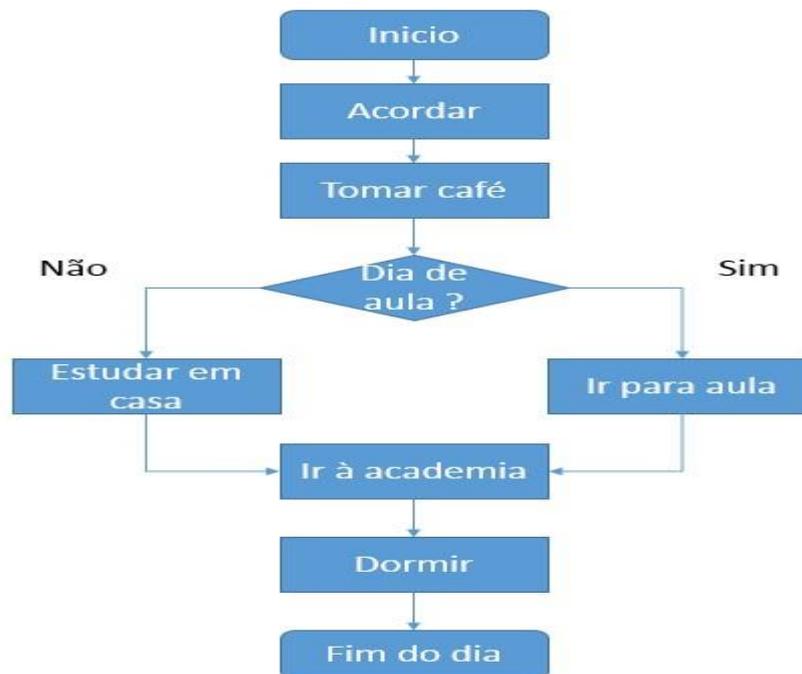


Figura 4. Algoritmo simplificado de um dia de um estudante

As seções a seguir descrevem os elementos considerados relevantes para a compreensão de algoritmos com um todo. Parte se da premissa de que se a pessoa for capaz de compreender estes conceitos, juntamente com a ideia inicial do que é um algoritmo, ela será capaz de confeccionar seu próprio algoritmo.

3.1.1.1 Tipos de Dados e Variáveis

Os algoritmos normalmente executam suas ações em cima de dados, que geralmente são fornecidos pelo usuário. Cada linguagem de programação possui seus próprios tipos de dados que por sua vez cumprem o propósito à que foram criados e seguem as convenções estipuladas por seus criadores. Damas (2007) cita os tipos de dados básicos presentes na linguagem C (char, int, float e double), os mesmos tratam de valores numéricos e alfanuméricos.

Segundo Damas (2007, p. 22) uma variável é: “nada mais nada menos que um nome que damos a uma posição de memória para conter um valor de um determinado tipo”. Damas de forma mais genérica, faz uma analogia à recipientes, que devem ser selecionados de maneira a comportar certos tipos de produtos. As variáveis podem ainda, trocar seus valores durante a execução de um programa, como o próprio nome sugere.

3.1.1.2 Operadores

Segundo Damas (2007) os operadores são elementos que aplicam ações/operações, sejam elas matemáticas, lógicas ou condicionais, em determinados trechos do algoritmo. Os operadores e sua carga lógica podem ser representados por símbolos, como visto na Figura 5. Os operadores podem verificar se uma variável é maior que outra (usando o símbolo “>”, na linguagem C, por exemplo), se são iguais, (utilizando o símbolo “==”, na linguagem C), podem também acrescentar valor condicional ao código (símbolo do operador AND/E “&&”, ou OR/OU “| |”).

OPERADORES
< <= > >=
== !=
&&
?:

Figura 5. Grupo de operadores simples.

3.1.1.3 Estruturas condicionais

Em quase todas as atividades cotidianas do ser humano há a possibilidade de se fazer escolhas, por onde andar, o que vestir, o que comer, dentre tantas outras possibilidades. Normalmente essas atividades são acompanhadas por condições, como por exemplo se o clima é de calor, então eu devo me vestir com uma blusa de tecido leve. Da mesma forma os algoritmos podem ter escolhas baseadas em condições impostas, geralmente esses caminhos são tomados com a ajuda de instruções de controle, exemplos claros são: SE, SENÃO, ENTÃO. Segundo Damas (2007) esse tipo de instrução permite indicar quais as circunstâncias em que se deve

executar determinada instrução ou conjunto de instruções. “SE” uma condição é cumprida “ENTÃO” certo passo é executado “SENÃO” outro caminho é tomado.

3.1.1.4 Estruturas de repetição

A medida em que o algoritmo vai crescendo, novas possibilidades vão surgindo. Além de seguir um conjunto linear de instruções e/ou seguir instruções que podem ou não ser executadas dependendo de condições pré-estabelecidas, um algoritmo pode executar ações em laços, que se repetem em um conjunto predefinido de vezes ou até que uma condição seja cumprida, o que caracteriza, segundo Damas (2010), uma estrutura de repetição.

A exemplo da estrutura *while*, que segundo Damas (2010) executa uma instrução ou bloco de instruções enquanto uma determinada condição for verdadeira. Há também uma estrutura que trabalha melhor em condições em que a quantidade de repetições é conhecida, o *for*.

3.1.2 Ensino de Algoritmos

Esta seção descreve o ensino e aprendizado de algoritmos e também do pensamento computacional de crianças, como apresentado por Wing (2006), ou até mesmo pessoas que não tinham contato prévio com lógica computacional.

De acordo com Wing (2006, p.33) o pensamento computacional envolve a solução de problemas e o entendimento do comportamento humano por meio de conceitos fundamentais da ciência da computação. O pensamento computacional inclui uma gama de ferramentas mentais e reflete a amplitude da ciência da computação. Wing afirma que ainda que o pensamento computacional é usar a abstração e decomposição quando em face à problemas complexos, é escolher a representação correta quando se aborda um problema. As características descritas são fundamentais para a criação e entendimento de algoritmos computacionais, sendo uma forma mais genérica de se chegar às atividades desenvolvidas por profissionais da ciência da computação, em se tratando de algoritmos.

Wing (2006) afirma que “pensar como um cientista da computação significa mais do que estar capaz de programar um computador. Requer um pensamento abstrato em múltiplos níveis de abstração”.

Kalelioglu (2015) descreve sete grandes ideias que acredita serem fundamentais para o pensamento computacional:

1. Computação é uma atividade de criatividade humana;
2. Abstração reduz a informação e o detalhamento para focar em conceitos relevantes para o entendimento e resolução de problemas;
3. Dados e informação ajudam a criar conhecimento;
4. Algoritmos são ferramentas para expressar e desenvolver soluções;
5. Programação é um processo criativo e produz artefatos computacionais;
6. Dispositivos digitais, sistemas, e redes interconectadas viabilizam e promovem soluções de problemas por meio de uma aproximação computacional;
7. Computação permite inovações em outros campos do conhecimento, como medicina, humanidades, ciências sociais, artes, engenharia e negócios.

Essas características são consideradas importantes para o desenvolvimento intelectual das crianças Wing (2006). Através desse desenvolvimento novas conexões podem ser feitas, melhorando assim a base do pensamento.

3.2 FERRAMENTAL TECNOLÓGICO

A seção apresenta o ferramental tecnológico utilizado como base para a realização do trabalho. As subseções 3.2.1 até 3.2.6 discorrem acerca de ferramentas relacionadas desde ao ensino e aprendizado virtual até a implementação de gamificação em ambientes virtuais.

3.2.1 Scratch

O Scratch ¹ é uma poderosa linguagem de programação (que vem acompanhada de um ambiente de desenvolvimento específico) para pessoas que não têm ou nunca tiveram experiência na confecção de algoritmos computacionais. É uma linguagem de código aberto desenvolvida por Mitchel Resnik e colaboradores. Resnick et al. (2009) afirma: “queríamos desenvolver uma abordagem para programação que cativasse pessoas que não tinham se imaginado antes como programadores”. O Scratch tem destaque no meio infanto-juvenil, porém também é utilizado por adultos que sentem a necessidade de conhecer a programação de uma maneira simples e divertida.

A programação lembra muito brinquedos de construção em blocos, no qual por meio de pequenas peças a pessoa pode dar asas a sua imaginação e criar praticamente tudo que desejar, uma das franquias mais famosas do mercado pertence ao grupo LEGO. Para construir seu código basta combinar diferentes tipos de blocos, arrastando-os e encaixando-os, como exemplificado na Figura 6, e verificar como as mascotes/atores se comportam. A mascote é um elemento auxiliar da ferramenta, que executa as ações construídas pelo usuário.



Figura 6. Bloco de código em *Scratch*.

¹ Link para acessar o Scratch: <https://www.scratch.mit.edu>

O Scratch conta com uma vasta gama de elementos que compõem um algoritmo, desde laço de repetição, variáveis, estruturas condicionais e operadores. A ferramenta oferece ainda ações sonoras e movimentos mais complexos, tal como também oferece extensões de novos blocos para atender a necessidade de usuários mais avançados.

Como linguagem aberta, o Scratch pode ser tanto instalada e utilizada em ambientes off-line como também pode ser utilizada *online* no próprio *site* da ferramenta, no endereço <https://scratch.mit.edu/>. Vale ressaltar que o Scratch em sua totalidade não consiste apenas de uma linguagem de programação, mas sim de um ambiente completo para execução dos algoritmos. Segundo Resnick et al. (2009) os três principais núcleos de *Design* do scratch são: Capacidade de construção/criação, maior significatividade e mais sociabilidade do que os outros ambientes de programação. A Figura 7 mostra o ambiente de desenvolvimento do Scratch presente no site do mesmo.

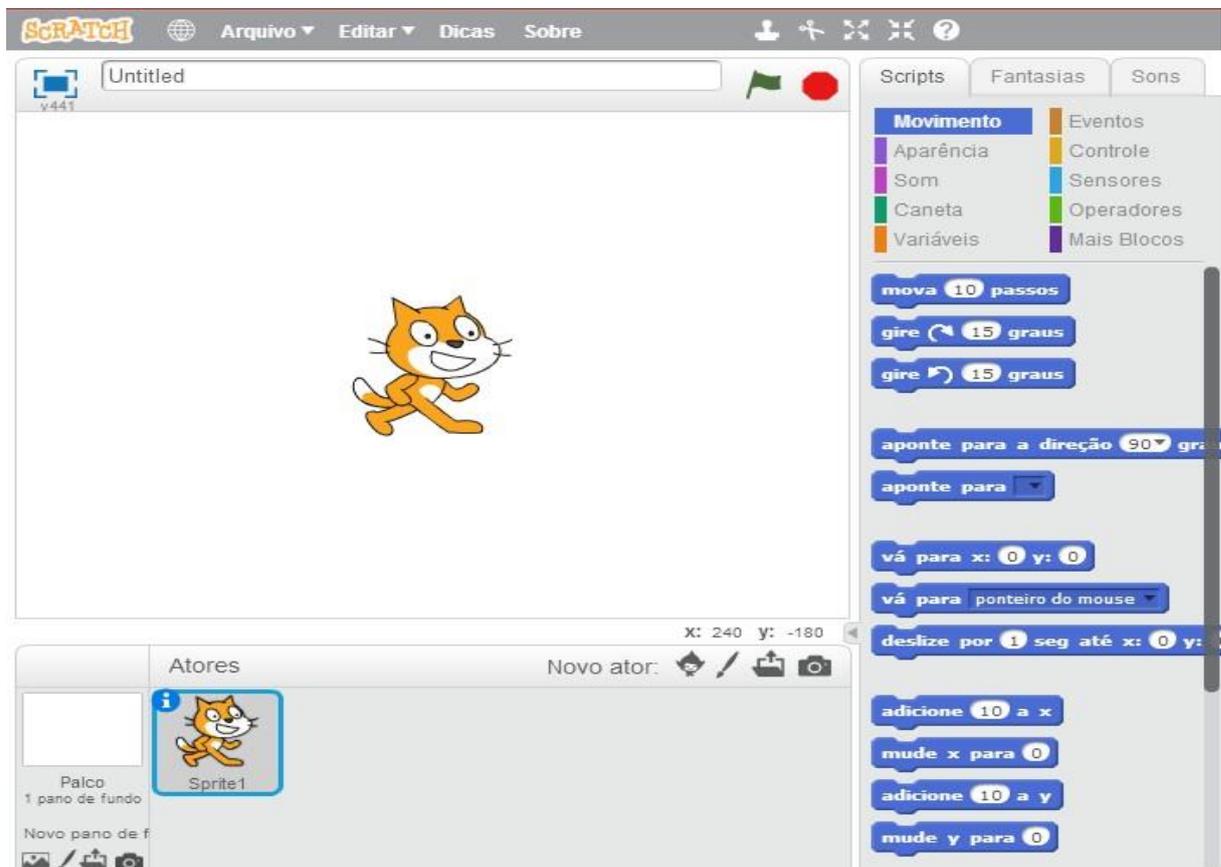


Figura 7. Ambiente de programação do *Scratch*.

3.2.2 Code Studio

O Code Studio é uma ferramenta que consiste da iniciativa code.org, a iniciativa conta com a ajuda de engenheiros da Google, Microsoft, Facebook e Twitter. A iniciativa do Code consiste em trazer cursos que incentivem a criação de código e o desenvolvimento do pensamento computacional por parte de crianças e adolescentes. A ferramenta lembra bastante o Scratch, os algoritmos são construídos em blocos que carregam alguma função semântica. Existem atores que interagem com o usuário e executam exatamente o que se é escrito pelos blocos construídos.

Existem algumas diferenças entre Code Studio e o Scratch: o Code Studio conta com ambiente mais simplificado e com atividades mais focadas no público infantil. Primeiramente, antes da maioria das atividades desenvolvidas dentro da ferramenta há um vídeo introdutório e motivacional, que explica um pouco do assunto e evidencia algumas das razões de se completar os exercícios.

O Code Studio conta com uma lista de tarefas divididas em cursos, nela o usuário tem a oportunidade de começar por algoritmos mais simples e evoluir até uma atividade mais complexa. Sempre escrevendo seus algoritmos de acordo com as recomendações e observando a ação dos atores.

A temática varia entre jogos e filmes para o público infanto-juvenil, o que serve para estimular a vontade de se executar as atividades dentro do ambiente. A figura 6 demonstra uma das fases de um curso dentro do Code Studio, a atividade utiliza o contexto já explorado pelo Scratch e é ambientado na temática do famoso jogo “Angry Birds”.

