

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**RENATO ANTONIO DE OLIVEIRA FOGAÇA**

**DESENVOLVIMENTO DE UM SOFTWARE EDUCACIONAL  
GAMIFICADO PARA PLATAFORMA WEB COM FERRAMENTA DE  
AUTORIA DE QUESTÕES**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2018**

**RENATO ANTONIO DE OLIVEIRA FOGAÇA**

**DESENVOLVIMENTO DE UM SOFTWARE EDUCACIONAL  
GAMIFICADO PARA PLATAFORMA WEB COM FERRAMENTA DE  
AUTORIA DE QUESTÕES**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr<sup>a</sup> Monica Hoeldtke Pietruchinski

**PONTA GROSSA**

**2018**



---

## TERMO DE APROVAÇÃO

### DESENVOLVIMENTO DE UM SOFTWARE EDUCACIONAL GAMIFICADO PARA PLATAFORMA WEB COM FERRAMENTA DE AUTORIA DE QUESTÕES

por

RENATO ANTONIO DE OLIVEIRA FOGAÇA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 07 de Novembro de 2018 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr<sup>a</sup> Monica Hoeldtke Pietruchinski  
Orientador(a)

---

Prof. Dr. Diego Roberto Antunes  
Membro titular

---

Prof. Msc. Rogério Ranthum  
Membro titular

---

Prof(a). Dra. Helyane Bronoski Borges  
Responsável pelo Trabalho de Conclusão  
de Curso

---

Prof. Dr. André Pinz Borges  
Coordenador do curso

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço à Deus por permitir estar vivo e cumprindo minha missão aqui na Terra, agradeço ao Dr. Fogaça, mais conhecido como meu pai e a Dr. Monica, mais conhecida como minha mãe, sem ambos eu não teria tanto êxito nessa caminhada, muito obrigado por todos os ensinamentos, suporte e apoio até agora e sempre.

Agradeço à minha orientadora Prof. Dr<sup>a</sup> Monica Hoeldtke Pietruchinski, pela sabedoria com que me guiou nesta trajetória, pelas exigências, críticas e paciência em toda essa jornada.

Agradeço as minhas 3 companheiras de morada, outrora minhas irmãs, Luiza, Ligia e Isis, por exercitar minha paciência e persistência.

Gostaria de deixar registrado também, o meu reconhecimento e enorme gratidão à minha noiva, Laiz, pois acredito que o apoio e suporte dela foram fundamentais para concluir esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa e a instituição que me acolheu nesses anos, meu muito obrigado.

Faça. Ou não faça. Não existe a tentativa.  
(YODA Mestre, Star Wars V)

## RESUMO

FOGAÇA, Renato Antonio de Oliveira. **Desenvolvimento de um sistema web educacional Gamificado com Ferramenta de Autoria**. 2018. 59 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

Neste trabalho de conclusão de curso é apresentado o desenvolvimento de um software educacional para plataforma web possuindo uma ferramenta de autoria, aplicam-se os critérios da gamificação relacionados a personalização de perfil, geração de *ranking* e *feedback* aos usuários. O foco do projeto foi aplicar vários conceitos aprendidos no decorrer da graduação juntamente com aqueles aprendidos durante a fase de pesquisas do projeto no que dizem respeito aos softwares educacionais, ambientes gamificados, gamificação e ferramenta de autoria. Durante o desenvolvimento foi possível compreender como aplicar a gamificação e como uma ferramenta de autoria pode auxiliar na elaboração de questões bem como geração de provas didáticas na modalidade escrita, apresenta-se também a modelagem do banco de dados do software. Por fim exibe-se a metodologia utilizada no processo de desenvolvimento do trabalho, os resultados encontrados e as dificuldades observadas após a aplicação dos conceitos e implementação do *software*.

**Palavras-chave:** Gamificação. Ferramenta de Autoria. Educacional.

## ABSTRACT

FOGAÇA, Renato Antonio de Oliveira. **Development of an educational Web system Gamificado with Tool of Autoria**. 2018. 59 f. Course Conclusion Work (Technology in Analysis and Development of Systems) - Federal University of Technology - Paraná. Ponta Grossa, 2018.

In this work of conclusion of course is presented the development of an educational software for web platform having a tool of authorship, apply the criteria of gamification related to profile personalization, generation of ranking and feedback to users. The focus of the project was to apply several concepts learned in the time of graduation along with those obtained during the research phase of the project in regard to educational software, gamification environments, gamification and authoring tool. During development it was possible to understand how to apply the gamification and as a tool of authorship can help in the elaboration of questions as well as generation of didactic tests in the written modality, also presents the modeling of the database of the software. Finally, the methodology used in the work development process, the results found and the difficulties observed after the application of the concepts and software implementation are presented.

**Keywords:** Gamification. Authoring Tool. Educational.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Distinção dos termos .....	21
Figura 2 – Ciclo de Vida da Metodologia Scrum .....	25
Figura 3 – Modelo Relacional do Banco de Dados do Software .....	37
Figura 4 – Diagrama de Componentes .....	39
Figura 5 – Telas de <i>Login</i> .....	41
Figura 6 – Painel do Professor .....	42
Figura 7 – Painel de Disciplinas .....	43
Figura 8 – Painel de Assuntos.....	44
Figura 9 – Painel de Questões .....	45
Figura 10 – Criação de uma questão usando <i>CKEditor</i> .....	46
Figura 11 – Exportação para Arquivo .doc .....	47
Figura 12 – Painel de Questionários .....	48
Figura 13 – Painel do Aluno sem Questionário .....	48
Figura 14 – Painel do Aluno com Questionário .....	49
Figura 15 – Página de Resposta do Questionário.....	50
Figura 16 – Painel de Ranking e Resultados do Aluno .....	51
Figura 17 – Página de Revisão do Questionário .....	52
Figura 18 – Painel de <i>Ranking</i> e Resultados do Aluno para Professor.....	53

## LISTA DE QUADROS

Quadro 1 – Relação de Requisitos do Projeto e Prioridades .....	32
Quadro 2 – Relação de Requisitos Funcionais e Prioridades .....	32
Quadro 3 – Estimativa das User Stories .....	36

## **LISTA DE SIGLAS**

CSS	Cascading Style Sheet
HTML	Hypertext Markup Text Language
PHP	Personal Home Page

## **LISTA DE ACRÔNIMOS**

TICs	Tecnologias da Informação e Comunicação
WYSIWYG	Expressão em inglês "what you see is what you get"

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 OBJETIVOS .....	14
1.1.1 Objetivo geral.....	14
1.1.2 Objetivos específicos .....	14
1.2 JUSTIFICATIVA .....	15
1.3 ESTRUTURA DO TRABALHO .....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>16</b>
2.1 SOFTWARE EDUCACIONAL .....	16
2.2 JOGOS EDUCACIONAIS .....	17
2.3 AMBIENTES GAMIFICADOS .....	18
2.4 GAMIFICAÇÃO .....	19
2.5 SOFTWARE EDUCACIONAL GAMIFICADO .....	21
2.6 FERRAMENTA DE AUTORIA .....	22
2.7 TECNOLOGIAS .....	24
2.7.1 Scrum .....	24
2.7.2 PHP .....	25
2.7.3 HTML .....	26
2.7.4 Cascading Style Sheet.....	27
2.7.5 JavaScript .....	27
2.8 FERRAMENTAS .....	28
2.8.1 Microsoft Visual Studio Code .....	28
2.8.2 XAMPP .....	28
2.8.3 Microsoft Visio.....	29
2.8.4 jQuery .....	29
2.8.5 CKEditor 4 .....	30
2.9 ANÁLISE DE REQUISITOS .....	31
<b>3 DESENVOLVIMENTO DO SOFTWARE E SEUS RESULTADOS</b> .....	<b>34</b>
3.1 ADAPTAÇÕES NA METODOLOGIA .....	34
3.2 <i>PRODUCT OWNER</i> .....	35
3.3 ESTIMATIVAS .....	35
3.4 DIAGRAMAS E MODELAGEM DO BANCO DE DADOS .....	36
3.5 PREPARAÇÃO DOS AMBIENTES.....	38
3.6 ARQUITETURA DO SOFTWARE.....	38
3.7 IMPLEMENTAÇÃO .....	39
3.8 ENTREGA DOS <i>SPRINTS</i> .....	40
3.8.1 Login .....	40
3.8.2 Professor .....	41
3.8.3 Disciplinas e Assuntos .....	42

3.8.4 Questões e Questionários .....	44
3.8.5 Aluno.....	48
<b>4 CONSIDERAÇÕES FINAIS .....</b>	<b>54</b>
4.1 TRABALHOS FUTUROS .....	55
<b>REFERÊNCIAS.....</b>	<b>57</b>

## 1 INTRODUÇÃO

As Tecnologias da Informação e Comunicação (TICs) tem sido umas das maiores responsáveis pelas constantes mudanças sociais, organizacionais e até mesmo comportamentais. Dada sua presença marcante no cotidiano percebeu-se mudanças comportamentais, ao invés de se utilizar o relógio analógico, por exemplo, as horas passaram a serem verificadas através do celular (ALVES, 2005).