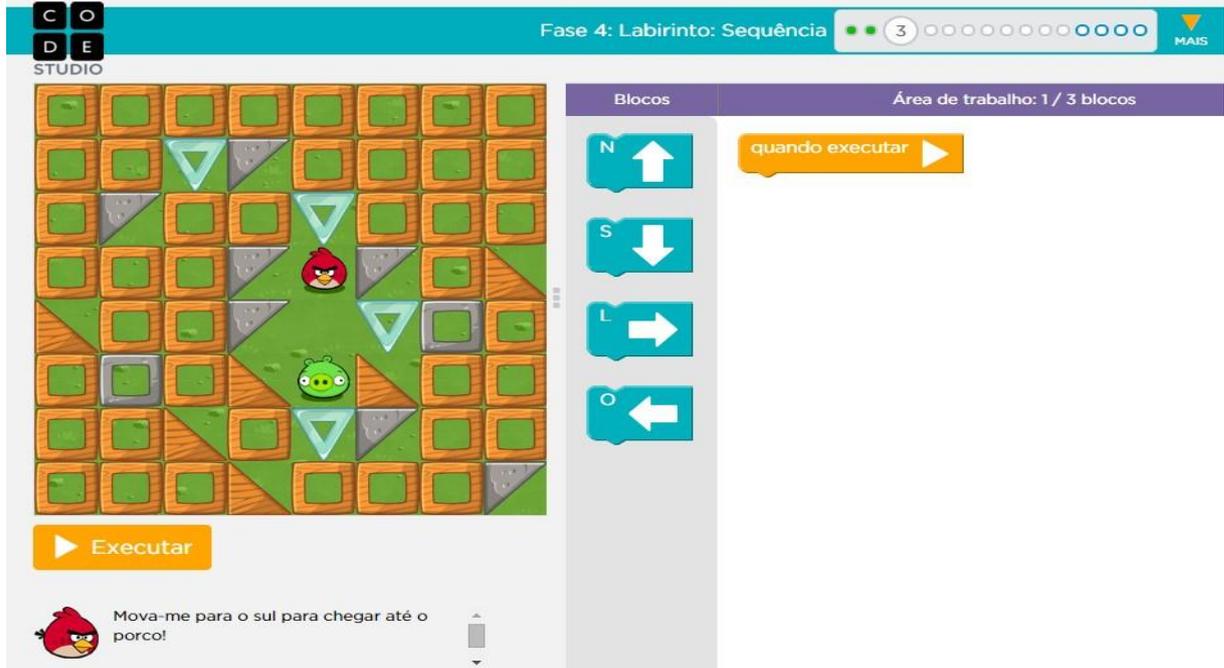


Figura 8. Interface de atividade do Code Studio.

3.2.3 Codecademy

O Codecademy é um site educativo, encabeçado pela companhia de mesmo nome, que busca ensinar estudantes interessados a aprenderem linguagens de programação de forma simplificada e interativa. Segundo a própria companhia (Codecademy, 2015): “Nós estamos comprometidos na criação de uma melhor experiência de aprendizado de dentro para fora, tornando o Codecademy o melhor lugar para nosso time aprender, ensinar, e criar a experiência de aprendizado do futuro”.

O Codecademy lembra muito um IDE (do Inglês, Integrated Development Environment ou Ambiente de desenvolvimento integrado) *online*, nele o estudante pode escolher entre cursos que abrangem desde um desafio oferecido pelo site, como por exemplo escrever um código HTML para aplicar efeitos em seu nome, até conceitos de Rails (Framework de desenvolvido da linguagem Ruby) mais completos. A figura a seguir demonstra uma atividade simples dentro do ambiente da ferramenta.

The screenshot shows the Codecademy interface for an HTML exercise. On the left, there's a sidebar with the title 'HTML Básico' and a description: 'navegador é transformar o código de test.html em uma página reconhecível! Ele sabe como estruturar a página seguindo a sintaxe HTML.' Below this is a section titled 'Instruções' with three numbered steps: 01. À direita, temos um arquivo test.html. 02. Substitua o texto na linha 2 (o trecho entre e) pelo que quiser! 03. Clique em Salvar e Enviar Código, e você vai ver como o arquivo test.html ficaria no navegador. Você viu isso? As tags deixaram nosso texto em negrito! At the bottom of the sidebar are links for 'Fórum de Perguntas &' and 'Glossário'. The main area is a code editor for 'test.html' with the following code: 1 - <!DOCTYPE html> 2 - Fique à vontade para modificar esse texto. 3 - Below the code editor are two buttons: 'Salvar e Enviar Código' and 'Restaurar'. At the very bottom, there's a navigation bar with a hamburger menu icon and the text '1. Por que aprender HTML?'.

Figura 9. Atividade de HTML Básico no Codecademy.

A ferramenta apresenta o seguinte funcionamento, ao lado esquerdo da tela estão as instruções para se resolver o exercício, nela há uma contextualização e também os passos que devem ser seguidos para se obter sucesso tal como algumas dicas importantes, do lado direito está o diferencial de se aprender programação no Codecademy, nele se encontra o campo de texto editável, onde você adiciona ou modifica o código na linguagem. Após você aprender escrevendo seu próprio código a ferramenta lhe oferece um *feedback* instantâneo, mostrando exatamente o que seu código escrito faria em uma condição real. As dicas oferecidas pelo Codecademy são diretas e a sua interface é simplificada, ajudando muito a identificação da informação e execução das atividades.

3.2.4 Khan Academy

O *Khan Academy* é um *software* educacional, desenvolvido inicialmente por Salman Khan, proprietário da organização não governamental de mesmo nome. A *Khan Academy* e seu *software* de apoio surgiram da vontade de Salman de mudar os paradigmas atrelados ao ensino e aprendizagem nas escolas do mundo todo.

O *software* é ferramenta central na filosofia da organização, através dele é possível a realização de atividades, tanto por parte do professor como também do aluno. São oferecidos conteúdos em diversas áreas do conhecimento traduzidos em um grande número de línguas faladas ao redor do mundo.

A principal função do *software* é permitir que alunos aprendam conteúdos diversos, por meio de vídeos e atividades feitas no ambiente. O aluno seleciona o conteúdo que deseja aprender, após isso ele deve responder às perguntas para avançar, nesse momento ele pode pedir dicas ou assistir um dos famosos vídeos da *Khan Academy* para tomar conhecimento do assunto e assim se habilitar em sua tarefa. Diferente dos *softwares* citados nessa secção, a *Khan Academy* não está diretamente ligada ao ensino e aprendizagem de algoritmos (apesar de conter esse módulo em sua base de dados), a mesma se encontra em um escopo mais amplo, que é a formação da base de conhecimento das pessoas em geral.

O ambiente da *Khan Academy* aplica conceitos de gamificação como método para incentivar os alunos a buscarem novos resultados dentro do *software*. Utilizando de premissas já citadas neste mesmo documento, na seção 2.2.1, o ambiente explora principalmente os conceitos de Propriedade e Posse, e Desenvolvimento e realização, concedendo medalhas para tarefas e objetivos cumpridos dentro da plataforma, permitindo verificar e alterar seu perfil com os bens adquiridos dentro do próprio ambiente. A figura (10) exemplifica uma atividade sendo executada dentro da *Khan Academy*.

Subtração de números decimais 2 Acerte os 1 primeiros, ou 5 em sequência

Faça exercícios práticos de soma de dois números dados até a primeira, segunda ou terceira casas decimais.

$6,87 - 0,483 = ?$

Resposta

Verificar resposta

Mostre-me como

Quero uma dica

Precisa de ajuda? Assista a um vídeo.

Subtração de números dec

Subtração de números decimais - Exemplo 2

Usar rascunho Relatar um erro nesta questão

Figura 10. Tela de atividade da Khan Academy.

3.2.5 Moodle

O Moodle é uma plataforma de aprendizado virtual *Open Source*, foi desenvolvido com o intuito de prover um ambiente robusto, seguro e integralizado capaz de permitir criações de ambientes de aprendizado personalizados por parte dos professores, administradores e aprendizes (MOODLE.). A ferramenta foi o resultado do Projeto Moodle que dirigido e coordenado pela empresa denominada Moodle HQ, empresa que contava com 30 desenvolvedores apoiados por financiadores do projeto em escala mundial.

A ferramenta engloba toda uma infraestrutura que vai desde a criação de cursos e atividades, até fóruns de discussão. O Moodle possibilita a criação de contas, upload de arquivos, inserção de vídeos etc. Além das suas atribuições básicas, como *software Open Source*, a ferramenta recebe contribuições de sua comunidade, as contribuições são concretizadas através de correções em possíveis defeitos e em *Plug-ins* que adicionam novas funcionalidades ao sistema.

A Figura 11 apresenta a tela inicial do Moodle, na qual o usuário pode visualizar diversas informações, informações essas escolhidas pelos administradores do sistema.

The screenshot shows the Moodle interface for the course 'Mundo do Grande Algoritmo'. At the top, the course name and language 'Português - Brasil (pt_br)' are visible, along with the user 'Sabria Briant'. The main heading is 'Trabalho de Conclusão de Curso'. Below this, there are two sections: 'Cursos' and 'Cursos disponíveis'. The 'Cursos' section has a dropdown menu set to 'Miscellaneous' and shows the course 'Mundo do Grande Algoritmo'. The 'Cursos disponíveis' section also shows 'Mundo do Grande Algoritmo' with a description: 'Você sabe o que é um algoritmo ? E o que seria o Grande Algoritmo ? Você gosta de fantasia ? Está disposto a viver grande aventuras ? Entre agora nesse mundo mágico e descubra coisas fantásticas.' To the right, there is a calendar for June 2016.

Figura 11 - Página inicial do Moodle.

3.2.6 Funifer

A ferramenta Funifier consiste de um ambiente implementador e catalizador de gamificações. Tem como propósito gamificar outros softwares implementando técnicas de gamificação propostas por Yukai com a finalidade de motivar os usuários do sistema à algum comportamento específico. A ferramenta é desenvolvida pela empresa de mesmo nome, a empresa possui parceiros em escala mundial e disponibiliza sua ferramenta por meio de contratos, tornando-a uma solução de gamificação paga.

A ferramenta conta com módulos que permitem que ela se integre com o software em questão e passe a coletar informações referentes a ações desempenhadas dentro do ambiente, além de coletar essas informações a ferramenta Funifier, também tem um módulo de avaliação dos dados. A ferramenta também possui feedback para o usuário, entregando elementos presentes nas técnicas de gamificação diretamente para o usuário enquanto o mesmo permanece no sistema que é alvo da gamificação.

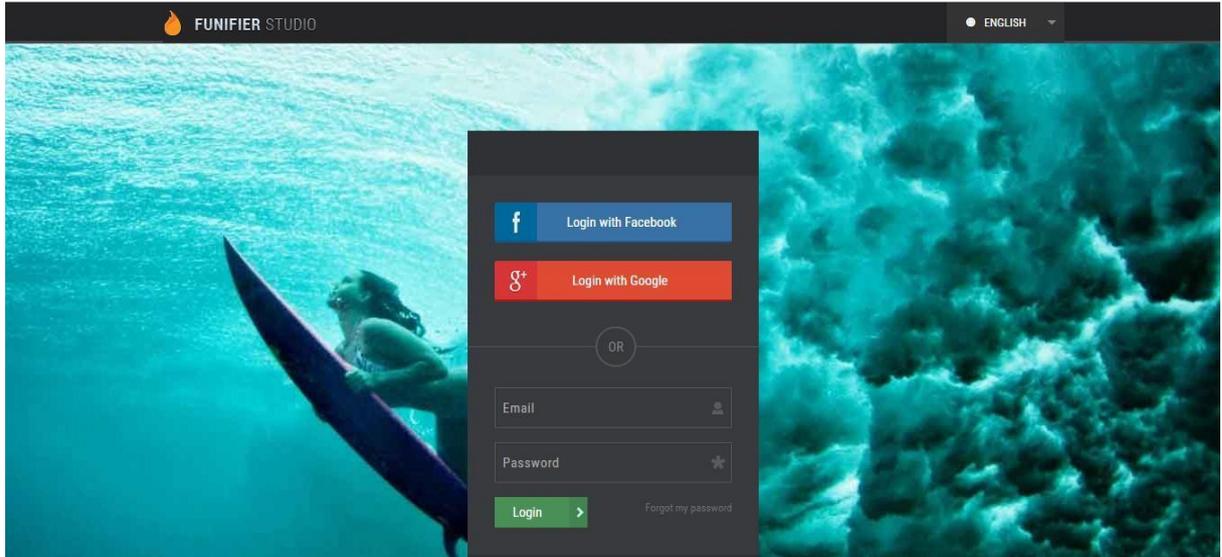


Figura 12 - Tela de Login Funifier.

3.3 CONSIDERAÇÕES FINAIS SOBRE O FERRAMENTAL TECNOLÓGICO

As ferramentas apresentadas na seção 3.2 serviram de base, tanto para uma avaliação do estado da arte dos ambientes virtuais como também para integrarem o ambiente virtual de aprendizado gamificado que foi adaptado neste trabalho. Dentre as ferramentas que foram expostas na seção 3.2 foram escolhidas três para a composição do ambiente de aprendizado, as ferramentas escolhidas foram: Scratch, Funifier e Moodle.

4. AMBIENTE DE APRENDIZADO GAMIFICADO

O trabalho realizado consistiu na adaptação e adequação de um ambiente de aprendizado virtual para ensino de algoritmos computacionais, com auxílio da Gamificação Yukai Chou por meio de seu Framework Octalysis. As atividades desempenhadas permeiam desde a configuração e manutenção do ambiente de software até o desenvolvimento do projeto de gamificação.

As seções a seguir detalham o ambiente de aprendizado gamificado construído. A seção 4.1 descreve a dinâmica das ferramentas, a seção 4.2 descreve a configuração do ambiente e por fim, a seção 4.3 descreve os elementos do projeto de gamificação.

4.1 PROJETO DE GAMIFICAÇÃO

Para se aplicar a gamificação em si e obter os resultados esperados da técnica, primeiramente é necessário se fazer um projeto de gamificação. O projeto de gamificação utilizado foi baseado no processo proposto por Yukai Chou e conta com auxílio de recursos oferecidos pela ferramenta Funifier. Para que a gamificação não seja fadada ao insucesso, o foco do levantamento dos requisitos, que são traduzidos em objetivos por Yukai, devem estar no usuário, ou seja, no jogador. Muitas gamificações ficam estagnadas no processo PBL (Points, Badges and Leaderboards - Pontos, Medalhas e Classificações), que consiste basicamente na distribuição de incentivos (Pontos e medalhas) e classificação de indivíduos, porém para se alcançar um nível mais elevado na gamificação é necessário manter o foco no usuário para poder avaliar como ele deve se sentir ao utilizar o software ou desempenhar uma ação qualquer.