Para inúmeros professores, os jogos são vistos como uma atividade sem importância, irrelevante para aprendizagem por não apresentarem resultados sólidos e expressivos. Contudo, ao jogar, os alunos demonstram seriedade, comprometimento, concentração e, como qualquer atividade lúdica, traz divertimento (KISHIMOTO, 2003).

Entretanto, os métodos tradicionais de ensino onde o professor é tido como provedor de informações e o aluno como um recipiente destas, tem apresentado algumas dificuldades ao longo dos últimos anos. A dificuldade no aprendizado, assimilação do conteúdo de fato é notória e, na maioria dos casos, é possível identificar falta de motivação do aluno (KAPP, 2012).

A educação com uso lúdico tem se mostrado uma importante ferramenta, pois aproxima o aluno do conteúdo através da motivação e ganho de interesse, pois a atividade lúdica agrada, entretém, prende atenção, entusiasmo e ensina com maior eficiência, que estimula diversos sentidos do corpo ao mesmo tempo não tornando cansativo o ato de estudar (FALKEMBACH, 2013).

As TICs têm influenciado crianças, jovens e adultos a buscarem todo tipo de informação, principalmente pelo fácil acesso à *Internet* por meio de computadores, celulares, entre outros dispositivos móveis. No entanto, mesmo com todo esse potencial eletrônico e digital, observa-se que os jogos digitais educacionais ainda são pouco empregados em sala de aula. Balasubramanian e Wilson (2006) citam que, diversos professores têm dificuldade em encontrar e utilizar bons jogos que auxiliem no processo de ensino e aprendizagem.

Se por uma ótica os *softwares* precisam seguir uma série de requisitos e atender os propósitos para os quais foram elaborados, os jogos computadorizados precisam criar a sensação de imersão nos usuários, além de proporcionar entretenimento, diversão e tornar sua utilização agradável (BATTAIOLA, 2000).

Portanto, trabalhar na construção desses jogos representa lidar com um projeto interdisciplinar, aproximando os aspectos computacionais de outras ciências, tais como a Educação, Psicologia, *Design* Gráfico e Música. Ao mesmo passo que o torna fascinante seu desenvolvimento, requer muito conhecimento tornando complexo sua implementação por envolver Computação Gráfica, Inteligência Artificial, Redes de Computadores e Análise de Algoritmos (BATTAIOLA, ELIAS, DOMINGUES, 2002).

## 1.1 OBJETIVOS

Nesta seção cita-se o objetivo geral e os objetivos específicos deste trabalho.

### 1.1.1 *Objetivo geral*

O principal objetivo deste trabalho é desenvolver um *software* educacional com aplicando-se técnicas de gamificação e que possua uma ferramenta de autoria para edição do conteúdo.

### 1.1.2 *Objetivos específicos*

Para que seja possível a conclusão do trabalho, alguns objetivos específicos são necessários:

- Efetuar a modelagem do sistema;
- Desenvolver o diagrama de componentes;
- Elaborar o banco de dados do programa;
- Implementar o *software* educacional gamificado;
- Implementar uma ferramenta de autoria para as questões do conteúdo;
- Aplicar critérios de gamificação ao *software*.

## 1.2 JUSTIFICATIVA

Este trabalho, visando contribuir com a comunidade acadêmica e auxiliar neste cenário, apresenta uma solução que pode atender o anseio dos professores, auxiliar na elaboração de provas, incentivar o aluno a praticar exercícios de modo constante e com menor complexidade de implementação do que um jogo tradicional computadorizado exige.

De modo geral o sistema deverá contribuir como uma solução para facilitar o ensino-aprendizagem, já que se utiliza um meio de interação mais realista e próximo do cotidiano do aluno. Utilizando de elementos da gamificação para contribuir com a experiência do aluno de forma diferenciada além de permitir ao professor construir questões que serão resolvidas pelos seus alunos em segundo momento através da ferramenta de autoria.

## 1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos principais; o primeiro define as motivações que levaram a idealização e execução do projeto, os objetivos esperados e a justificativa de implementação.

No segundo capítulo, são apresentados o embasamento teórico, as tecnologias e ferramentas utilizadas para o planejamento, modelagem e desenvolvimento do projeto, explicitando as características levadas em consideração no momento de sua escolha.

No terceiro capítulo é descrita a modelagem do *software*, aplicando os conceitos, tecnologias e ferramentas apresentadas além de algumas explicações sobre a utilização de cada etapa do desenvolvimento.

Por fim, no último capítulo são apresentadas as conclusões, como também as dificuldades encontradas e sucessos obtidos no decorrer da execução do projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é abordado todo o embasamento teórico utilizado para compor a estrutura do trabalho e desenvolvimento do sistema. Aborda-se de modo explicativo os conceitos de *software* educacional, jogos educacionais, gamificação, e ambientes gamificados que foram fundamentais para embasar o trabalho.

### 2.1 SOFTWARE EDUCACIONAL

Segundo Sommerville (2003), *software* é caracterizado como um programa de computador e toda a documentação associada a ele. Já Pressman (2011) define *software* como um elemento de sistema lógico, e não físico que não se desgasta.

Desde a década de 70, um tipo mais “específico” de *software* obteve crescimento de forma acelerada. Alguns países como Inglaterra, França e EUA, entre vários outros, desenvolveram projetos para inserção dos computadores na educação. Como consequência direta tiveram de desenvolver *softwares* específicos para suas necessidades, os chamados *softwares* educacionais (CAMPOS, 1989).

Um *software* educacional pode ser considerado como tal desde que atenda um objetivo educacional de relativa importância, independente da finalidade com a qual ele foi concebido. De maneira simples a ideia desse tipo de *software* é fazer com que a criança ou adolescente aprenda o conceito, exercite a fixação por meio de exercícios ou adquira a habilidade embutida na aplicação, realizando-as através de meio lúdico independentemente do *software* ter sido desenvolvido com enfoque educacional (PIETRUCHINSKI, 2016).

A respeito dos termos computacionais relacionados com a elaboração de um *software* educacional, considera-se adequado o desenvolvimento de forma iterativa e incremental, dado os seguintes aspectos (SOMMERVILLE, 2003):

- Incremental, levando-se em consideração a visão de que o *software* terá o seu escopo definido na primeira etapa, conjuntamente entre educadores e desenvolvedores, sendo posteriormente dividido em estágios para entrega. Desta forma, viabilizando a validação dos objetivos de aprendizagem para cada estágio da aplicação;

- Iterativo, em que partes do processo são repetidas, sendo aplicado às etapas do processo proposto. Sendo assim, também é considerado que não há uma especificação completa do sistema até que o estágio final seja especificado”.

Na contemporaneidade, tem-se que lidar com métodos, procedimentos e ferramentas para aumentar a produtividade e qualidade dos produtos. Não é diferente para projetos de *softwares* educacionais que, além de envolver em seu desenvolvimento conhecimentos multidisciplinares, os produtos devem refletir os objetivos educacionais propostos e o ambiente de aprendizagem almejado, criando situações que estimulem o desenvolvimento das habilidades desejadas (DETERDING, 2011, PRESSMAN, 2011).

Para tanto, os *softwares* com a característica educacional devem ser estruturados levando em consideração aspectos pedagógicos e o conteúdo adequado ao público que futuramente fará uso. Isto porque quando bem empregados trazem benefícios, porém quando não, podem atrapalhar produzindo resultados negativos (PIETRUCHINSKI, 2016).

## 2.2 JOGOS EDUCACIONAIS

As pessoas são motivadas a jogar por quatro razões específicas: para obterem o domínio completo do assunto; para aliviarem o estresse; como forma de entretenimento; e como meio de socialização. Esses aspectos podem ser analisados de forma conjunta ou separadamente. Além disso, observaram-se quatro diferentes aspectos de diversão durante o ato de jogar: quando o jogador está competindo e busca vitória; quando está imerso na exploração de um universo; quando a forma como o jogador se sente é alterada pelo jogo; e quando o jogador se envolve com outros jogadores (ZICHERMANN e CUNNINGHAM, 2011).

Desses aspectos citados e das motivações encontradas na seção de Justificava deste trabalho, temos uma relação do entretenimento possibilitado pelos jogos atrelado ao aprendizado dos conteúdos das disciplinas, ou fixação dos mesmos, possibilitando maior engajamento já que reforça o que foi visto em sala de aula.

O ato de aprender utilizando jogos pode ser associado a duas premissas: a primeira é de que os jogadores mudaram com o decorrer dos anos, e a segunda de que as experiências são outras para essa geração atual que joga. No âmbito estudado, crianças e adolescentes gostam de jogar porque podem errar sem que haja uma recriação ou penalização permitindo que arrisquem novos caminhos ou soluções alternativas, que nem sempre poderiam utilizar em situações reais, para resolução dos problemas (MATTAR, 2010).