No início do projeto, foi definido que o estudante deveria se sentir motivado ao longo das suas atividades no curso, por meio do desligamento aparente das atividades com conteúdo normalmente ensinados em sala de aulas convencionais. Por mais que o conteúdo mantenha sua essência, a forma de aprendizado se dá por um meio mais informal, imergindo o estudante em um jogo lúdico em que ele assume um personagem e aprende à medida que é desafiado dentro do curso. Após a definição de como o estudante deveria se sentir e como deveria se comportar dentro do curso, as técnicas para se conduzir o processo foram levantadas. As técnicas utilizadas no

projeto foram baseadas do Framework Octalysis. A lista completa das técnicas utilizadas se encontra no apêndice C.

As técnicas foram organizadas para melhor se adequarem ao momento em que o estudante se encontra no ambiente, seja esse momento de descoberta da gamificação, ou seja, um momento de finalização das atividades de um determinado módulo, ou até mesmo a despedida do ambiente. O projeto de gamificação do Octalysis divide-se em um ciclo de vida composto de 4 fases, essas fases são as de DESCOBERTA, ENTRADA, DIA-A-DIA, FIM DE JOGO. As técnicas são distribuídas em cada fase de modo a conduzir a experiência do estudante dentro da gamificação, sendo que não há problemas na repetição de técnicas em uma ou mais fases. A distribuição das técnicas se encontra na Figura 13.

TÉCNICAS	Descoberta	Entrada	Dia-a-Dia	Fim de Jogo
				Amizades
			Dinâmica do Compromisso	Rankings
			Coleção	Coleção
			Estante de Troféus	Elitismo
		Dinâmica de compromisso	Brag Buttons	Estante de Troféus
		Coleção	Avatar	Avatar
		Avatar	Escolhas Significativas	Dinâmica do Compromisso
Medalhas		Brag Buttons	Orientações	Brag Buttons
Pontos		Sinfonia de aumento de nível	Rankings	Orientações
Avatar		Orientações	Amizades	Amizades
Sorte de principiante		Medalhas	Medalhas	Pontos de Status
Curva de Aprendizado		Pontos	Pontos	Pontos
Narrativa	Narrativa	Narrativa	Narrativa	Narrativa
	Descoberta	Entrada	Dia-a-Dia	Fim de Jogo
FASES				

Figura 13 - Distribuição das técnicas.

Dentro do projeto de gamificação há uma divisão de níveis, no qual a gamificação amadurece e alcança novos resultados, o nível um consiste na divisão das fases e atribuições das suas respectivas técnicas. O segundo nível do Octalysis consiste em traçar uma Gamificação que abranja fases dentro de fases, que nada mais é que definir sub fases para cada macro fase. Uma forma simples de se visualizar esse procedimento vem com o seguinte exemplo: dentro das 4 fases existentes dentro do Octalysis existem mais 4 sub fases, há a sub fase descoberta dentro da fase descoberta, há a sub fase entrada dentro da fase descoberta, há a sub fase

descoberta dentro da fase fim de jogo e assim por diante, para todas as fases, como apresentado na figura 10.

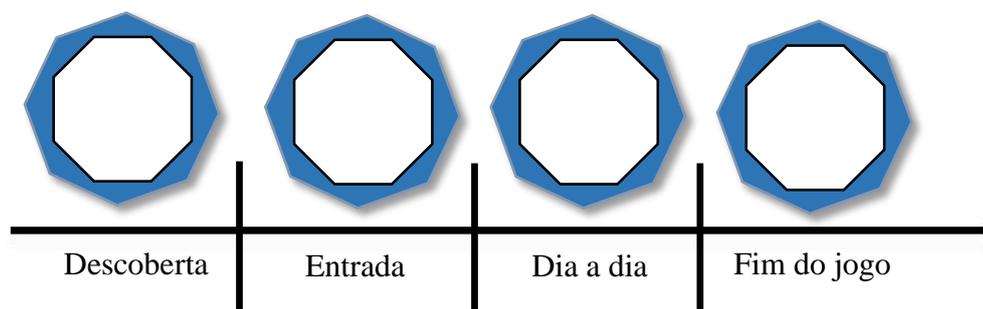


Figura 14 - Nível 2 Octalysis.

Na gamificação implementada no trabalho também foi utilizada o nível 2, no qual procura-se manter um fluxo nas atividades desempenhadas para se quebrar a monotonia, e até mesmo a criação de uma rotina desgastante. O jogador deve sentir uma evolução constante de forma a perceber essa evolução em suas ações dentro do ambiente, no caso da gamificação do ambiente virtual de aprendizado aqui apresentada o exemplo.

A divisão das técnicas utilizadas nas subfases encontra-se organizada da seguinte forma:

4.1.1 Descoberta

Descoberta

- Narrativa: Será mostrado por parte da narrativa que o Mundo do algoritmo existe.

Entrada

- Narrativa: Mais detalhes iniciais do Mundo do Grande algoritmo serão introduzidos.

Dia-a-Dia

- Sorte de principiante: Será presenteado com algo que o faça se sentir especial, fazendo-o pensar que tem sorte ou que foi escolhido por um propósito maior.
- Narrativa: O estudante já deve estar se ambientando ao curso e já deve estar a par do que está por vir.

Fim de jogo

- Narrativa: O estudante saberá que o tempo de se ambientar acabou, ele faz, finalmente, parte do mundo de grande algoritmo.
- Avatar: Nesse ponto o aluno poderá escolher um avatar (Foto de perfil) que irá lhe representar dentro do curso.
- Pontos: O conceito de pontos será apresentado ao estudante, pontos serão dados por ele estabelecer seu avatar, por entrar no curso e ter contato com o Mundo do Grande Algoritmo.
- Medalhas: O conceito de medalhas será apresentado ao estudante, medalhas serão dadas por ele estabelecer seu avatar, por entrar no curso e ter contato com o Mundo do Grande Algoritmo.

4.1.2 Entrada

Descoberta

- Narrativa: Os primeiros desafios serão apresentados ao estudante, mais detalhes do ambiente serão apresentados por meio da narrativa.
- Pontos: Pontos serão distribuídos por completar sua primeira atividade.
- Medalhas: Medalhas serão distribuídas por se completar ações desejadas, completar 1 atividade.

Entrada

- Pontos: Pontos serão distribuídos por completar sua primeira atividade, quiz etc.
- Medalhas: Medalhas serão distribuídas por se completar ações desejadas, completar 1 atividade, 5 atividades etc.

- Brag buttons: o estudante poderá mostrar em suas redes seus primeiros avanços dentro do Mundo do Grande Algoritmo.
- Amizades: os estudantes poderão fazer suas primeiras amizades.
- Curva de aprendizado: A simplicidade dos níveis iniciais está ficando para trás, agora que o estudante se sente detentor de seu conhecimento ele está pronto para adquirir um mais elevado.

Dia-a-Dia

- Avatar: O avatar poderá ser substituído a qualquer momento.
- Orientações: Sempre que desejar o aluno poderá requisitar orientações de grandes conselheiros experientes (Monitores) para seguir em frente com suas jornadas.
- Pontos: Pontos serão distribuídos por completar as atividades iniciais, quiz etc.
- Medalhas: Medalhas serão distribuídas por se completar ações desejadas, completar 1 atividade, 5 atividades etc.

Fim de jogo

- Escolha significativas: Este é o momento de escolher entre o caminho da Espada, Arco ou Cajado.
- Narrativa: A narrativa deve tornar o momento da escolha algo muito significativo para o estudante.
- Pontos: Pontos serão distribuídos para recompensar sua transição e escolha
- Medalhas: Medalhas serão distribuídas para recompensar sua transição e escolha.

4.1.3 Dia-a-dia

Descoberta

- Narrativa: Os níveis intermediários serão introduzidos, o estudante agora é um aventureiro experimentado dentro do mundo do grande algoritmo.

Entrada

- Thank-You Economy: Os alunos poderão ajudar um ao outro dentro do ambiente, tirando dúvidas e propondo soluções juntamente de seus colegas, gerando assim a agradecimentos e motivação.
- Orientações: Poderá pedir ajuda à monitores ou até mesmo ao professor (que portarão seus respectivos nomes de fantasia) para lhe auxiliarem no ambiente de aprendizado.

Dia-a-Dia

- Avatar: O avatar poderá ser substituído a qualquer momento.
- Dinâmica de compromisso: Atividades semanais, desafios apresentados para os estudantes, pontuando e medalhando os melhores colocados e os que concluíram as atividades.
- Thank-You Economy: Os alunos poderão ajudar um ao outro dentro do ambiente, tirando dúvidas e propondo soluções juntamente de seus colegas, gerando assim agradecimentos e motivação.
- Dinâmica de aumento de nível: Agora ele poderá progredir para os níveis avançados.
- Curva de aprendizado: Cada vez mais complexos, porém cada vez mais dentro do círculo de propriedade intelectual do aluno.
- Dinâmica de compromisso: Atividades semanais, desafios apresentados para os estudantes, pontuando e medalhando os melhores colocados e os que concluíram as atividades.
- Pontos: Pontos serão mais abundantes do que anteriormente, porém também serão distribuídos em tarefas de maior dificuldade.
- Medalhas: Medalhas serão mais raras, porém mais valorosas.

Fim de jogo:

- Narrativa: Mais uma etapa do curso se finalizou, a etapa final está por vir e o Mundo do Grande Algoritmo está por revelar seus maiores segredos.
- Elitismo: A elite do Mundo do grande Algoritmo será apresentada, agora mais próxima do que nunca.

4.1.4 Fim de jogo

Descoberta

- Narrativa: Um novo e derradeiro momento dentro do curso é apresentado ao estudante.
- Sinfonia de aumento de nível: As últimas classes (especialistas) são introduzidas aos estudantes, o seu status dentro do jogo mudou.
- Elitismo: Agora ele faz parte de um grupo seleta, a elite do Mundo do Grande algoritmo.

Entrada

- Narrativa: A narrativa deve mostrar ao estudante que o final está próximo, e nessa etapa desafios enormes estão por vir.

Dia-a-Dia

- Avatar: O avatar poderá ser substituído a qualquer momento.
- Coleção: Com sua coleção bem consolidada agora o estudante poderá juntar itens mais raros dentro do ambiente.
- Ranking: Poderá se tornar o Rei do Grande algoritmo ou membro da cúpula.
- Pontos: Os pontos acumulados neste momento são abundantes, por mais que as atividades sejam complexas os pontos serão recompensadores.
- Dinâmica de compromisso: Atividades semanais, desafios apresentados para os estudantes, pontuando e medalhando os melhores colocados e os que concluíram as atividades.
- Orientações: O aluno poderá passar para o outro lado das orientações, ajudando os que tem dificuldade e os em estágios mais iniciais.
- Medalhas: As medalhas mais raras do ambiente serão distribuídas para os jogadores que atingiram o nível mais alto do Mundo de Grande algoritmo.

Fim de jogo

- Estante de troféus: poderá verificar todas as conquistas que adquiriu dentro do curso ao longo do mês.

- Narrativa: O fim do curso chegou e a história pode ser encerrada com o desfecho épico, mas sempre encorajando novas descobertas.
- Brag Button: Poderá mostrar para os amigos que concluiu o curso de algoritmos.
- Amizades: O curso acabou, mas as amizades construídas podem ser mantidas, tanto por afinidade quanto por interesse nas mesmas áreas de conhecimento

4.1.5 Técnicas utilizadas na gamificação implementadas

Aqui são descritas as principais técnicas utilizadas no ambiente gamificado e implementado. São explicadas as técnicas e seu conteúdo, tal como o nível em que são utilizadas.

4.1.5.1 Narrativa

Para criação de um ambiente de fantasia dentro do Moodle foi utilizada a técnica de narrativa, a qual tenta, por meio de uma história direcionada para o usuário inseri-lo em um contexto desejado. O contexto escolhido para a narrativa foi o “Mundo do Grande algoritmo”, que nada mais é que um mundo fictício criado pelo autor. Este contexto transfere o estudante a um papel de aventureiro, no qual deve aprender e executar desafios dentro desse novo mundo para ganhar prestígio e poder. Foram separadas medalhas e níveis baseados no contexto para melhor inserção na fantasia. Todas as atividades presentes no Moodle tratam o algoritmo como algo mágico e especial e os conteúdos e atividades são apresentadas se utilizando do artifício da narrativa.

4.1.5.2 Nível

Para acompanhamento e incentivo na progressão dos estudantes foi definida uma dinâmica de níveis, no qual se é possível progredir à medida que se ganha pontos e completa atividades. Os níveis foram divididos de maneira hierárquica e tem a seguinte granularidade:

- Iniciado;
- Aprendiz;
- Intermediários;
- Avançados;
- Especialistas;
- Alto Conselho;
- Grande Rei.

Os níveis de “Alto conselho” e “Grande Rei” são distribuídos aos participantes com mais pontos na semana e servem como um incentivo a mais dentro da dinâmica de aquisição de pontos. Esses dois níveis também fazem parte da técnica de elitismo, que é explicada nesta mesma secção.

Os níveis ainda podem variar de estudante para estudante à medida que ele escolhe qual “caminho” seguir, o caminho foi criado para dar uma escolha significativa ao estudante, que pode escolher uma classe de personagem que mais se identifica com suas características.

4.1.5.3 Medalhas

As medalhas são um tipo comum de incentivo na gamificação em geral, você incentiva positivamente o jogador com prêmios específicos, baseados em ações que o mesmo desempenhou. Dentro do ambiente as medalhas têm o mesmo fim, incentivar os jogadores com algo significativo, sendo atribuído a eles, a cada novo desafio superado.

4.1.6.4 Pontos:

Tal qual as medalhas, os pontos são técnicas comuns quando tratamos de gamificação, o ambiente distribui pontos aos estudantes por atividades que realizam dentro do ambiente. Os pontos incentivam os estudantes a continuarem desempenhando as atividades em busca de progressão e acúmulo dos mesmos, trazendo realização e status dentro do contexto, os pontos servem ainda de apoio à outras técnicas de gamificação, tal qual as classificações e acompanhamento da progressão.

4.2 DINÂMICA DAS FERRAMENTAS

As seções a seguir descrevem o funcionamento e papel das três partes do trabalho Moodle, Funifier e Scratch. O que cada uma representa no ambiente e como ela pode contribuir para se alcançar o objetivo da gamificação. A Figura 15 demonstra a comunicação dos elementos do ambiente.