Uma definição sucinta para jogos educacionais seria aqueles jogos que apresentam a finalidade educativa promovendo de maneira lúdica o aprendizado, levando os jogadores à aprendizagem de habilidades ou conhecimento divertindo enquanto os ensinam (SOFFA e ALCÂNTARA, 2008).

### 2.3 AMBIENTES GAMIFICADOS

Os ambientes gamificados são capazes de interagir com as emoções e desejos dos usuários, são eficazes para o engajamento do indivíduo pois possuem elementos que contribuem para a motivação, que podem ser de dois tipos: as intrínsecas e as extrínsecas (ZICHERMANN e CUNNINGHAM, 2011):

- Intrínsecas: originada dentro do próprio sujeito e necessariamente não está baseada no mundo externo;
- Extrínsecas: são baseadas no mundo que envolve o indivíduo e lhe são externas, tendo como ponto de partida o desejo do sujeito em obter uma recompensa, como, por exemplo, reconhecimento social

Há alguns anos estuda-se as motivações indiretas dos jogos na aprendizagem. Segundo Alves (2005), as motivações podem contribuir alavancando o potencial do aluno como influência indireta, sendo estas motivações as seguintes:

- Elevar a memória visual, a coordenação motora ampla e fina, percepção visual de tamanho, cor, detalhes, forma e posição dos elementos expostos na tela;
- Estimular a reorganização de funções cognitivas como a criatividade, atenção, imaginação, coordenação motora e memória;

- Como as perguntas podem ser variadas e podem ser respondidas em momentos distintos, a fixação da aula, do tema e resolução de exercícios deixa de ser um tédio repetitivo;
- Após análise dos resultados o professor pode inferir quais alunos precisam de um auxílio maior, por exemplo.

## 2.4 GAMIFICAÇÃO

A aplicação de elementos, mecanismos, dinâmicas e técnicas de jogos no contexto fora do jogo, ou seja, na realidade do dia a dia profissional, escolar e social do indivíduo, é compreendida como gamificação, que é a tradução do termo *gamification* cuja criação foi possivelmente do programador britânico Nick Pelling, em 2003 (PELLING, 2003).

As mídias digitais e indústrias utilizaram a terminologia a partir de 2008 segundo datam registros gerais encontrados na *internet*, porém foi em 2010 que o termo ganhou popularidade devido sua introdução em conferências sobre mídias digitais e tecnologia (MATTAR, 2010).

O crescente número de pesquisas sobre gamificação em diversas áreas, o interesse pela gamificação e seu emprego têm aumentado também na área de educação, salientando que, com pouca idade os alunos já possuem contato com a tecnologia de modo geral (LEE e DOH, 2012; DOMÍNGUEZ et al., 2013).

Este crescente interesse pode ser explicado, principalmente, pelo potencial da gamificação para influenciar, engajar e motivar pessoas. Estudos sobre o uso de jogos e elementos de jogos em educação vêm sendo realizados há algumas décadas, porém, nos últimos anos, o interesse pelo tema vem aumentando em ritmo acelerado (KAPP, 2012).

Ainda sobre Kapp (2012), gamificação trata-se da aplicação de elementos utilizados no desenvolvimento de jogos eletrônicos, tais como “o uso de mecânicas, estética e pensamentos dos jogos para envolver pessoas, motivar a ação, promover a aprendizagem e resolver problemas”, em outros contextos, não relacionados a jogos.

A seguir uma breve explanação dos elementos que compõem essa definição (KAPP,2012):

- Mecânicas: os elementos mais básicos da gamificação, como as regras, as saídas quantificáveis, os níveis, o *feedback*, as recompensas, o sistema de pontuação entre outros pontos secundários.
- Pensamentos: segundo o autor, essa é a mais importante das características, a ideia aqui é de transformar um determinado contexto ou situação em uma espécie de jogo, incorporando a ela elementos que tornam *games* em experiências agradáveis e prazerosas;
- Jogos: o principal objetivo é criar um sistema que envolva os jogadores em um desafio abstrato, definido por regras, interatividade e *feedback*, que resulte em uma saída positiva, através de um sistema em que os indivíduos desejem investir seu tempo, raciocínio e energia;
- Envolver: um objetivo primário do processo de gamificação é capturar a atenção do indivíduo, ou de um grupo, envolvendo-o na experiência criada pelo sistema, de modo que sua participação seja mais significativa;
- Pessoas: podem assumir esse papel, aprendizes, alunos, professores, clientes ou jogadores. Serão os indivíduos que se envolverão nos processos criados e serão motivados a agir;
- Motivar a ação: segundo o autor, motivação como um todo é um processo que cria energia e dá direção, propósito e significado ao comportamento e ações. Para existir motivação, o desafio proposto não pode ser muito difícil nem muito fácil, mas adaptado à capacidade de cada um. Motivar a participação em uma atividade é um elemento central da gamificação;
- Promover a aprendizagem: na gamificação existe maior interesse em unir esses elementos educacionais e de jogos para alcançar a similaridade com os jogos de entretenimento, o que resulta em uma linguagem a qual os indivíduos e jogadores estão mais acostumados conseguindo então alcançar essas metas de forma aparentemente mais eficiente;
- Resolver problemas: a gamificação tem um grande potencial de ajudar a resolver problemas. A sua natureza competitiva encoraja muitos ao esforço máximo para obter a vitória dentro desses contextos.

A gamificação propõe, como estratégia aplicável aos processos de ensino aprendizagem nas escolas ou outro ambiente de aprendizagem, utilizar um conjunto de elementos similarmente encontrados na maioria dos jogos e aplicá-los nesses processos, com o intuito de gerar níveis semelhantes de envolvimento, dedicação e demais esforço que os jogos geram por si (ZICHERMANN e CUNNINGHAM, 2011).

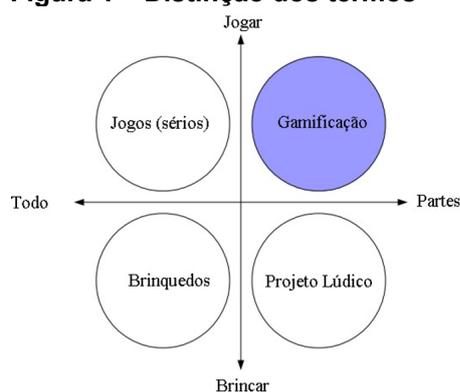
Utilizando-se então de uma definição focada nas práticas educacionais, que satisfaz o maior número de aspectos, e, possivelmente a torna uma prática mais compreensiva e menos propensa a insatisfações, neste trabalho, aplicaram-se algumas destas definições durante a implementação do projeto.

## 2.5 SOFTWARE EDUCACIONAL GAMIFICADO

Dada que, a gamificação se refere à aplicação de elementos dos jogos fora do contexto dos jogos (DETERDING, 2011), pode-se inferir que um *software* gamificado trata-se de um tipo de *software* que possui elementos de jogos como parte integrante da composição final (PIETRUCHINSKI, 2016).

Groh (2012) apresenta uma diferenciação entre gamificação e jogos sérios (*serious gaming*), e entre design e interação lúdicas sobre as dimensões de brincar/jogar e partes/todo. A Figura 1 a seguir demonstra a distinção dos termos.

**Figura 1 – Distinção dos termos**



**Fonte: Pietruchinski (2016)**

A cerca da imagem, a dimensão partes/todo coloca a gamificação como a utilização de partes de um projeto lúdico, contrastando com jogos sérios, que são para fins que envolvem treinamento e quase nenhum entretenimento.

Segundo Pietruchinski (2016) o *software* educacional gamificado é aquele que possui elementos de jogos para tornar-se mais atrativo aos alunos ou jogadores que o utilizam.

Como diferencial no presente projeto, propõe-se a aplicação de alguns dos critérios de gamificação, como *ranking*, uso o tempo e *feedback* que foram apresentados nesta seção visando o maior engajamento do aluno na resolução dos exercícios e como consequência um aumento em seu aprendizado.

## 2.6 FERRAMENTA DE AUTORIA

O intuito do *software* educacional é, que seja proporcionado direta ou indiretamente algum aprendizado ao indivíduo e/ou aluno. Para auxiliar na construção das informações que futuramente irão compor esse aprendizado existem as ferramentas de autoria.

Uma das definições para ferramentas de autoria é a de sistemas, ou parte de um todo, que possuem como objetivo principal facilitar o processo de elaboração de sistemas multimídias, reduzindo custos e o tempo de desenvolvimento do produto almejado. Para tanto tais ferramentas procuram assistir o programador e/ou professor com ambientes de programação de alto nível permitindo uma maior abstração na especificação de tarefas (MURRAY, 2003).

As ferramentas de autoria podem ser qualquer aplicativo, vários aplicativos e, até mesmo, parte de um aplicativo com as quais o autor interage para inserir, editar ou elaborar um determinado conteúdo, que será utilizado por outras pessoas. Em âmbito geral, é um recurso para que leigos em computação ou profissionais de outras áreas possam construir com agilidade, qualidade e onde quer que estejam, não se limitando a lugar ou situação física, um determinado conteúdo ou programa (MURRAY, 2013).

Atualmente encontram-se, em pesquisas na *Internet*, uma gama de ferramentas de autoria sendo comerciais, gratuitas e livres para criação de questões ou *design* de questões que podem auxiliar no processo de aprendizagem dos alunos. A escolha de uma ferramenta deve ter em conta aspectos como: a complexidade, a metodologia pedagógica escolhida e o nível de criatividade (MURRAY, 2003).

A ferramenta de autoria, portanto, deve oferecer um ambiente gráfico, ou seja, deve abstrair os utilizadores das instruções de programação provendo uma maior facilidade de utilização e flexibilidade da ferramenta (REIS e DAMIÃO, 2011).

Para uma melhor compreensão dos exemplos das ferramentas a seguir, o acrônimo WYSIWYG precisa ser explicado. Trata-se do acrônimo da expressão em inglês "*what you see is what you get*", cuja tradução remete a algo como "o que se vê é o que se obtém", o próprio *Microsoft Word* é um exemplo (WYSIWYG, 2018).

Ainda em WYSIWYG (2018) são citadas citar ferramentas de autoria já existentes como:

- *CourseLab*: de fácil utilização, empregada na criação dos conteúdos das aulas, que oferece um ambiente WYSIWYG de programação. O conteúdo interativo de alta qualidade pode ser publicado na *Internet*;
- *Myudutu*: com um conjunto de ferramentas WYSIWYG, permitindo aos usuários criar cursos em um ambiente acessível, de forma rápida, sem grandes conhecimentos de programação ou de desenvolvimento de aplicações *web*;
- *AutoScreenRecorder*: aplicativo ideal para fazer pequenas gravações da sua área de trabalho, no formato *.avi*. O programa é leve e muito fácil de usar, sendo possível capturar a tela do computador ou alguma janela específica, por exemplo.
- *Prezzi*: permite criar apresentações interativas, com efeitos de movimento e zoom, com efeitos variados o que causa um alto impacto na audiência.

Portanto a temática sobre ferramentas de autoria possui uma gama grande de possibilidades, cada uma auxiliando em pontos específicos com aplicabilidades variadas, como auxiliar na elaboração de questionários e provas.

Uma das dificuldades para professores está e na elaboração de diversas provas ao longo do ano letivo e a criação de exercícios cada vez mais distintos, condizentes com a realidade dos alunos. Nesse contexto uma ferramenta de autoria embutida em um *software* educacional auxiliaria no processo de produzir conteúdo didático seja ele para fixação ou avaliativo.

## 2.7 TECNOLOGIAS

Nesta seção são apresentadas as tecnologias utilizadas neste trabalho, sendo elas o *Scrum*, PHP, HTML, CSS e *JavaScript* fundamentais para implementação do software.

### 2.7.1 *Scrum*

Com base teórica no desenvolvimento e gestão ágil de projetos, a adoção da metodologia ágil *Scrum* tornou-se mais indicada, pois pode permitir reduzir o tempo de desenvolvimento, as entregas serem mais rápidas, lidar melhor com as mudanças inevitáveis no escopo. O *scrum* hoje é aplicado em projetos de áreas distintas e utilizado em diferentes mercados, que incluem empresas de caminhões, desenvolvimento de *software* e empresas de *marketing*, por exemplo (SABBAGH, 2014).

Para que os resultados sejam obtidos alguns valores representam os pilares para todo trabalho que faz uso da metodologia, que são: foco, coragem, franqueza, compromisso e respeito (SCRUM ALLIANCE, 2012).

Embora esses valores sejam voltados para a gestão e implementação de projetos compostos por várias pessoas, ou seja, equipes, para esse trabalho o enfoque foi a aplicação da metodologia fazendo uso dos artefatos do *Scrum* como *Product Backlog*, as *Sprints*, as entregas e os acompanhamentos.

De modo simplificado o *Product Backlog* é uma lista, não literal, de tudo que se acredita que será desenvolvido no decorrer do projeto, essa lista é organizada pelo *Scrum Master*,

A definição das *User Stories*, requisitos do projeto, é de responsabilidade do *Product Owner*, representada pela orientadora neste trabalho, de acordo com a necessidade. Após a *Sprint Planning*, isto é o planejamento das atividades para as próximas duas semanas, através de uma reunião quinzenal, a partir do que se tinha no *Product Backlog* compõem-se a *Sprint* que é outra lista dos itens que devem ser desenvolvidos no prazo, quinze dias (SABBAGH, 2014).

Como forma de acompanhamento, diariamente ocorrem pequenas reuniões de quinze minutos cada, essas reuniões são chamadas de *Daily Scrum*, porém, por questões de disponibilidade e tempo. para o atual projeto essa etapa tornou-se inviável não sendo utilizada.

A Figura 2 demonstra o ciclo de vida do *Scrum*, com exceção da *daily scrum*, os demais fazem-se presentes neste trabalho. Todas atividades necessárias para o desenvolvimento do projeto são inseridas no *Product Backlog*.

**Figura 2 – Ciclo de Vida da Metodologia Scrum**



**Fonte: Scrum Alliance (2018)**

Feita a primeira reunião para definição da *Sprint* inicial, as demais reuniões ocorrem tanto para verificação do que foi concluído com sucesso e designação do que deve ser desenvolvido para a próxima entrega.

Ressalta-se que o enfoque principal do trabalho é a entrega do *software* conforme proposto e almejado nos objetivos e que a metodologia apresentada irá auxiliar na execução e gerência do mesmo, não sendo objeto de discussão

### 2.7.2 PHP

PHP é um acrônimo para *Hypertext Preprocessor*, originalmente significa *Personal Home Page*, e é uma linguagem de programação de código aberto que possui utilização ampla e adequada ao desenvolvimento *web* pois pode ser embutida dentro do HTML, linguagem que será abordada em um tópico futuro (CONVERSE, PARK, 2003).

A linguagem tem pouco a ver com *layout*, ou qualquer coisa relacionada à aparência da página *web*. A maior parte do que o PHP realiza ou é responsável fica

invisível ao usuário final, ou seja, alguém visualizando uma página criada com PHP não será capaz de afirmar que ela foi ou não escrita com PHP, porque o resultado final será sempre o HTML (CONVERSE, PARK, 2003).

Um dos pontos fortes que encontrou-se ao pesquisar sobre a linguagem é que, para um iniciante pode ser simples, e ao mesmo tempo que oferece para os profissionais que trabalham com ela recursos avançados como conexão com outras plataformas bem como conexão com sistemas bancários.

A linguagem será também responsável pela maior parte lógica do *software*, isto é, a parte de *back-end* que envolve as ações, eventos, inserção de informações, consulta e alteração de tudo que é inserido dentro do sistema quando estiver em funcionamento.

### 2.7.3 HTML

HTML é um acrônimo para *HyperText Markup Language*, que em português, significa Linguagem de Marcação de HiperTexto, e ela é o componente mais básico das páginas *web* e da *web* como um todo por estar presente em praticamente tudo que é relacionado à *Internet* e possui uma interface.

Analisando a terminologia, temos que hipertexto se refere aos links que conectam as páginas umas nas outras, tanto dentro de um site quanto outros diferentes sites. O *Markup* é usado para mostrar os elementos, texto, imagens, e outros necessários, sendo comum a utilização de *tags* como `<head>`, `<title>`, `<body>`, `<div>`, e muitas outras (CASTRO; HYSLOP, 2013).

O criador foi Tim Berners-Lee, um físico britânico, tendo sua primeira versão no ano de 1991 que originalmente teve motivação para sua existência a resolução de um problema de Tim, disseminação das pesquisas entre ele e seu grupo de colegas (CASTRO; HYSLOP, 2013).

A maior utilização no trabalho foi usufruir das *tags* para estruturação de todas as páginas do *software* agilizando o desenvolvimento do mesmo.

#### 2.7.4 Cascading Style Sheet

*Cascading Style Sheet* é uma maneira de facilitar o desenvolvimento de aplicação *Web*, pois centraliza as configurações referentes a exibição dos elementos como cores das letras, bordas, espaçamentos, tamanho dos títulos, posição de um elemento na tela ou tamanho de uma imagem em um arquivo *.css*, futuramente importado para as páginas, garantindo assim uma padronização e melhorando o carregamento da mesma (W3C, 2017).

Uma desvantagem da utilização do CSS é que cada navegador possui um motor de renderização, ou seja, cada um irá interpretar o código da página de uma forma, podendo gerar conflito, comumente encontramos erros ao utilizar o *Internet Explorer* da *Microsoft*.

Neste trabalho a utilização do CSS será essencial para compor a parte visual do *software* e melhorar a visibilidade dos elementos.

#### 2.7.5 JavaScript

*JavaScript* é uma linguagem de programação com orientação a objetos que pode ser usada para aumentar a interatividade das páginas *Web*, ela suporta a utilização de expressões regulares, objetos, vetores e matrizes. Ela executa os *scripts* que estão inseridos em um código HTML (FLANAGAN, 2006).