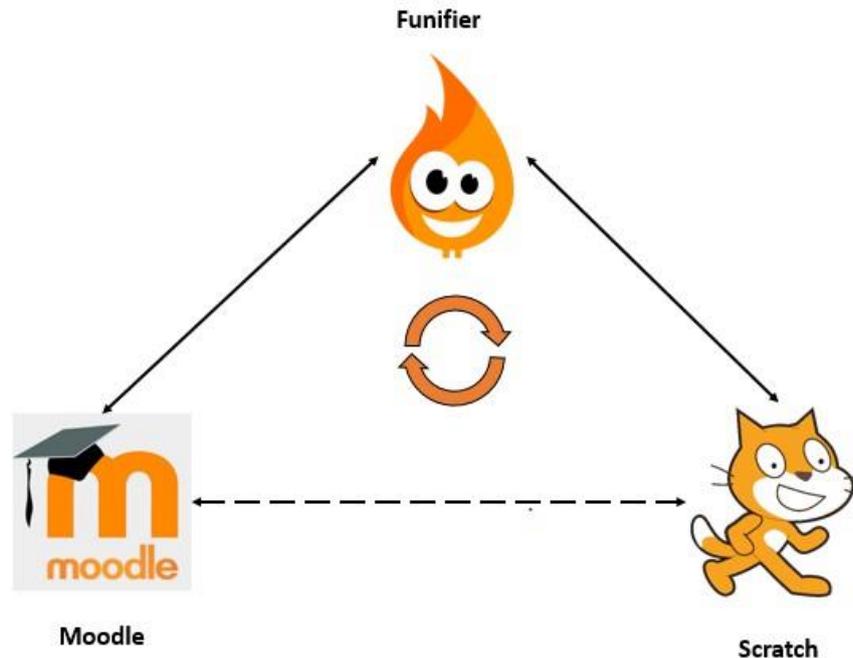


Figura 15 - Modelo visual de integração. Fonte: Autor.

4.2.1 Funcionamento do Moodle

O Moodle guarda as informações referentes às atividades do curso. A parte prática está contida neste ambiente, onde as atividades começam e também onde as atividades terminam. O Moodle é o ambiente no qual o estudante poderá ver seus resultados da gamificação, medalhas, pontos, coleção, avatar e etc.

As amizades e tutorias se encontram presentes no Moodle. A dinâmica de compromisso também está presente nessa plataforma por meio de atividades agendadas dentro do ambiente.

Informações da conta dos estudantes e de tarefas cumpridas também ficam a encargo do Moodle, tal como o material didático: PDFs, livros, vídeos etc.

4.2.2 Funcionamento do Scratch

O Scratch é utilizado para realização de atividade de construção de algoritmos do curso. O estudante constrói seus algoritmos dentro do Scratch e submete os resultados no Moodle. O uso do Scratch visa trazer mais motivação para a realização da criação de algoritmos. Apresentando uma forma lúdica de criação de algoritmos.

Não é possível, sem o acesso ao código do Scratch, o acoplamento de uma instância do *software* Scratch dentro do Moodle, porém há a possibilidade de se adicionar um link redirecionando para o ambiente de criação Scratch dentro do Moodle.

4.2.3 Funcionamento do Funifier

O Funifier funciona como o agente que aplica uma parte da gamificação no ambiente do Moodle, técnicas previamente estabelecidas dentro do projeto de gamificação são estruturadas e aplicadas no Funifier. Os pontos, *badges* e *rankings* são estabelecidos no Funifier. As notificações de conquista e algumas interações diretas do usuário com a gamificação estão no Funifier. É por meio do plug-in do Funifier que há a interação do Moodle com a gamificação.

A parte de análise de dados ficará ao encargo do Funifier: quantas medalhas foram distribuídas, quantos pontos foram acumulados, quantos escolheram determinado caminho. Essas informações de feedback estarão guardadas no Funifier para que haja o melhoramento da gamificação.

4.3 CONFIGURAÇÃO DE AMBIENTE

A seção discorre sobre as ações necessárias para se produzir um ambiente funcional. O que deve ser feito para que a configuração ocorra da forma correta e se possa utilizar, de fato, o ambiente completo e gamificado. As seções a seguir descrevem como se instalar o Moodle, como se configurar o Funifier e como integrar as três partes do ambiente, respectivamente.

4.3.1 Instalação Moodle gamificado

A instalação do Moodle se dá por meio dos procedimentos presentes no apêndice B. A versão do Software Moodle utilizada foi a 3.0.3, instalada num servidor Apache utilizando o Banco de dados MySQL. O Software está configurado localmente na máquina utilizada para a adaptação do ambiente.

O Moodle por si só não é gamificado, para que o mesmo seja gamificado é necessário implementar as técnicas presentes no projeto de gamificação. Boa parte do projeto de gamificação é implementado por meio da integração Moodle e Funifier,

descrita na seção 4.2.3. O projeto de gamificação está descrito na seção 4.3 deste mesmo capítulo.

4.3.2 Configuração Funifier

Para a utilização do Software Funifier é necessário acessar o site do mesmo. A aquisição da ferramenta é paga, e pode ser adquirida no site: www.funifier.com. Para a realização das atividades do Trabalho de Conclusão de curso a plataforma web de gamificação da empresa Funifier foi utilizada, a mesma encontra-se disponível no seguinte endereço: studio2.funifier.com, a ferramenta foi cedida pela empresa à Universidade de Brasília para utilização acadêmica.

Para a utilização dos recursos do Funifier é necessário logar no sistema e criar uma gamificação. Procedendo da seguinte forma:

- Escolhendo a opção de “New Gamification”;
 - Das opções de objetivos de gamificação / Gamification Business Goal selecionar a opção Engage students Virtually / Engajar os estudantes virtualmente.
 - Defina o nome da sua gamificação e descrição da sua gamificação;
- O ambiente em que as ações são executadas por ser visualizado na Figura 16.

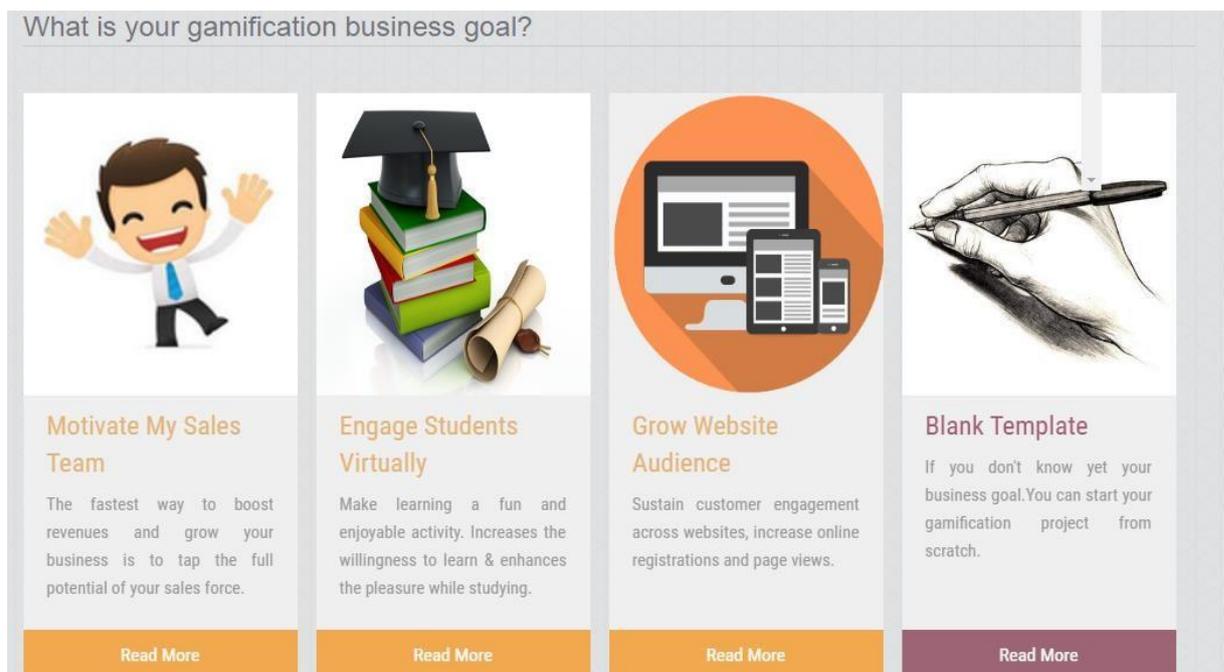


Figura 16 - Criando uma Gamificação Funifier.

A opção de *Engage students Virtually* já possui as ações básicas que são automaticamente atreladas ao Moodle, após sua configuração e integração já oferecem *feedback* (A integração do Moodle + Funifier encontra-se na seção 4.2.3).

Após a criação de uma gamificação o usuário do Funifier é direcionado para a página principal do sistema. Nesta página há um menu que se encontra no canto esquerdo da tela e que concentra as funcionalidades do software, como demonstrado na Figura 17.

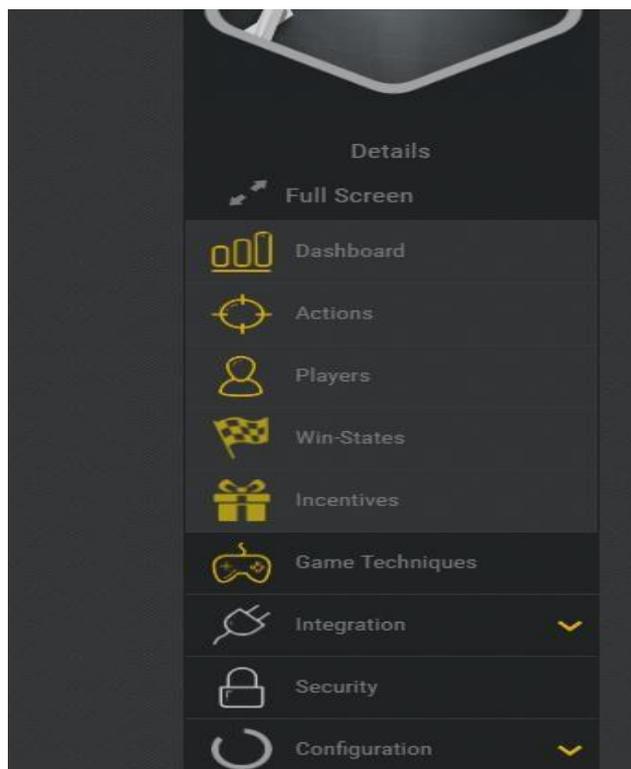


Figura 17 - Menu Funifier.

As seções a seguir apresentam os procedimentos para a adição dos principais elementos de gamificação oferecidos pelo Software Funifier.

4.3.2.1 Ações

Para implementar as ações que os estudantes devem desempenhar para ganhar pontos deve-se ir no menu esquerdo e selecionar a opção Actions. Dentro da opção Actions deve-se selecionar o botão “New” e preencher os campos necessários. Os campos são:

- Nome (O nome da ação);
- Seleção de imagem (A imagem que representará a ação);

- Atributos (São os atributos inerentes da ação, como courseid, contextid e etc);
- Pontos (São os pontos que o usuário adquire após realizar a ação);
- Notificações (O *feedback* que o sistema dará ao usuário após o mesmo completar uma ação);
- Ações correlatadas (Cria um link entre a ação que está sendo criada e ações existentes).

A tela de criação pode ser visualizada na Figura 18.

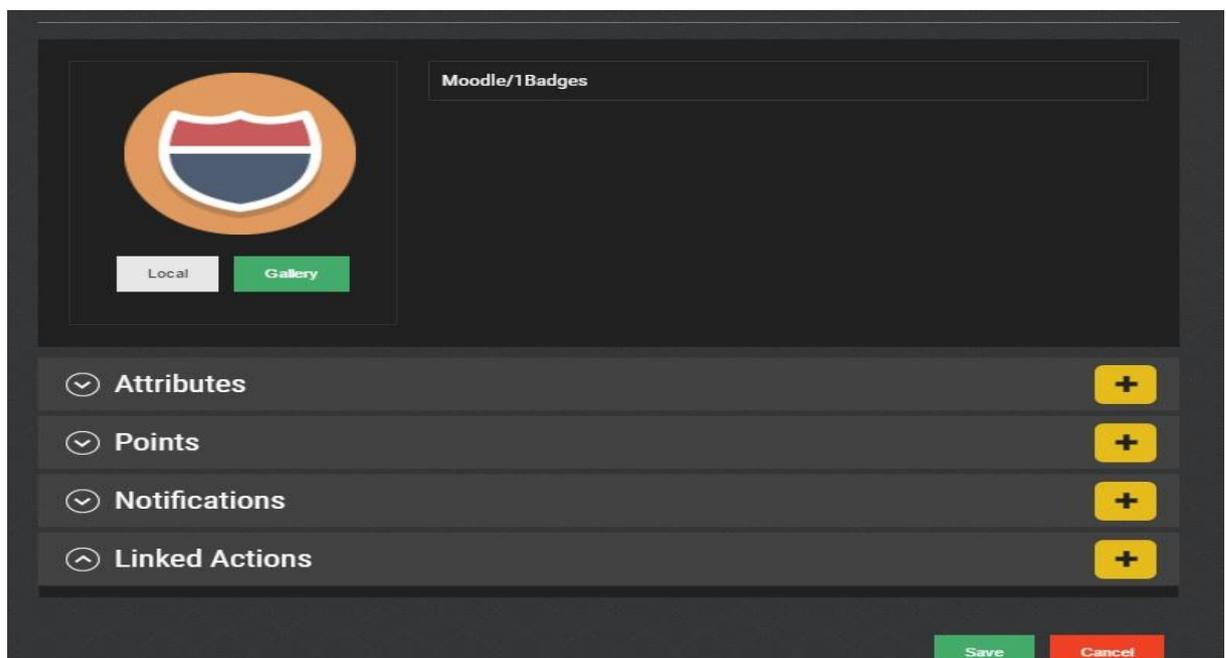


Figura 18 - Criação de Ações Funifier.

4.3.2.2 Níveis

Para definição dos Níveis que os jogadores vão adquirir ao longo da gamificação deve-se ir no menu esquerdo e selecionar a opção Configuration > Levels. Clicando no botão “New” e preenchendo os seguintes campos para todos os níveis definidos no projeto de gamificação, os níveis estarão prontos para serem utilizados:

- Nome;
- Seleção de imagem;
- Descrição;
- Pontos;
- Restrições;
- Notificações.