Embora essa linguagem possua uma sintaxe parecida com a linguagem Java, ela permite programação do lado cliente no arquivo *.html*. O *JavaScript* possui diversas finalidades, desde enviar uma mensagem após alguma ação do usuário, até exportar os dados para um arquivo de texto ou então para importar um arquivo *.sql* para o *software* (DUCKETT, 2013).

Neste projeto o *JavaScript* foi usado para tratamento dos erros através de mensagens emitidas pelo *browser*, gerenciar eventos, estava presente nos *frameworks* utilizados, auxiliando no desenvolvimento ágil.

## 2.8 FERRAMENTAS

Aqui serão apresentadas todas as ferramentas utilizadas para implementação do *software*, programa para codificação, para modelagem, *plugin* que auxilia na composição da ferramenta de autoria, *software* que realiza execução do servidor que suportará o projeto.

### 2.8.1 Microsoft Visual Studio Code

O *Microsoft Visual Studio Code* é uma ferramenta desenvolvida pela *Microsoft* e tem ganhado um espaço e presença marcantes no cenário de desenvolvimento e programação de aplicações web pois possui suporte não somente a *ASP.NET*, mas também ao *Node.js*, *PHP*, *Ruby*, *Python*, entre outras mais.

Sua origem foi no ano de 2015 e sua licença é gratuita tendo seu código fonte disponibilizado no *GitHub*. Vale explicitar que o editor não é uma cópia do *Visual Studio Enterprise*, por sua vez, é semelhante a outros editores como o *Sublime Text*, *Atom* ou *Brackets* (*VISUAL STUDIO CODE*, 2018).

Neste trabalho usou-se o *VS Code* como ferramenta principal para elaboração de todos os códigos do projeto independente da linguagem.

### 2.8.2 XAMPP

Como já se sabe, a tarefa de instalar um servidor *web Apache*, inicialmente requer certo conhecimento, pois pode se tornar mais difícil ainda se for necessário acrescentar *MySQL*, *PHP* e *Perl*, portanto, após pesquisas encontrou-se a ferramenta *XAMPP* para auxiliar no desenvolvimento.

O objetivo do *XAMPP* é fornecer através de uma distribuição fácil de instalação os recursos do *Apache*, a ferramenta é gratuita, fornecida e mantida pela *Apache Friends*, hospedada atualmente na *SourceForge*. Ele é um utilitário que instala o *Apache*, *MySQL*, *PHP*, *Perl*, *PHPMyAdmin* entre outros recursos (*XAMPP*, 2018).

Após a instalação e ativação dos componentes necessários podemos acessar o painel de administração através do link *http://localhost* de algum dos *browsers* presentes no computador do usuário.

Para que o projeto possa funcionar existe um diretório específico, genérico para toda utilização do *XAMPP*, onde é inserido todo conteúdo que futuramente será interpretado. No *Windows* por padrão o diretório encontra-se “*C:\xampp\htdocs\nome\_da\_pasta*”.

Para este trabalho foi utilizada a versão PHP 5.6.38 por se tratar, até o momento, de uma versão mais estável e com maior documentação. A utilização do *XAMPP* foi essencial pois praticamente todo desenvolvimento foi utilizando a linguagem PHP e necessita de um servidor com *Apache* instalado para interpretação e execução do código (PHP:DOCUMENTATION, 2018).

### 2.8.3 Microsoft Visio

A modelagem de um *software* é essencial para facilitar o papel do desenvolvedor além de auxiliar em possíveis dúvidas, como por exemplo, em relação a composição do banco de dados. O *Microsoft Visio* é um complemento do pacote *Office* da *Microsoft*, o qual originou-se na década de 1990 como produto da *Shapeware Corp.*, comprado em 2000 pela *Microsoft* (MICROSOFT VISIO, 2018).

Essa ferramenta permite desenhar e modelar uma grande variedade de diagramas, incluindo fluxogramas, organogramas, diagramas de fluxo de dados, diagramas de classes, modelos relacionais de banco de dados, mapas 3D entre outros (MICROSOFT VISIO, 2018).

Como ferramenta de auxílio para o projeto atual o uso principal foi a modelagem do banco de dados, e também a criação do diagrama de componentes.

### 2.8.4 JQuery

Desenvolvida por John Resig, em 2006, se tornou uma das bibliotecas *JavaScript* mais populares na *internet*, de código aberto, ela é disponibilizada para fins pessoais ou comerciais sem necessário nenhum pagamento pelo uso (DUCKETT, 2013).

Essa biblioteca, *JQuery*, possui funções em *JavaScript* que interage com o HTML. Seu principal objetivo é descomplicar os *scripts* interpretados no navegador do usuário, simplificar a navegação em documentos HTML, criar animações, manipular eventos entre tantos outros. Ela também oferece suporte a *plug-ins* e conta com uma comunidade que atualiza constantemente novas funções à biblioteca e corrige possíveis erros (RESIG, 2006).

Outro fato interessante é que com o *JQuery* possuímos também, suporte ao CSS, mesmo que o navegador não tenha suporte, a utilização é possível dado que o próprio *JQuery* é quem irá implementar os seletores CSS criando certa independência no momento da execução.

No presente trabalho a biblioteca se torna essencial para manipular diversos objetos e eventos, assistindo também no retorno de algumas mensagens ao usuário.

#### 2.8.5 *CKEditor 4*

Maximizando a produção e poupando tempo, visando a entrega do protótipo inicial, no prazo correto, foi necessário pesquisar uma ferramenta ou *plugin* que permitisse a edição das questões de modo mais simples e eficaz pelo professor e ao mesmo tempo não custasse muito para ser implementado pelo programador.

Uma solução plausível encontrada foi o *CKEditor*, editor no WYSIWYG gratuito e com código aberto. Ele foi desenvolvido em *JavaScript* sua função é criar páginas visualmente, gerando um código HTML como saída (CKEDITOR, 2018).

Através dos módulos que ele possui é possível inserir no código uma caixa que contém o campo editável para inserção do cabeçalho da questão, sendo possível inserir uma tabela, símbolos e até mesmo expressões matemáticas.

O *plugin* permite também que seja aplicada tabulações no texto e contém alguns tipos de fontes para diferenciar o título do subtítulo da questão, além de possibilitar o uso de tabulações e inserção de imagens.

Portanto, o *plugin* torna multidisciplinar a ferramenta de autoria por permitir que a questão seja inserida independentemente do contexto e disciplina que ela pertença, sendo possível criar inclusive questões de vestibular ou concursos através dos componentes citados.

Outro ponto que justifica sua escolha é que, este é popular e nativo de alguns sistemas de gerenciamento de conteúdo como *Drupal*, o *PHP-Nuke* ou *Joomla*, entre demais aplicativos existentes na *internet* que o integram em sua estrutura (CKEDITOR, 2018).

## 2.9 ANÁLISE DE REQUISITOS

Através da análise de requisitos tem-se o ponto de partida para toda a definição do sistema, sendo estes requisitos fatores essenciais no desenvolvimento do produto final. A análise pode ser uma descrição de uma funcionalidade do usuário, especificações do processo, restrições do sistema, entre outros.

Seguindo corretamente o processo de análise de requisitos teríamos que contemplar o Processo Unificado da Rational (RUP), porém, como não se tem como foco uma análise completa, descrevendo todos os requisitos, seguem, para fins de conhecimento, um breve resumo das divisões do RUP que se utilizou (PRESSMAN, 2011):

- Do Projeto: requisitos necessários do negócio, gerenciamento e entrega do produto.
- Funcionais: estabelecem como o sistema irá agir, o que se deve fazer, as funcionalidades e serviços do mesmo.
- Não-Funcionais: são definidas as propriedades do sistema e suas restrições, como a tempo de resposta, confiabilidade.

Atendendo às demandas do *Product Owner* e adaptando para realidade do trabalho, dividiu-se em duas partes a análise dos requisitos. A primeira parte sendo mais abrangente em relação á todo escopo do trabalho, e, a segunda mais específica no que diz respeito aos pontos que iriam compor cada uma das futuras *sprints*.

A respeito da primeira parte, ela corresponde a análise de projeto, descrita anteriormente, pois trata-se dos aspectos necessários para que a execução daquilo que foi proposto, logo foi realizada por primeiro para que através dela fosse possível extrair os requisitos funcionais e não-funcionais.

Trata-se, portanto, de uma análise geral, sendo menos específica e mais abrangente por isso possui poucos itens resultantes da análise, seu resultado está ilustrado na Quadro 1.

**Quadro 1 – Relação de Requisitos do Projeto e Prioridades**

Requisitos do Projeto	Prioridade
1. Elencar as necessidades para o desenvolvimento	Alta
2. Estudo e aprendizagem das linguagens e ferramentas	Alta
3. Definir uma metodologia de desenvolvimento	Média
4. Implementar o sistema	Alta

Fonte: Autoria Própria

Na Quadro 2 é possível visualizar os requisitos funcionais com sua respectiva prioridade, bem como no Quadro 1, não são ordenados pela prioridade, mas pela ordem que elencou-se cada uma das necessidades pelo *product owner*.

**Quadro 2 – Relação de Requisitos Funcionais e Prioridades**

Requisitos Funcionais	Prioridade
1. <i>Software</i> deverá possuir <i>login</i> e um painel para o professor visualizar todas as informações e questões	Alta
2. O Aluno deve conseguir se cadastrar e também possuir um painel geral contendo os questionários	Alta
3. O sistema deve permitir ao professor a elaboração em pelo menos três diferentes tipos de questão.	Média
4. Após inserida as diversas questões o sistema possibilitará a montagem de questionário e atribuição aos alunos	Média
5. Cada questionário deve gerar uma pontuação ao aluno e o professor selecionará se aquele questionário fornecerá um <i>feedback</i> ao aluno.	Alta
6. Conceitos de gamificação devem ser aplicados, como <i>ranking</i> , utilização de tempo para resolução das questões, entre outros que julgar importantes.	Alta
7. Existindo tempo hábil deve-se tornar a plataforma adaptável a qualquer <i>layout</i> entre os diversos dispositivos que existem.	Baixa

Fonte: Autoria Própria

Essa segunda parte da análise foi essencial pois possibilitou uma maior compreensão do escopo total do trabalho além de auxiliar na entrega das *sprints* pois através dos requisitos funcionais somados de sua prioridade passaram a compor as entregas.

Ao finalizar das análises, seguiu-se para modelagem do banco de dados e implementação do *software* seguindo tanto a prioridade de cada módulo respeitando a sequência que ele fora descrito acima.

Após os estudos e pesquisas realizadas, obteve-se ter uma maior compreensão do que é um *software* educacional e quais suas peculiaridades, além dos itens que o distingue de um *software* empresarial, por exemplo.

Outro fator que contribuiu para fase do desenvolvimento foram os conceitos estudados sobre gamificação pois através deles os itens solicitados pelo *product owner* e demais necessidades puderam ser implementados já que houve melhora na abstração do significado de gamificar um *software*.

A metodologia de desenvolvimento estudada auxiliou na separação das diversas partes e histórias que seriam implementadas diminuindo o tempo que seria usado e aumentando a produtividade.

### 3 DESENVOLVIMENTO DO SOFTWARE E SEUS RESULTADOS

O objetivo deste trabalho foi desenvolver um *software* educacional aplicando técnicas de gamificação referentes a personalização do perfil, *ranking* e *feedback* além da criação de uma ferramenta de autoria que possibilite ao professor a criação das questões que serão utilizadas nos questionários do próprio programa.

Nesta seção apresenta-se o software educacional e a ferramenta de autoria e também as adaptações na metodologia de desenvolvimento utilizada, os detalhes de cada módulo e particularidades que foram inseridas, além dos ajustes que foram necessários e os resultados obtidos.

#### 3.1 ADAPTAÇÕES NA METODOLOGIA

A metodologia *Scrum* aborda o desenvolvimento ágil de projetos, como foi aplicada ao desenvolvimento de *software*, neste trabalho, foram feitas algumas adaptações à metodologia no início do projeto, como estão citadas a seguir:

- Normalmente a equipe seria composta por mais de 3 membros, para este trabalho reduziu-se em apenas 2 membros, orientador e orientando, o segundo assumiu o papel de analista de requisitos e desenvolvedor;
- Apesar da metodologia sugerir as reuniões diárias, *Daily Scrum*, devido à fatores externos, como trabalho, as reuniões não foram realizadas;
- Não houve também a utilização do gráfico de *Burndown*, que demonstra dentro da metodologia o progresso dentro de cada *Sprint*.
- Por fim a presença do *Scrum Master* foi subtraída

E além dessas, todas as práticas normalmente aplicadas à uma equipe de desenvolvimento foram dispensadas. Outras mudanças menores foram surgindo e sendo superadas ao longo do tempo.

Embora as alterações foram necessárias, manteve-se o papel do patrocinador de negócio, a presença da lista de estórias, lista de tarefas e as reuniões de planejamento e revisão, pois sem elas a elaboração das correções, até mesmo as entregas, seria inviável.

### 3.2 PRODUCT OWNER

No *Scrum* todo projeto deve possuir um *product owner*, que é o responsável pelo projeto, não pela execução, mas quem patrocina o desenvolvimento daquela solução, neste trabalho a orientadora assume esse papel.

Em termos gerais, pode-se dizer que o orientador teve êxito cumprindo todas as obrigações, como patrocinador do negócio, desde o período de análise de requisitos e durante a execução do projeto. O acompanhamento foi realizado a cada quinze dias com o *product owner* presente, esclarecendo dúvidas e solicitando alterações na presença de algum erro.

### 3.3 ESTIMATIVAS

As estimativas foram realizadas da seguinte maneira: a cada quinze dias haveria um entrega definida pelo *product owner* e aceita pelo desenvolvedor, caso contrário durante a reunião, *sprint planning*, as tarefas a serem entregues eram revistas e negociadas.

Como parte implícita do trabalho, a análise de requisitos não foi contabilizada nas estimativas por ter composto a parte inicial e prévia do trabalho, fazendo então, parte de pesquisas e estudos.

O período total de desenvolvimento foi de 8 semanas aproximadamente, totalizando 5 dias de execução do projeto por semana. Vale ressaltar que a partir do primeiro *sprint* de desenvolvimento foi dedicado 4 horas por dia com exceção da quinta-feira e sexta-feira nas quais o orientando possuía aula presencial na instituição.

Como não havia um controle rígido das horas utilizadas diariamente para execução do projeto, o tempo gasto foi calculado pela média semanal utilizada entre a implementação e escrita do trabalho.

Para chegar ao tempo estimado foi utilizado como conteúdo de apoio o *layout* do portal Racha Cuca que contém uma variedade de temas e assuntos a serem respondidos nos questionários (RACHA CUCA, 2018), facilitando assim a escolha dos temas e apresentações das questões na ferramenta de autoria.

A Quadro 3 exibe as *User Stories* em ordem sequencial em relação ao conteúdo que deveria ser desenvolvido e suas respectivas datas de entrega previamente programadas na fase de análise de requisitos.

**Quadro 3 – Estimativa das User Stories**

<b>User Story</b>	<b>Data Estimada</b>
- Desenvolvimento da modelagem do <i>software</i> - Modelagem do banco de dados - Implementação do banco de dados	16/08/2018
- Desenvolver a tela de <i>login</i> - Cadastro do aluno e o painel de administração do professor	30/08/2018
- Desenvolver o cadastro das Disciplinas - Desenvolver o cadastro dos Assuntos de cada Disciplina	13/09/2018
- Desenvolver o cadastro de 3 tipos de Questões - Questionário e a Ferramenta de Autoria	27/09/2018
- Realizar a integração entre o questionário e os alunos - Aplicar os conceitos de Gamificação no <i>software</i>	11/10/2018

**Fontes: Autoria Própria**

### 3.4 MODELAGEM DO BANCO DE DADOS

Após diversas pesquisas e leitura de artigos, a metodologia Scrum não adota uma modelagem tal qual outras metodologias, como o XP (*Extreme Programming*) que utiliza o diagrama de classes como uma documentação posterior à programação, normalmente gerada através de engenharia reversa (TELES, 2004).

Como a metodologia não impõe qualquer forma particular de registrar os requisitos, além de reuniões regulares, a melhor maneira de lidar com essa ausência de um modelo de classes ou casos de uso, por exemplo, foi realizar as atividades e definições de escopo com a orientadora.

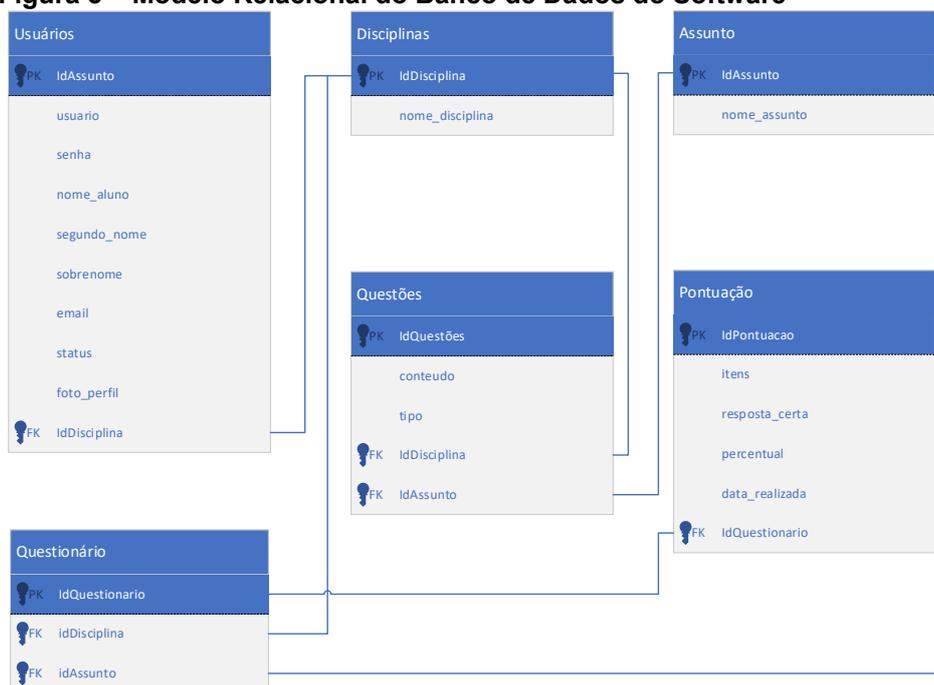
Portanto, a modelagem principal foi a do banco de dados, onde buscou-se atender todos os requisitos de armazenamento para que não fosse necessária uma mudança futura, ou uma alteração que viesse a comprometer a execução do projeto ou atrasar as entregas dos *sprints*.