A Figura 19 mostra a tela de criação de um nível dentro do Funifier.

Figura 19 - Criação de níveis Funifier.

4.3.2.3 Medalhas

Para definição das Medalhas que os jogadores vão adquirir ao longo da gamificação deve-se ir no menu esquerdo e selecionar a opção “Win-states”, na página de “Win-states” deve-se clicar no botão “New” e preencher os campos que seguem. As ações devem ser repetidas para todas as medalhas definidas no projeto de gamificação:

- Nome: Leva o identificador da medalha;
- Descrição: Uma breve descrição do que é a medalha;
- Seleção de imagem: A imagem que irá representar a medalha;
- Ações necessárias: As ações atreladas à medalha, que devem ser realizadas para sua conquista;
- Notificações: Os textos ou vídeos que deverão aparecer quando a medalha for conquistada;
- Pré-requisitos: Os pré-requisitos necessários para receber a medalha, são eles os pontos, itens e níveis;
- Quem: Servem para restringir a medalha à um grupo específico;
- Restrição de quantidade: A quantidade máxima da medalha que um jogador pode adquirir;

- Restrição de tempo: O intervalo de tempo no qual a medalha estará disponível.

A Figura 20 demonstra a tela de criação de medalhas dentro do Funifier.

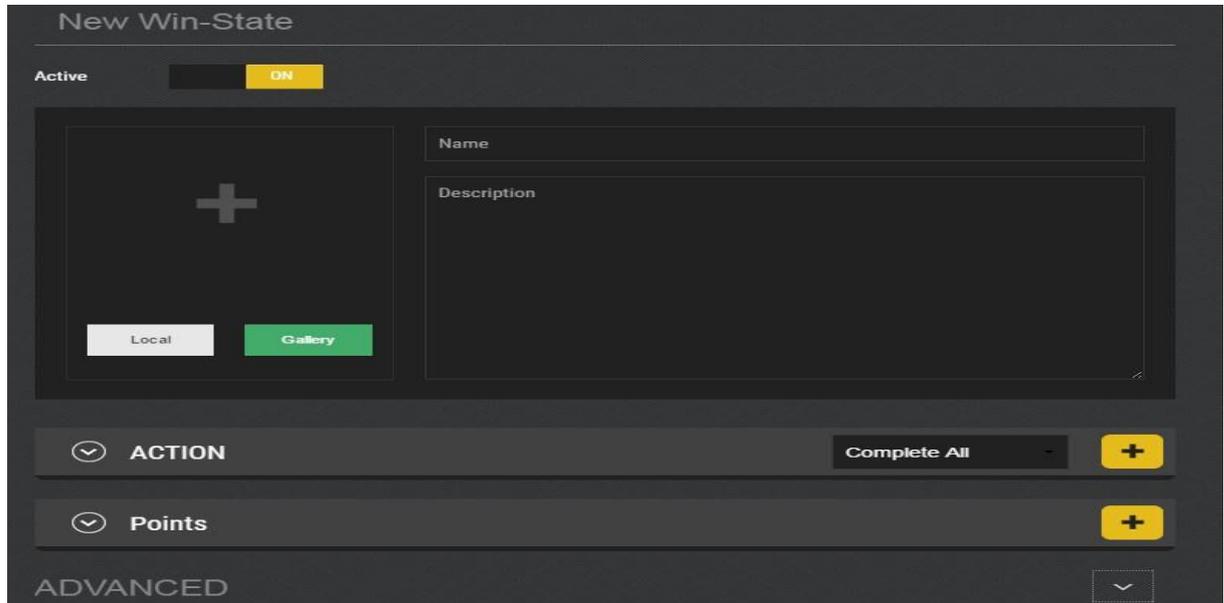


Figura 20 - Criação e Medalhas Funifier.

Há também os campos Tag e Itens, porém estes não são necessários no escopo do projeto e assim foram deixados em branco em todas as medalhas.

4.3.3 Integração entre Funifier e Moodle

A integração do Moodle e Funifier é o que de fato o transforma em um ambiente de aprendizado virtual gamificado. Em termos técnicos, a integração do Moodle com o Funifier se dá por meio de instalação de alguns *plug-ins*. Para que a integração ocorra é necessário que se tenha acesso de administrador ao Moodle e uma conta no Funifier. As seções que seguem descrevem os passos necessários para a instalação dos *plug-ins*.

4.2.3.1 Instalando a administração

O *plug-in* de administração é o componente principal da integração entre o Moodle e o Funifier, sendo necessário para a instalação dos outros *plug-ins*. Os passos a seguir descrevem as ações necessárias para sua instalação.

- Fazer o Login como administrador na instância do Moodle;
- Fazer download do arquivo funifierModule.zip(<http://studio2.funifier.com/files/moodle/funifiermodule.zip>);
- O arquivo pode ser encontrado clicando no menu esquerdo do Funifier e selecionando *Integration* > Moodle, como apresentado na Figura 21.

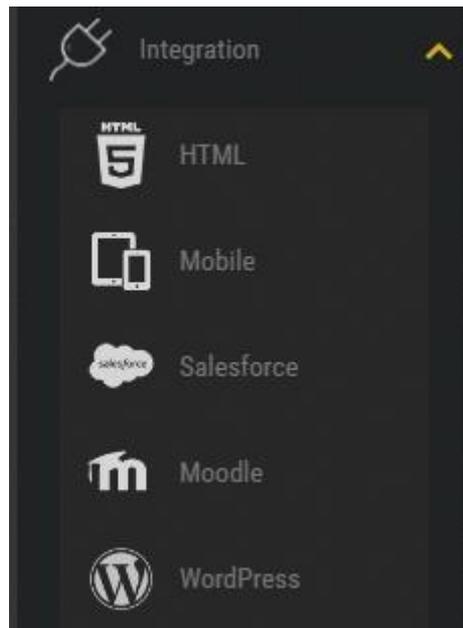


Figura 21 - Menu Integration do Funifier.

- Ir em ADMINISTRAÇÃO > Administração do site > Plug-ins > Instalar Plug-ins, como apresentado na Figura 22.

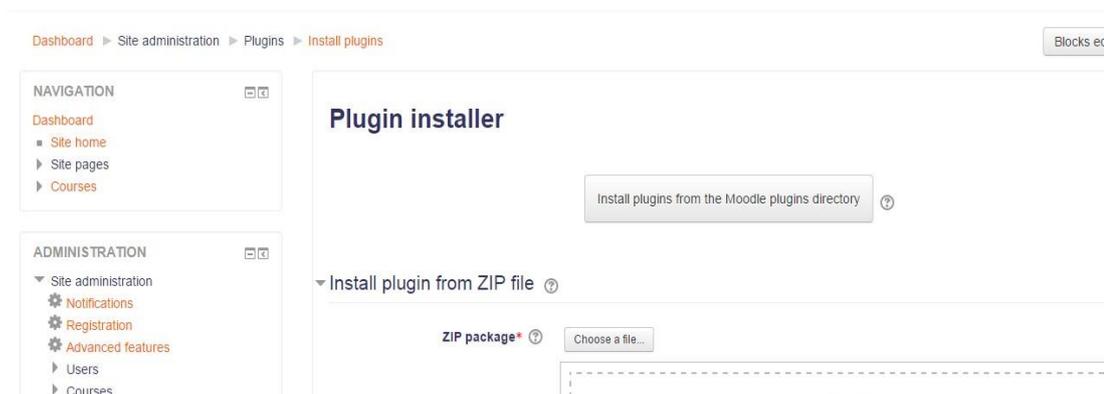


Figura 22 - Instalador de Plug-ins Moodle.

- Na Área de “instalar o Plug-in de um arquivo ZIP”;
- Selecione “Modulo de atividade (local) ”, no campo Tipo de Plug-in;
- Fala o Upload do arquivo “funifierModule.zip” no botão de upload de pacote zip;
- Confira os termos e clique em “Instalar Plug-in de arquivo ZIP”;
- Clique em “Instalar Plug-in! ”, na tela de validação de Plug-in;
- Clique em “Atualizar a base de dados Moodle agora”, na página de conferencia do Plug-in;
- Clique em “Continuar”, na página de Upgrading de nova versão;
- Nas configurações do Funifier adicione os seguintes valores:
- Client URL: <https://client2.funifier.com>;
- Service URL: <https://service2.funifier.com>;
- Key: [Esse valor depende da aplicação e pode ser encontrado nas informações do Funifier];
- Secret "Você pode criar uma chave secreta na área de segurança no Funifier";
- Clique em "Save changes".

4.2.3.2 Instalando o modulo *Listener*

O plug-in *Listener* é responsável pela comunicação, de fato, entre o Funifier e o Moodle. O *listener* insere as funções que “escutam” as ações realizadas dentro do Moodle e envia um feedback para o Funifier, que analisa a procedência das ações e retorna, caso esteja especificado, os elementos de gamificação, como por exemplo os pontos e medalhas. Os passos que seguem descrevem as ações necessárias para se configurar o módulo *Listener*.

- Faça o Login como administrador na sua instância do Moodle;
- Faça o Download do arquivo `funifierPlug-in.zip` (<http://studio2.funifier.com/files/moodle/funifierPlug-in.zip>) ;
- Vá em ADMINISTRAÇÃO > Site Administration > Plug-ins > Install Plug-ins
- Na área “Instalar Plug-in from ZIP”;
- Selecione no campo de tipo de Plug-in “Local Plug-in(local)”;
- Faça o upload do arquivo “funifierPlug-in.zip” no botão de upload de pacotes”;
- Marque os termos e clique em “Install Plug-in from the ZIP file”;
- Clique no botão “Install Plug-in! ”, na página de validação de plug-in;

- Clique em “Upgrade moodle database now”, na página de checagem de plug-ins;
- Clique em “Continue”, na página de upgrade de nova versão.

Como observador os dois procedimentos seguem a mesma linha de ações, mesmo culminando em elementos diferentes da integração. Após a instalação dos Plug-ins o Moodle se encontrará integrado com o Funifier.

4.3.4 Integração entre Funifier e Scratch

A integração do Scratch juntamente com o Funifier tem o intuito de coletar as ações desempenhadas dentro do Scratch. Para conduzir a Gamificação é necessário a utilização da opção *Inline* ou Developer Plug-in, do Funifier. Essa opção não é a mais aconselhável, porém foi a maneira utilizada porque a configuração de *listeners* por meio de injeção dos *scripts Funifier* requer acesso direto ao código e, mesmo que o software do Scratch seja livre, ele não permite que não desenvolvedores Scratch façam modificações no site.

Para proceder com a integração siga os passos:

- Na página principal do Funifier selecione a opção “Integration”, no menu esquerdo;
- Selecione a opção de HTML;
- Faça o download e instalação do Plug-in para o Browser, obs: só há a opção de plug-ins para os Browsers Firefox e Chrome;
- Para utilização no Firefox proceda da seguinte maneira:
 - Digite "about:config" na barra de endereço do seu navegador;
 - A chave "xpinstall.signatures.required" foi identificada;
 - Defina-se o valor como “false”;
 - Instale o Plug-in que foi baixado na aba HTML do Funifier;
- Para utilização no Chrome proceda da seguinte maneira:
 - Após fazer o Download do arquivo na aba HTML do Funifier;
 - Digite "chrome://extensions" na barra de endereço do navegador;
 - Arraste e solte o arquivo baixado para instalação do plug-in;

- Ative o plug-in no browser e adicione o nome de usuário de desenvolvedor ativo (O e-mail cadastrado também poderá ser utilizado), como apresentado na Figura 23. Para verificar se o usuário é um desenvolvedor ativo ou para atribuir essa função a ele, vá em “User”, clique em editar e marque a opção “Developer”;

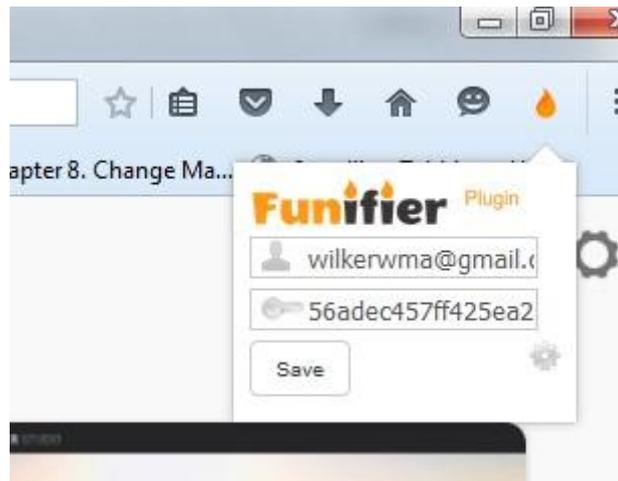


Figura 23 - Plugin instalado no Browser.

- Adicione o Api-Key da gamificação em questão, o identificador único da gamificação pode ser encontrado no menu “Details” do Funifier. Após a configuração do Plug-in todas as páginas do navegador foram integradas ao Funifier;

Após a configuração do Plug-in Funifier, vá até a página do Scratch e ative a opção “Inline”, que se encontra no canto superior direito do navegador, proceda da seguinte maneira para a aplicação da gamificação em todas as ações definidas no projeto de gamificação:

- Clique com o botão direito do mouse no elemento html da página Scratch no qual deseja gamificar;
- Selecione a opção desejada para inserção dos *listeners*, (seja eles actions, tour ou Widgets);
- No formulário de mapeamento, clique no botão “+” para adicionar uma ação;
- Selecione a ação que desejava atrelar a aquele elemento;
- Escolha o evento html que dispararia a ação, ex: Clique, visualização de página, like, assistir vídeo etc.

A Figura 24 demonstra a adição de ações com a ferramenta Inline do Funfier.