Para desenvolver a modelagem do banco de dados utilizou-se a ferramenta *Microsoft Visio* somado de um *plugin* para realização da modelagem, porém, mesmo com o uso da ferramenta e do *plugin* não existe a utilização de cardinalidade entre as relações no modelo, o que não foi um problema do decorrer do desenvolvimento, pois a quantidade de tabelas não representou obstáculo na criação do *script .sql*.

Pela proposta buscar contemplar também os professores, omitiu-se o cadastro para o papel professor, logo não está presente no modelo relacional. Solucionando essa questão definiu-se um usuário e senha padrão no banco para que o futuro docente pudesse ser autenticado no sistema.

Portanto a modelagem do banco de dados se deu pela observação de cada necessidade do *software*, bem como campos necessários para o correto funcionamento dos *plugins* e ferramentas. O Modelo-Relacional com as relações entre as tabelas podem ser vistos na Figura 3.

**Figura 3 – Modelo Relacional do Banco de Dados do Software**



**Fonte: Autoria Própria**

É importante ressaltar que para primeira versão foi implementado a participação do aluno em uma única disciplina, devida a proposta ágil de desenvolvimento, mas para demais versões e ampliações do *software* é possível que o aluno participe de mais de uma disciplina.

Após ser finalizada a modelagem implementou-se o banco de dados em seguida os ambientes foram configurados e próximo passo foi a implementação da interface e da lógica do *software*.

### 3.5 PREPARAÇÃO DOS AMBIENTES

Antes da codificação foi necessário preparar o ambiente para o desenvolvimento, procedeu então com a instalação do *XAMPP* e *VS Code*. Após isto no diretório específico “*C:\xampp\htdocs\tcc*” criou-se um subdiretório para armazenar todos arquivos do projeto. Realizou-se também uma pesquisa para compor as imagens que seriam incluídas na interface do *software*.

Algumas extensões para o *VS Code* foram instaladas para ajudar no processo de programação, principalmente aqueles que, por meio das colorações alteram e distinguem variáveis, objetos, funções e estruturas, permitindo uma rápida percepção do código. Utilizou-se as seguintes extensões:

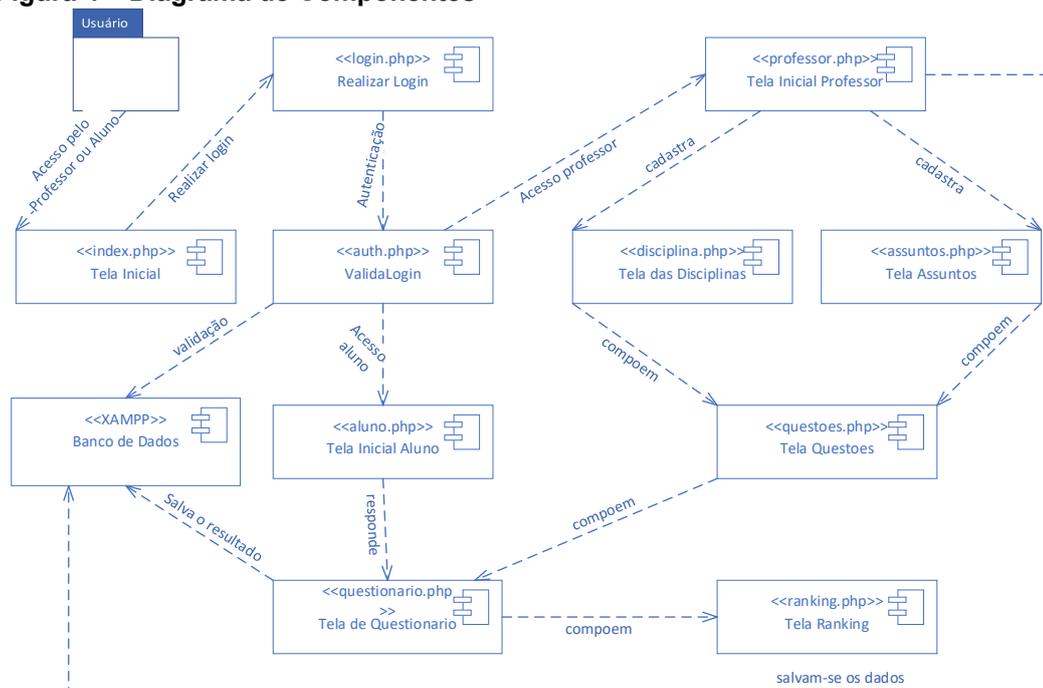
- *Auto Close Tag*: fecha *tags* HTML automaticamente conforme são inseridas;
- *Better Align*: minimiza o tempo para realizar indentação do código, seja PHP, HTML, CSS ou JavaScript;
- *Bracket Pair Colorizer*: ajuda na abertura e fechamento de parênteses e colchetes;
- *PHP Debug*: auxilia na depuração de possíveis erros no código;
- *PHP Extension Pack*: agiliza o desenvolvimento com sugestões para completar a função ou trecho do código em PHP

### 3.6 ARQUITETURA DO SOFTWARE

Embora a metodologia *Scrum* não trabalhe especificamente com uma modelagem ou arquitetura de *software* como MVC (*Model-View-Control*), por exemplo, optou-se por não utilizar nenhuma arquitetura específica, mas utilizar o diagrama de componentes.

O diagrama foi criado utilizando-se dos requisitos elencados compondo a base principal tanto para organização de fluxo durante o desenvolvimento bem como auxílio para criação da parte lógica do sistema. Na Figura 4 tem-se o resultado do diagrama.

**Figura 4 – Diagrama de Componentes**



**Fonte: Autoria Própria**

Cada módulo foi constituído de uma página que monta sua estrutura, outra para pesquisa, a terceira para inserção das informações e uma última para edição e exclusão das informações se necessário for.

Desta forma por mais que exista diversos códigos sql no mesmo arquivo se facilitou e agilizou a implementação do *software* visto que separando pelas operações a serem realizadas naquele momento simplifica a manutenção do código ou correção de erros.

### 3.7 IMPLEMENTAÇÃO

A implementação e programação, foi realizada individualmente tendo como ponto de partida a prioridade de cada requisito previamente analisado em paralelo com a geração dos códigos que compõem o banco de dados.

Para o desenvolvimento do sistema, foi necessário um estudo maior sobre a linguagem *JavaScript*, pois esta linguagem foi utilizada para o tratamento de erros, gerando algum retorno ao usuário, além da exportação das questões para um arquivo.doc.

Para agilizar o carregamento de alguns conteúdos e manipulação de outros foi utilizado o *framework JQuery* já que ele oferece alguns recursos, que foram utilizados:

- Seleção e manipulação de elementos HTML;
- Manipulação do CSS e Eventos.

Além das ferramentas já citadas neste capítulo, foi necessária uma maior compreensão quanto ao uso do *plugin CKEditor*, o qual permite um ganho expressivo no tempo de desenvolvimento, já que contém diversas personalizações para o professor, conforme citado no referencial teórico.

### 3.8 ENTREGA DOS *SPRINTS*

Aqui tem-se a descrição de cada entrega realizada em todos os *sprints* separadas por entrega, os resultados, as dificuldades encontradas e outras considerações sobre o desenvolvimento.