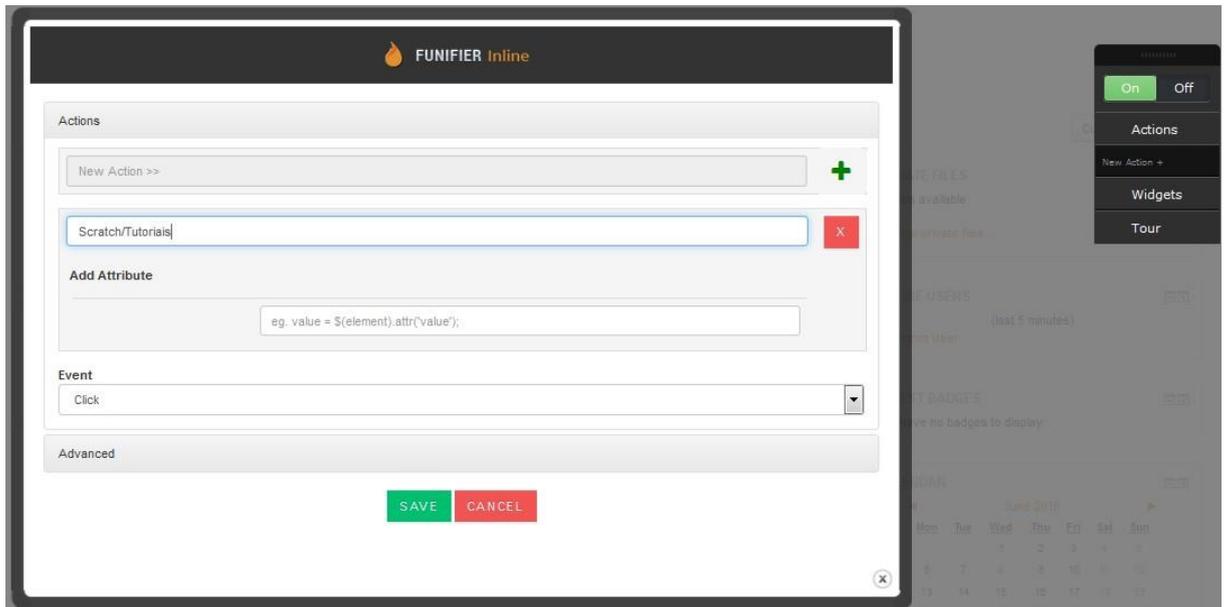


Figura 24 - Plugin Inline do Funfier.

4.3.5 Integração entre Moodle e Scratch

A integração do Moodle com Scratch não é realizada de forma direta, ou seja, não há configurações que permitam a troca de informações de forma automática entre as aplicações. A comunicação entre Moodle e Scratch acontece de forma indireta e por meio da ação dos estudantes, que submetem atividades realizadas dentro do Scratch, no Moodle, para conseguir completar um desafio.

Uma atividade é criada dentro do Moodle, contendo instruções de como o estudante deve proceder para concluí-la, instruções de como deve ser o algoritmo. A atividade redireciona o estudante para o ambiente do Scratch, no qual o mesmo confecciona seu algoritmo, após concluir as requisições o estudante deve capturar uma imagem da tela do algoritmo e submete-la no Moodle como prova de sua conclusão, fazendo assim uma ponte entre as duas ferramentas.

5. FLUXO BÁSICO DO AMBIENTE DE APRENDIZADO GAMIFICADO

O fluxo básico do ambiente de aprendizado consiste no caminho percorrido pelo jogador para desempenhar suas tarefas se comportando da maneira que se espera que ele se comporte. O fluxo básico é um referencial para verificar o funcionamento comum da aplicação e não comporta todas as ações possíveis que

podem ser desempenhadas dentro da mesma, serve ainda para ter-se um retorno avaliativo mais imediato que testes mais incisivos e que cobrem com mais robustez os pormenores das exceções dentro da aplicação.

A seção 5.1 apresenta o fluxograma da aplicação e a seção 5.2 apresenta o detalhamento de algumas atividades desse fluxograma de maneira que o mesmo foi exercitado e testado.

5.1 FLUXOGRAMA DO CAMINHO EFETUADO

A figura Figura 25 apresenta o fluxograma básico do caminho efetuado dentro da aplicação. As atividades acontecem em dois softwares que compõe o ambiente no contexto do trabalho. O estudante deve efetuar o login nos dois ambientes e prosseguir com as atividades, ressaltando que um estudante só poderá submeter atividades no Moodle se estiver inscrito no curso do Mundo do Grande Algoritmo. O ambiente do Scratch é dependente do Moodle, dentro do contexto do trabalho, e suas atividades são secundárias. O fluxo básico se repete semanalmente, com adição de atividades variadas e novos elementos de dificuldade e gamificação.

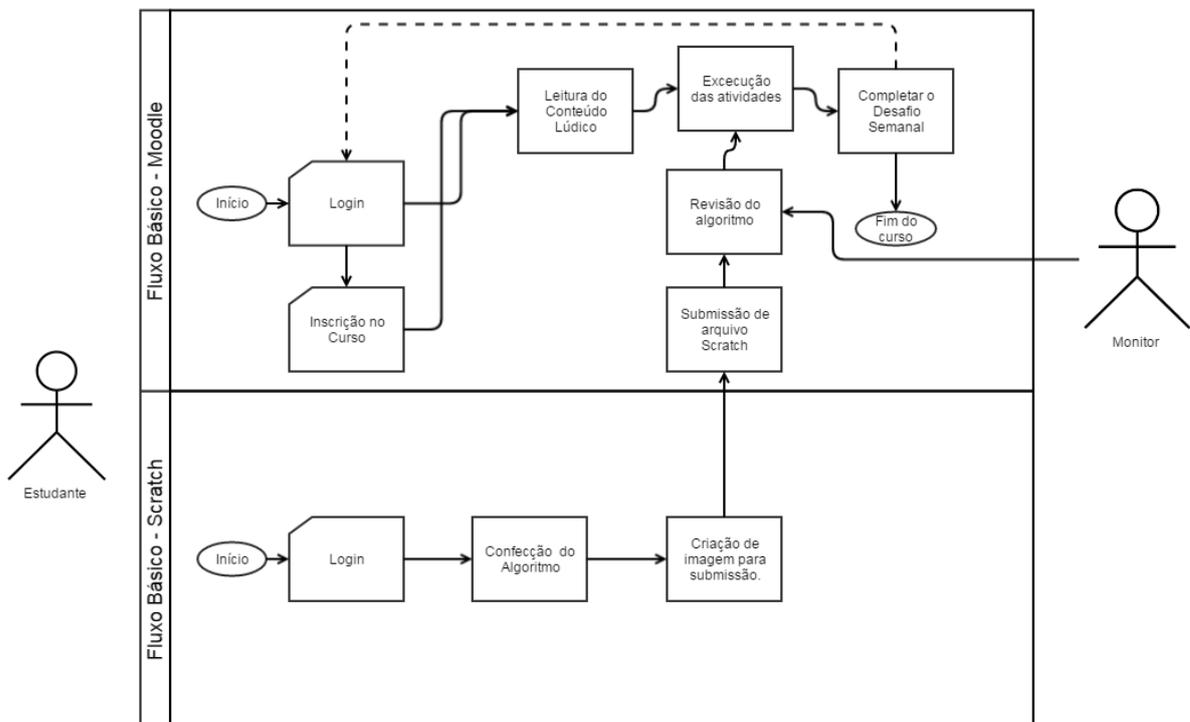


Figura 25- Fluxo Básico do ambiente.

A secção a seguir explora alguns aspectos pontuais do fluxo básico do ambiente, tal como apresente possibilidades que não são apresentadas de forma direta na Figura 25.

5.2 DETALHAMENTO DAS PRINCIPAIS ATIVIDADES DO FLUXO

Com todas as preparações iniciais do ambiente feitas e com o jogador possuindo acesso ininterrupto a internet as ações a seguir foram realizadas com finalidade de testar o fluxo básico da aplicação.

5.2.1 Login

Ao acessar a página inicial do ambiente o usuário deve entrar no sistema, colocando suas credenciais de nome de usuário e senha ou acessar o ambiente utilizando as suas credenciais do Facebook (Processo automatizado) clicando em “Fazer login com Facebook”. A questão do Login com o Facebook tem impacto em uma das técnicas de gamificação, o Facebook foi escolhido por ser a rede social mais utilizada. Para trabalhos futuros novas integrações com outras redes sociais podem ser implementadas.

The image shows a login interface with the following elements:

- Title:** "Acessar" in a large, bold, dark blue font.
- Form Fields:**
 - "Identificação de usuário" with a text input containing "admin".
 - "Senha" with a password input showing masked characters (dots).
- Checkboxes:** A checked checkbox labeled "Lembrar identificação de usuário".
- Buttons:**
 - A grey button labeled "Acessar".
 - A blue button with the Facebook logo and text "Fazer login com Facebook".
 - A grey button labeled "Acessar como visitante" at the bottom.
- Links:** A link "Esqueceu o seu usuário ou senha?" in orange text.
- Footer:** A grey bar with the text "O uso de Cookies deve ser permitido no seu navegador" and a help icon.

Figura 26 - Tela de login.

5.2.2 Inscrição no curso

Após acessar o sistema por meio de login o jogador deverá se cadastrar no curso denominado “Mundo do Grande Algoritmo”, que consiste no ambiente lúdico de aprendizado, o curso está presente na página principal do Moodle e aparece para todos os usuários autenticados na aplicação, a ação pode ser executada clicando no curso e selecionando a opção “Inscreva-me”.



Figura 27 - Curso da gamificação.

Quando o jogador se inscreve no curso ele passa a ter acesso a página principal do curso, nesta página ele encontrará as informações do seu perfil de gamificação (No lado esquerdo da tela), informações dos fóruns, atividades, eventos (todos localizados no lado direito da tela) e também das tarefas presente dentro do curso (Essas tarefas estarão centralizadas na tela). No momento inicial do curso a única seção que estará disponível é a “Início da Jornada”, mas com o passar do tempo as outras seções serão habilitadas.

Mundo do Grande Algoritmo



Painel ► MGA

FUNIFIER PROFILE

teste@gmail.com

606 pts

Novato 0/11

60.6%

394 points to next level

Challenges In Progress

Início da jornada

O mundo do grande algoritmo é um local fantástico onde as pessoas possuem a capacidade de conjurar(Escrever) formas poderosas de magia, os algoritmos. Neste mundo tudo é centralizado nos segredos dos algoritmos e quanto mais as pessoas tem conhecimento nestes algoritmos, maior o seu posto na sociedade.

O objetivo dos moradores do mundo do grande algoritmo é adquirir conhecimento e evoluir em sociedade, a ponto de compor o Alto conselho ou até mesmo se tornar o "Grande Rei". Podendo assim ajudar os outros moradores e guiar o reino com seu poder e conhecimento.

Quanto maiores seu conhecimento e suas contribuições maior o seu posto dentro do mundo e maiores são suas responsabilidades para com o grande algoritmo.

PESQUISAR NOS FÓRUNS

Vai

Pesquisa Avançada ?

ÚLTIMAS NOTÍCIAS

Conversas

12 abr, 07:36 Admin User

test

1 fev, 00:27 Admin User

Tópicos antigos ...

PRÓXIMOS EVENTOS

Não há nenhum evento próximo

Figura 28 - Primeira vista do curso.

As seções de conteúdo são divididas basicamente em: texto introdutório, imagem representativa, lista de atividades/ações que podem ser desempenhadas na semana e desafio. Cada atividade tem sua pontuação específica e conteúdo que condiz com a seção atual de conteúdo. Na primeira seção o estudante poderá se engajar nas seguintes atividades.

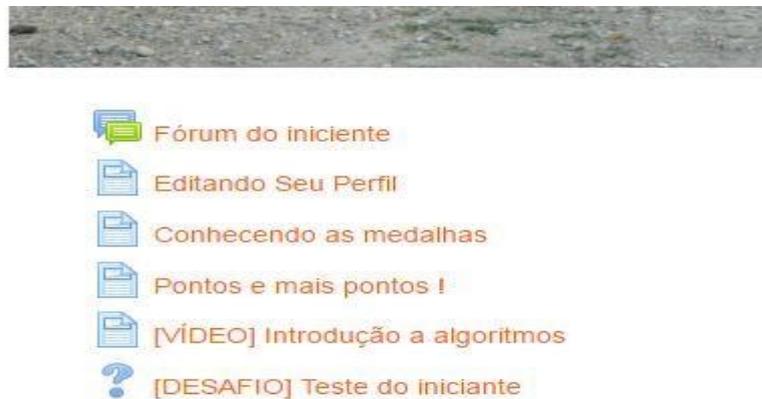


Figura 29 - Lista de atividades

5.2.3 Primeiras atividades

Ao escolher, por exemplo a atividade “Conhecendo as medalhas”, o jogador terá ao conteúdo da mesma e ganhará o pontos e medalhas referentes à essa atividade.

Conhecendo as medalhas

As medalhas são demonstrações de seus feitos dentro do Mundo de Grande algoritmo, além de mostrar como você está se saindo você pode se divertir colecionando-as, afinal um grande aventureiro além de histórias para contar também tem troféus para mostrar. Você pode receber medalhas por coisas simples, como completar uma atividade ou assistir um vídeo. Você também pode receber medalhas por feitos mais grandiosos, como resolver um algoritmo utilizando conceitos avançados de estruturas de repetição ou completando todos os desafios do mundo do Grande Algoritmo (Essa de fato é uma medalha grandiosa).

Calma, calma ... Cada medalha no seu tempo !



Figura 30 - Atividade "Conhecendo as medalhas".

Vale ressaltar que existem estruturas de atividades diversas dentro do ambiente do Moodle e que cada uma delas é apresentada de forma diferente ao usuário. A exemplo a atividade “[VÍDEO] Introdução a Algoritmos”, ao clicar na atividade o jogador terá acesso direto a um vídeo incorporado dentro da estrutura da página no Moodle, podendo assim assistir o conteúdo sem sair do ambiente.

[VÍDEO] Introdução a algoritmos

O que são algoritmos e quais poderes eles nos dão? Bom, se você não tem tanta paciência para ler poderá acompanhar vídeos didáticos.

Vale lembrar que o professor que apresenta o vídeo é de outra dimensão e não faz parte do Mundo do Grande algoritmo. É importante que você possa abstrair os conhecimentos para o seu treinamento.

Créditos: Professor Gustavo Guanabara



Figura 31 - Atividade "[VÍDEO] Introdução a algoritmos".

5.2.4 Desafios

Após completar todas as atividades da seção o jogador terá que enfrentar o desafio à ser completado. O desafio estará presente ao final de todas as seções para avaliar o conteúdo aprendido pelo jogador, a dificuldade dos desafios está acima da dificuldade das atividades comuns, cada desafio pode conter questões discursivas,

questões múltipla escolha, questões verdadeiras ou falsas e questões de submissão de arquivos. Ao clicar na atividade “[DESAFIO] Teste do iniciante”.

[DESAFIO] Teste do Iniciante

Esta lição corresponde a 0 pontos. Você recebeu 0 ponto(s) de um total de 0 pontos até agora.

Você deve nos dizer o que é um algoritmo, parece simples ? É isso mesmo, devemos começar por baixo. Você logo verá que grandes feitos serão realizados através dos algoritmos !

- É uma lista finita de atividades que ajudam a solucionar problemas
- É um tipo obscuro de mágica
- É um grupo de armas que servem para combater o mal.
- É uma lista de afazeres que escrevemos para conseguirmos nos lembrar no dia a dia.

Enviar

Você completou 0% da lição

Figura 32 - Desafio "Teste do Iniciante".

5.2.5 Atividades do Scratch

Após a ambientação da primeira semana de curso o jogador já terá passado um tempo considerável exposto ao conteúdo de algoritmos e chegará o momento em que ele explorará novas atividades. As novas atividades consistem em construção de algoritmos por meio do software Scratch. O jogador acessará por meio da atividade o site do Scratch, procederá com a instalação do plug-in e criação da conta no ambiente.

[SCRATCH] Primeiro desafio - Invocando o gatinho !

Uma jornada sempre é mais fácil quando se tem amigos, o mascote do Scratch irá lhe ajudar a adquirir novos conhecimentos de algoritmos. Subme

A atividade deve conter uma imagem que possua a tela do algoritmo no Scratch, contendo ao menos:

- Início do algoritmo
- 3 passos para resolver um problema do seu dia a dia
- Fim do algoritmo.

Boa sorte !

Submission status

Submission status	No attempt
Grading status	Not graded

Figura 33 - Atividade Scratch.

Com a conta criada o jogador poderá visualizar o seu progresso também no seu perfil Scratch. Ao clicar na opção “Criar”, presente na barra de menu superior do Scratch, o jogador será redirecionado para o ambiente de criação Scratch e poderá assim completar as atividades, no caso, atividades presentes no ambiente Moodle.

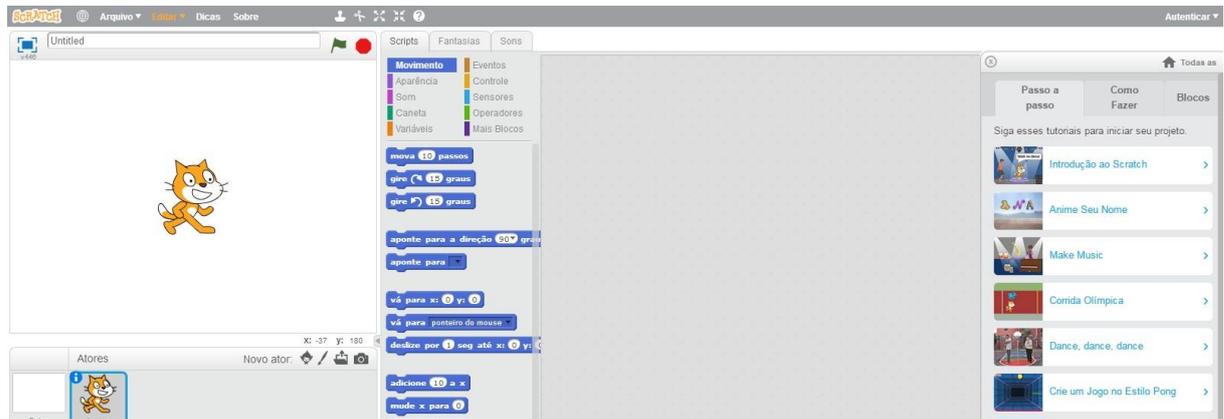


Figura 34 - Ambiente de criação de algoritmos no Scratch

Ao concluir o seu algoritmo no Scratch o jogador deverá submeter uma imagem do seu trabalho no ambiente do Moodle para avaliação do supervisor.

5.2.6 Escolha de classe

Com a aquisição de pontos o suficiente, os jogadores começaram a progredir nos níveis estipulados no jogo. Dos níveis estipulados no projeto de gamificação os jogadores devem, em certo momento, escolher qual caminho querem seguir, a partir dos níveis intermediários o jogador deverá escolher entre: Cavaleiro, Arqueiro ou Mago. O jogador poderá realizar a ação clicando na atividade “Nome da atividade” e respondendo o “Quiz” dentro da mesma, após essa ação seu novo grupo será atribuído e ele poderá seguir adquirindo os níveis referentes a sua escolha.



Figura 35 - Escolha de caminhos.

Com a aquisição cotidiana de pontos é natural que surja uma disputa entre os jogadores mais competitivos, para saber quem tem a maior quantidade acumulada, para os estudantes mais competitivos há a classificação geral de pontuações, a mesma se encontra no lado esquerdo do ambiente juntamente com os outros *Widgets* Funifier do ambiente.

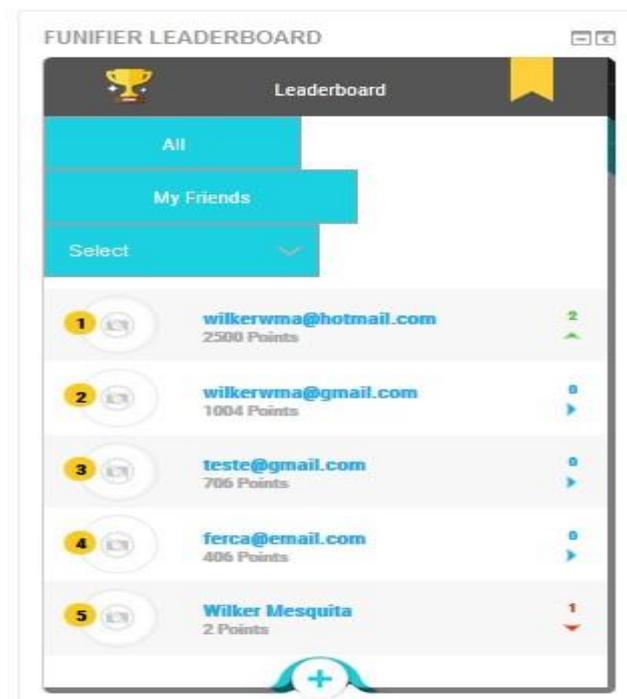


Figura 36 - Widget de Classificações.

A fluxo de atividades dos estudantes dentro do ambiente se repete semanalmente, a cada nova semana uma nova seção é habilitada e suas atividades disponibilizadas, por exemplo a seção que trata estruturas condicionais para algoritmos pode ser visualizada.

Agora que você já tem uma experiência significativa dentro do mundo do Grande Algoritmo, mais suas possibilidades de combate. Aprenda agora a controlar as estruturas condicionais



-  [VÍDEO] Estruturas Condicionais
-  [SCRATCH] Estruturas condicionais, faça o gatinho tomar decisões !
-  [HISTÓRIA] Estruturas condicionais
-  [Desafio] Estruturas Condicionais

Figura 37 - Seção de atividades condicionais.

5.2.7 Fim do curso

O procedimento continua o mesmo, ao clicar em uma atividade a mesma é iniciada e o jogador poderá adquirir o conhecimento tal qual os pontos e medalhas atreladas as atividades desempenhadas.

Ao fim da jornada dentro do ambiente do Mundo do Grande algoritmo os jogadores serão incentivados a continuarem no estudo de algoritmos, novos conteúdos serão apresentados e o curso será encerrado.

EXTRA: Siga seu próprio caminho

Agora que você domina o que precisa de algoritmo sua jornada está aberta.

Não se acomode, fora do mundo de grande algoritmo existe UM UNIVERSO DE ALGORITMOS !!! Mas isso não acaba por aí, além de algoritmos existem outros assuntos altamente divertidos para que você evolua cada vez mais !

Não há limite para o conhecimento.



Sites interessantes

Figura 38 - Seção EXTRA motivacional.

6. CONCLUSÃO

O processo de adaptação do ambiente virtual de aprendizado pôde ser realizado com base em um projeto bem especificado dentro das técnicas de engenharia de software, delineando suas etapas e utilizando ferramentas adequadas. O ambiente de aprendizado virtual Moodle corresponde as expectativas, tanto em sua gama de funcionalidades, capacidade de extensão das mesmas, e também, no correto comportamento de suas atribuições técnicas, sendo observado os elementos de ensino à distância juntamente com a gamificação.

A Gamificação foi aplicada no ambiente de ensino virtual de maneira sistêmica, resultando em um ambiente capaz de se comportar da maneira a prover um feedback ao usuário, tanto no que tange à técnicas como as do contexto lúdico, como técnicas de pontos e medalhas. Ferramentas gamificadas visam guiar o comportamento do aluno e despertar a sua motivação, mesmo sem incentivo presencial e direto do professor. Este trabalho não aferiu o contato e feedback do usuário com o ambiente virtual gamificado, ficando assim para execuções futuras, as questões relacionadas aos testes com usuários.

A utilização do software Scratch como ferramenta de aprendizado paralela faz com que as atividades dos estudantes não caiam em um ciclo repetitivo e traz uma novo paradigma no processo criativo dos algoritmos. A ferramenta Scratch não pôde ser totalmente integrada na gamificação e poderia entrar em trabalhos futuros. O processo de integração direta entre Scratch e Funifier foi realizado, obtendo-se o ambiente capaz de mapear as ações dos estudantes.

Tendo em vista projetos futuros, este trabalho abre precedentes para a customização da ferramenta em contextos mais específicos, com foco em grupos que necessitam de atenção especial em seu ensino e aprendizagem. Um exemplo mais claro é o grupo caracterizado pela Síndrome de Down, que necessita de abordagem e métodos diferenciados para melhor extrair os benefícios do ensino. A engenharia de *software* deve continuar trabalhando para a inserção e integração de todos os indivíduos da sociedade.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- BISSOTO, M. L. Aprendizado de portadores de Síndrome de Down. *Ciência e Cognição*, p. 80-88, 2005.
- BRASIL. Constituição da República Federativa do Brasil. Brasília: Senado Federal, 1988.
- BUCKLEY, S. J.; BIRD, G. Meeting the educational needs of children with Down syndrome, Portsmouth, 1994.
- BUCKLEY, S.; BIRD, G. Including children with Down syndrome. *Down Syndrome News and Update*, 1998.
- CAPONE, G. T. Down Syndrome genetic insights and thoughts on early intervention. 2. ed. [S.l.]: *Infants Young Child*, v. 1, 2004.
- CHOU, Y. Actionable Gamification: Beyond points, Badges, and Leaderboards. [S.l.]: Amazon, 2014.
- CODECADEMY. About. Disponível em: <<https://www.codecademy.com/pt/about>>. Acesso em: 08 nov. 2015.
- CORMEN, T. H. et al. Algoritmos: Teoria e prática. São Paulo: Editora Elsevier, 2002.
- CUCKLE, P. Getting in and staying there: children with Down syndrome in mainstream schools. *Downs Syndr Res Pract*, v. 6, n. 2, p. 95-99, 1999.
- DAMAS, L. Linguagem C. Rio de Janeiro: Editora LTC, 2007.
- FERREIRA, A. B. D. H. Dicionário Aurélio da Língua Portuguesa. Rio de Janeiro: Editora Positivo, 2010.
- FOUNDATION, E. OpenUP Unified Process. Disponível em: <<http://epf.eclipse.org/wikis/openup/>>. Acesso em: 22 out. 2015.
- GAYCHO, L.; GUNN, P.; SIEGAL, M. Couting by children with Down syndrome, 1991.
- GURALNICK, M. J. Involvement with peers: comparisons between young children with and without Down's syndrome. *J Intellect Disabil Res.*, v. 46, n. 5, p. 379-393, 2002.
- HOLDEN, B.; STEWART, P. The inclusion of students with Down syndrome in New Zealand schools. *Down Syndrome News and Update*, v. 2, n. 1, p. 24-28, 2002.
- INSTITUTE, P. M. Um Guia do Conhecimento em Gerenciamento de Projetos. [S.l.]: Editora Saraiva, 2014.
- KHAN, S. Um mundo, uma escola. Rio de Janeiro: Editora Intrínseca, 2013.
- LORENZ, S. Making inclusion work for children with Down syndrome. *Down Syndrome News and Update*, v. 1, n. 4, p. 175-180, 1999.

- LUIZ, R. F. M. et al. A inclusão da criança com Síndrome de Down na rede regular de ensino: desafios e possibilidades. *Revista Brasileira de Educação Especial*, Marília, v. 14, n. 3, p. 497-508, Setembro 2008.
- MICROSOFT. N-Tier Data application Overview. Disponível em: <<https://msdn.microsoft.com/en-us/library/bb384398.aspx>>. Acesso em: 23 out. 2015.
- MOODLE. Moodle Architecture. Disponível em: <https://docs.moodle.org/dev/Moodle_architecture>. Acesso em: 23 out. 2015.
- PUESCHEL, S. M. Síndrome de Down: guia para pais e educadores. Campinas: Editora Papirus, 1999.
- RESNICK, M. et, al. *Scratch: Programming for All*. AMC, 2009.
- SCHWARTZMAN, J. S. E. A. Síndrome de Down. São Paulo: Ed. Memnon, 1999.
- SILVA, M. D. F. M. C.; KLEINHANS, A. C. D. S. Processos cognitivos e plasticidade cerebral na Síndrome de Down. *Revista Brasileira de educação especial*, v. 12, n. 1, p. 123-138, janeiro 2006.
- UMPHRED, D. A. *Reabilitação Neurológica*. 4ª. ed. São Paulo: [s.n.], 2004.
- WING, J. M. Computational thinking. *Communications of ACM*, v. 39, n. 3, p. 33-35, 2006.

APENDICE A – Apanhado sobre Síndrome de Down

A.1 Aspectos biológicos

A Síndrome de Down é uma condição genética caracterizada pela desordem do material genético presente no cromossomo de par 21, tornando a posição 21 um trio de cromossomos, sendo também denominada como trissomia do cromossomo 21. Indivíduos portadores de Síndrome de Down apresentam diferenças físicas e psicológicas, quando comparados a pessoas com cromossomos normais. O fator de risco preponderante é a idade materna avançada. (CAPONE, 2004; PUESCHEL, 1999 apud LUIZ et al., 2008)

A Trissomia do cromossomo 21 pode ser diagnosticada após poucas horas de vida, por características fenotípicas e posteriormente por análises cito genéticas do recém-nascido (SCHWARTZMAN, 1999).

O portador de síndrome de Down pode se ramificar em 3 diferentes tipos, a trissomia livre (com 95% dos indivíduos com o diagnóstico), mosaïcismo e translocação (5% dos indivíduos com o diagnóstico) (UMPHRED, 2004). A trissomia simples consiste na existência de um cromossomo extra no par 21, o mosaïcismo consiste em diversificação celular, onde o indivíduo pode ou não apresentar o cromossomo extra em suas células, a translocação é caracterizada pela translocação de parte do cromossomo 21 para outro cromossomo.

Schwartzman e Pueschel (1999 apud Luiz et al., 2008) descrevem características que podem aparecer em um portador de síndrome de Down, são elas: baixa estatura, face estreita e achatada, olhos com inclinação lateral e afastados um do outro, pálpebras estreitas e levemente oblíquas, cabelo liso e fino com possibilidade de falhas, orelhas pequenas e com implantação craniana baixa, boca pequena, língua projetada para fora da boca, nariz pequeno com base nasal achatada, palato ogival, pés, assim como as mãos, são pequenos e grossos, cardiopatia congênita, excesso de pele na nuca, genitais hipodesenvolvidos.

A.2 Aspectos do aprendizado

Buckley e Bird (BUCKLEY e BIRD, 1994) levantaram algumas características de crianças portadoras de Down no que tange o desenvolvimento linguístico e

cognitivo, elas são: atraso no reconhecimento de regras gramaticais e sintáticas da língua e conseqüentemente comprometimento no desenvolvimento na linguagem, o que pode causar uma dificuldade de se expressar à medida que aprendem um novo conteúdo, essas dificuldades podem comprometer ainda o desenvolvimento de outras capacidades cognitivas a medida que os portadores encontram dificuldades em empregar os recursos da linguagem para raciocinar e relembrar informações. A dificuldade encontrada pelos portadores de síndrome de Down às levam a serem subestimadas por quem as ensina.

Estudos de Bower e Hayes (1994 apud Bissoto, 2005) apontam que a memória auditiva de curto prazo de crianças portadoras de Síndrome de Down é mais breve, dificultando o acompanhamento de ordens oralizadas, principalmente aquelas que se prolongam em demasia e consistem de inúmeras instruções consecutivas. Em contraponto às dificuldades com a memória auditiva, a memória visual dos portadores é mais desenvolvida, auxiliando assim seu aprendizado com formas de ensino voltadas à expressão visual da linguagem.

Tendo em vista as dificuldades dos portadores de síndrome de Down e colocando o foco do estudo em seu aprendizado, surge a necessidade de se focar nos aspectos que mais os auxiliariam. Restringindo as instruções faladas e segmentando-as, tal como inserindo uma grande carga de auxílio visual, a exemplo: linguagens aumentativas, figuras, animações. Utilizando dessa estratégia embasada nos estudos à cima referidos espera-se aumentar a chance de sucesso no que diz respeito à inclusão e ensino de crianças com síndrome de Down.

Caycho e Colaboradores (apud BISSOTO, 2005) apontam que: “as dificuldades no desenvolvimento de raciocínio matemático dos portadores está mais ligada a fatores culturais do que com as próprias dificuldades decorrentes da síndrome”, este fato está ligado ao preconceito de pessoas que circundam esses indivíduos, na grande maioria das vezes um preconceito cultural, os fazendo restringir o acesso do portador a esse conhecimento, achando que o mesmo não terá capacidade ou então terá uma capacidade limitada, quando se tratando de pensamento matemático.

Entretanto, vários estudos contrabalançam essa tendência, apontando que o desenvolvimento do indivíduo portador de Síndrome de Down é, tanto quanto o de qualquer não portador, resultante de influências sociais, culturais e genéticas; incluindo-se aí as expectativas havid

em relação às suas potencialidades e capacidades e os aspectos afetivo-emocionais da aprendizagem. Deve-se então observar que, muito embora portadores de Síndrome de Down apresentem características peculiares de desenvolvimento, isso não se constitui numa uniformidade a predizer comportamentos e potencialidades. (BISSOTO, 2005)

A.3 Aspectos educacionais

A Constituição federal de 1988 (BRASIL, 1988) assegura todos os direitos aos portadores de SD, inclusive de inserção social e educacional com ressalva às suas necessidades especiais, tal como qualquer outro cidadão brasileiro.

Segundo Luiz et al. (2008) a educação de crianças portadoras de SD é normalmente realizada em salas de aula em instituições como a APAE, Escolas regulares e escolas de ensino especial, todas essas podem de alguma forma instruir e integrar as crianças portadoras, valendo a ressalva de que as crianças devem preferencialmente receber atendimento educacional em escolas regulares de ensino. Ao contrário do que muitos pensam a maior dificuldade da inclusão de crianças com Síndrome de Down está nos: pais, professores e escolas.

A.3.1 Estudos

Em estudo realizado por Buckley e Bird (1998, apud Luiz et al. 2008) foi constatado que a inserção de crianças com SD na rede regular de ensino traz melhoras significativas, mesmo quando se tratava de crianças com maior dificuldade de aprendizado, a inserção em ensino regular trazia ganhos para a criança no que se refere ao engajamento em um grupo social. Essa melhora pode variar dependendo de condições diferentes de se lidar com essas crianças, em contextos diferentes de inclusão há caminhos diferentes para o sucesso.

Já em outro estudo realizado por Buckley et al. (2006, apud Luiz et al.) quando se trata de adolescentes, o estudo realizado no reino unido verificou que não houve diferença significativa entre os grupos de escolas regulares e escolar especiais no que se refere à capacidade de independência, contato social, atividades de lazer e inclusão na comunidade, contudo o mesmo estudo apontou que a inclusão trouxe melhora na linguagem, comportamento e aspectos acadêmicos.

Alguns fatores identificados para inclusão bem-sucedida de uma criança SD no ensino regular (HOLDEN e STEWART, 2002). Mostrando assim que o sucesso da

criança está diretamente ligado à instituição de ensino e professores responsáveis, as características são:

- Ambiente estruturado e adaptado às suas necessidades;
- Abordagem de ensino que facilite sua aprendizagem e adaptação curricular.

A.3.2 Os pais

São os pais que escolhem se vão introduzir as crianças numa escolar regular, ou não aspectos sociais, biológicos e pessoais influenciam diretamente nessa escolha. Há casos em que crianças em que não acompanham o ritmo da turma ou ficam mais de um ano no mesmo nível, o que acaba frustrando os pais, que acabam por transferir os alunos para escolas especiais a procura de uma melhora no aprendizado e satisfação de seu filho (CUCKLE, 1999).

Levando em consideração o bem-estar familiar e a percepção dos pais em relação ao aprendizado de seus filhos, deve-se oferecer um *feedback* positivo e o máximo de apoio para a progressão da criança, o ambiente almeja sanar tais falhas inserindo tecnologia refinada e direcionada a esse público para que eles possam aprender de uma nova forma.

Os pais têm a capacidade de ensinar seus próprios filhos tanto academicamente quanto socialmente trazendo ganhas significativos para o contexto de inclusão, já que a criança também aprende por meio da observação dos adultos mais próximos (BUCKLEY; BIRD, 1998; apud Luiz et al., 2008). Pesquisa (GURALNICK, 2002; apud LUIZ et al., 2008) realizada por questionários e entrevistas comparou mães de portadores de SD e mães de crianças não portadoras que apresentavam atraso de desenvolvimento. Crianças com SD obtiveram melhores resultados, quando comparados com as outras crianças no que diz respeito ao aprendizado, à variedade de atividades oferecidas à criança e ao seu preparo par ao mundo real. Guralnick sugere ainda que a participação de crianças no cenário inclusivo seja encorajada.

A.3.3 Os professores

Estudo de Lorenz (1999), sugere que, para dar apoio ao estudante com SD, o professor deve: oferecer a ele possibilidades de acesso ao currículo normal, encorajar a sua independência, promover e incentivar o trabalho com outros colegas de classe.

Um dos objetivos da educação inclusiva é aumentar a independência dos alunos. Pais e professores devem estar vigilantes para que o excesso de assistência cause dependência dos alunos na escola (LORENZ, 1999), com uma visão não passiva do ambiente de aprendizado virtual no qual a criança é a maior responsável por seu aprendizado, as questões de dependências podem ser sanadas sem grandes dificuldades sempre com ressalvas para o acompanhamento e atenção que são necessários no que diz respeito à portadores de Síndrome de Down.

Os professores não sabem ao certo se a escola regular é a melhor para alunos com SD, pois acreditam que com o passar do tempo as diferenças acadêmicas entre elas e as crianças normais podem aumentar. Sabendo-se que o desenvolvimento cognitivo da criança decorre da sua interação com o ambiente, faz se necessária uma estimulação bem estruturada para minimizar as dificuldades e evidenciando as possibilidades de plasticidade cerebral (SILVA e KLEINHANS, 2006).

A.3.4 A escola

A proporção de crianças com síndrome de Down em escolas regulares tem aumentado, se for comparada às escolas especiais. Mais crianças estão entrando na rede regular de ensino e mais crianças também estão permanecendo na rede de ensino (CUCKLE, 1999). Mesmo crianças que apresentam dificuldades maiores níveis de dificuldade, apresentam melhoras quando inseridas em escolas regulares, respeitando sempre a pluralidade dos portadores (BUCKLEY; BIRD, 1998; apud Luiz et al., 2008).

Para se obter sucesso na inclusão, o ambiente escolar deve oferecer condições favoráveis ao portador, com abordagem de ensino que atenda as demandas dos portadores de síndrome de Down e também de portadores de outras individualidades. Os profissionais devem estar capacitados para que se haja um ensino de qualidade para essas crianças (HOLDERN e STEWART, 2002)

APENDICE B – Instalação do Moodle

Para a realização dos procedimentos presentes no trabalho uma instância do Software Moodle na versão 3.0.3 foi configurada e instalada localmente (Na máquina do estudante), os passos para essa atividade são descritos a seguir.

Requisitos Mínimos:

- Apache;
- PostgreSQL, MySQL ou MariaDB;
- PHP.

Requisitos de Hardware: espaço em disco de 200MB para o armazenamento do código do Software e por recomendação 5GB de espaço para armazenamento de conteúdo;

- Processador: 1GHz(Mínimo), 2GHz ou superior (Recomendado);
- Memória: 512MB(Mínimo), 1GB ou superior (Recomendado).

Os requisitos mínimos e recomendados do sistema podem variar de acordo com as necessidades do provedor, à exemplo de uma instância que possui um grande fluxo de dados passando pela aplicação.

Para configuração de servidor o software WAMP foi instalado e utilizado. O software WAMP consiste em uma ferramenta de integração de elementos para gerenciamento de sites baseados em PHP, o mesmo engloba uma lista de softwares que centralizados no WAMP cobrem todos os requisitos necessários para se manter um solução de Software PHP no ar. A versão do WAMP utilizada é:

WampServer3_x64. O WAMP tem em sua composição os seguintes softwares:

- Apache : 2.4.17
- MySQL : 5.7.9
- PHP : 5.6.16
- PHP : 7

Há duas maneiras de se obter os arquivos do Moodle, a primeira consiste no Download da versão desejada, a mesma se encontra no link: <http://moodle.org/downloads>, vale a ressalva de que é uma versão com

arquivos compactados. A segunda opção é puxar o código do repositório Git, hospedado no site GitHub, por meio do comando `Git Push <nome_do_repositório>`.

A maneira utilizada foi o Download do arquivo pelo link moodle.org. Os arquivos devem ser colocados dentro do diretório WWW do servidor, normalmente criadas na instalação. Como a plataforma de desenvolvimento Web WampServer foi utilizada, a localização do código moodle se encontra na pasta: `caminho_do_usuario/wamp/www`.

Uma base de dados vazia deve ser criada para o armazenamento dos dados do Moodle, esta base deve contar as seguintes informações.

- `dbhost`: o host para o moodle, ex: `localhost`, `www.nomedosite.com`;
- `dbname`: Um nome para a base de dados, ex: `Moodle`;
- `dbuser`: Um usuário para a base de dados, não é recomendável utilizar o usuário padrão `root` ou `superusuário`;
- `dbpass`: A senha de acesso à base.

Criar um diretório chamado `moodledata`, que irá armazenar informações referentes à uploads do site, dados de cache, dados temporários, dados de sessão etc. É importante que esse diretório esteja fora do diretório do Moodle e que o mesmo não seja acessado de fora do seu servidor, políticas de privacidade são altamente recomendadas.

Acesse o instalador web do moodle via browser no endereço do seu servidor (`seuhost/moodle`) e proceda com a instalação, provendo o caminho para os diretórios e os respectivos dados da base. Após a instalação o Moodle deverá estar acessível por meio de Web Browser no caminho: `ip_do_servidor/moodle`.

APÊNDICE C – Lista de técnicas Octalysis

- Pontos: São elementos dados aos jogadores com intuito de incentivar a continuidade de suas atividades, funcionam também como incentivo ao completar uma atividade específica, que ganha um sentido extra;
- Medalhas: São prêmios dados aos jogadores com intuito de reconhecer seu progresso e ações dentro do ambiente, funcionam também como itens colecionáveis;
- Rankings: São elementos que organizam os jogadores de acordo com algum critério, incentiva a competição entre os jogadores, tornando-os assim, mais propensos à concluírem suas atividades.
- Avatar: É o perfil do jogador dentro do ambiente, o toma sua face dentro do jogo e o faz se sentir proprietário de uma personalidade.
- Sorte de Principiante: São as ações de incentivar os jogadores iniciantes fazendo-os crer que são especiais, escolhidos pelo acaso, pela sorte;
- Curva de aprendizado: Controla os níveis de dificuldade das atividades executadas, com intuito de manter um fluxo de motivação no ambiente. Não permita que as atividades sejam sempre muito fáceis ou que sejam muito difíceis;
- Narrativa: Transforma o ambiente gamificado em uma história contada através do interlocutor ao jogador, transportando-o para uma fantasia/contexto novo e colocando um novo teor em suas atividades;
- Dinâmica de compromisso: Faz com que o jogador precise estar em determinado horário em determinado local para resolver alguma atividade,
- Coleção: Faz com que o jogador tenha motivos para continuar procurando itens colecionáveis para completar sua coleção;
- Sinfonia do aumento de Nível: Faz com que o jogador percebe de forma suave a sua progressão dentro do ambiente, sabendo que sempre está dando um passo à frente em seus objetivos;
- Orientações: Traz conforto ao estudante, mostrando que há alguém que poderá ajudá-lo caso precise, também serve como incentivador para pessoas que se motivam com o fato de poder instruir e ajudar alguém que precise;

- Amizades: Faz com que o jogador se sinta conectado à alguém, motivando-o a manter-se ativo no ambiente no qual ele possui laços;
- Escolhas significativas: Traz significância as escolhas do jogador, no sentido de atitudes que geram ações que tem impacto duradouro em sua jornada dentro da gamificação;