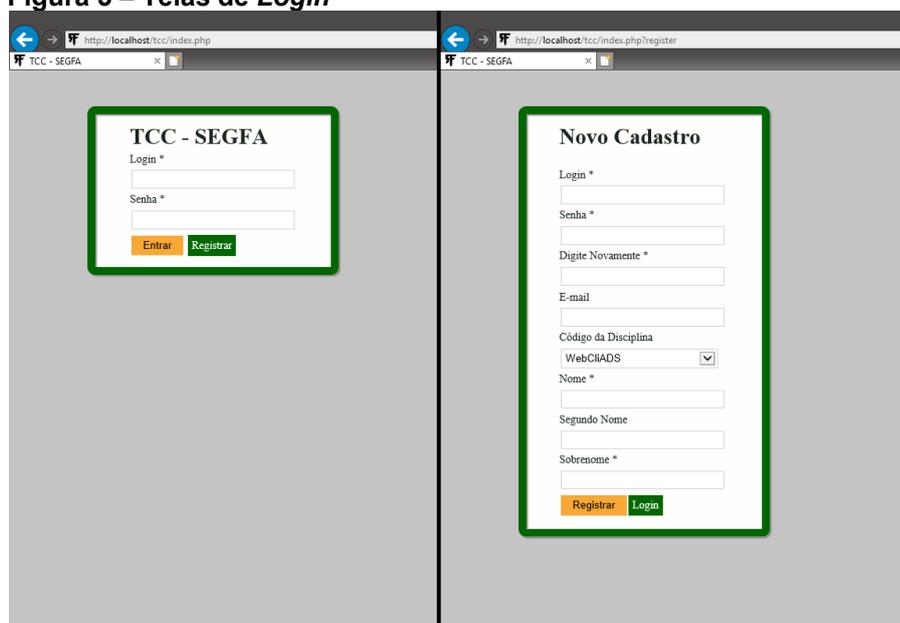
#### 3.8.1 *Login*

O primeiro *sprint* tratava-se da entrega da implementação do *login*. Nessa implementação, foi realizado tanto o *login* por parte do aluno como por parte do professor.

Para agilizar o processo de cadastro do professor, não foi criado um cadastro padrão, utilizando, por exemplo, *login*, *e-mail*, nome, e outros dados padrões de páginas de cadastro. Já para o cadastro do aluno, foi necessário a implementação de um cadastro básico, onde é possível, também, a inserção do código da disciplina na qual o aluno encontra-se matriculado.

Quando o *software* é acessado, é então realizado um redirecionamento direto para a página de *login*. No arquivo *index.php* consta apenas as chamadas dos *scripts* em *JavaScript*, e as verificações do *login*. A Figura 5 ilustra graficamente o *login* e o registro de um aluno.

**Figura 5 – Telas de *Login***



**Fonte: Autoria Própria**

É importante ressaltar que o Código da Disciplina só será exibido ao aluno depois da inserção da mesma e de seu código no *software*. Após o cadastro do aluno, então ocorre um redirecionamento para a tela de *login*, vale-se ressaltar que, a tela de *login* é a mesma para ambos papéis.

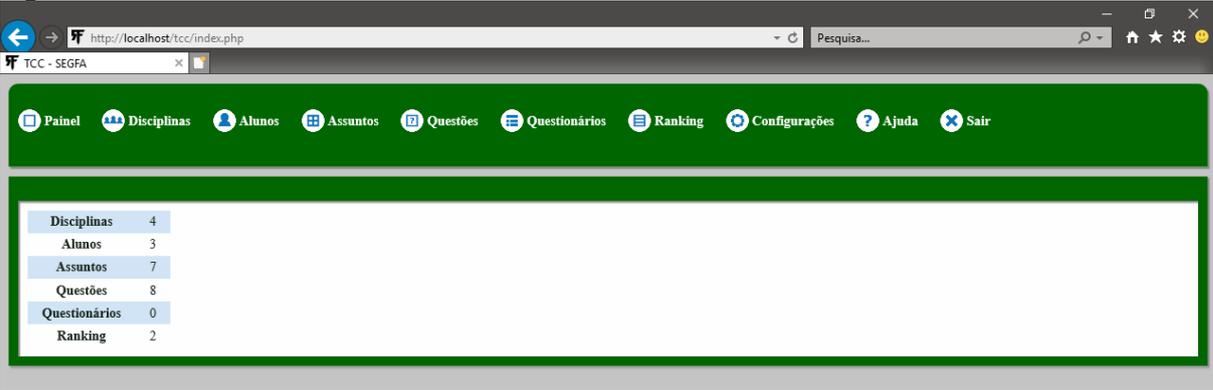
### 3.8.2 Professor

Efetuada com sucesso o *login*, será exibido o painel de administração do sistema, o qual consta uma lista com Disciplinas, Alunos, Assuntos, Questões, Questionário e o *Ranking*.

No menu superior ainda consta mais duas opções uma página de Ajuda e outra com as Configurações, a primeira auxilia o professor a proceder com os primeiros passos dentro do sistema; a segunda disponibiliza ao professor opções de gerar questões randômicas e alterar a senha do aluno.

Existe uma lista que simula um relatório de tudo aquilo que está cadastrado. Por tratar-se apenas da estrutura principal do *software* e não envolver tantas consultas SQL obteve-se o resultado com um tempo menor que o estimado já que essa implementação não exigiu complexidade alta. As características são visíveis através da Figura 6.

**Figura 6 – Painel do Professor**



The screenshot shows a web browser window with the URL `http://localhost/tcc/index.php`. The page title is "TCC - SEGFA". The navigation menu includes: Painel, Disciplinas, Alunos, Assuntos, Questões, Questionários, Ranking, Configurações, Ajuda, and Sair. The main content area displays a summary table:

Disciplinas	4
Alunos	3
Assuntos	7
Questões	8
Questionários	0
Ranking	2

**Fonte: Aatoria Própria**

Essa entrega finaliza o segundo *sprint*, logo, após outra reunião se procedeu à implementação do segundo ponto da lista de tarefas. Vale ressaltar que para criar tanto o *login*, que está presente na seção 3.8.1, quanto o painel do professor, foi necessário a criação de arquivos CSS para facilitar a visualização dos elementos gerando agilidade no desenvolvimento.

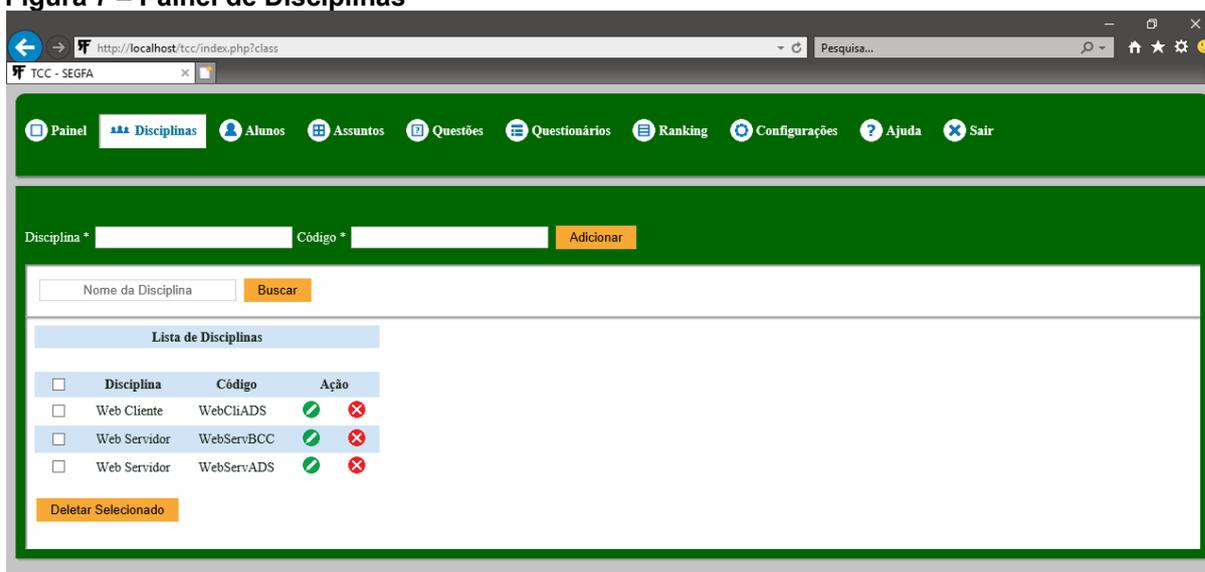
### 3.8.3 Disciplinas e Assuntos

Após o acesso ao sistema, o primeiro passo é inserir uma disciplina com seu respectivo código. Caso não exista nenhuma disciplina, não é possível inserir um assunto ou questão, muito menos questionário, isto ocorre devido as regras inseridas durante a implementação do sistema, as quais não permitem uma questão permanecer vazia, ou seja, não está associada em nenhum assunto.

Sempre priorizando a agilidade no uso da ferramenta por parte do professor, seja para inserção, busca, listagem e exclusão de uma disciplina, estas ações ocorrem na mesma página, evitando carregamento desnecessário de outro *layout* e reaproveitando o CSS de elementos anteriormente utilizados.

De forma ilustrativa, cadastrou-se 3 disciplinas, com códigos genéricos mas que fazem alusão aos cursos de Análise e Desenvolvimento de Sistemas e Ciência da Computação para compor um exemplo próximo da realidade acadêmica. A Figura 7 demonstra o resultado da implementação.

**Figura 7 – Painel de Disciplinas**



**Fonte: Autoria Própria**

Na listagem são exibidos o nome da disciplina, seu código e uma ação a ser tomada, seja ela deleção ou edição representadas por ícones que tendem a melhorar o uso da ferramenta.

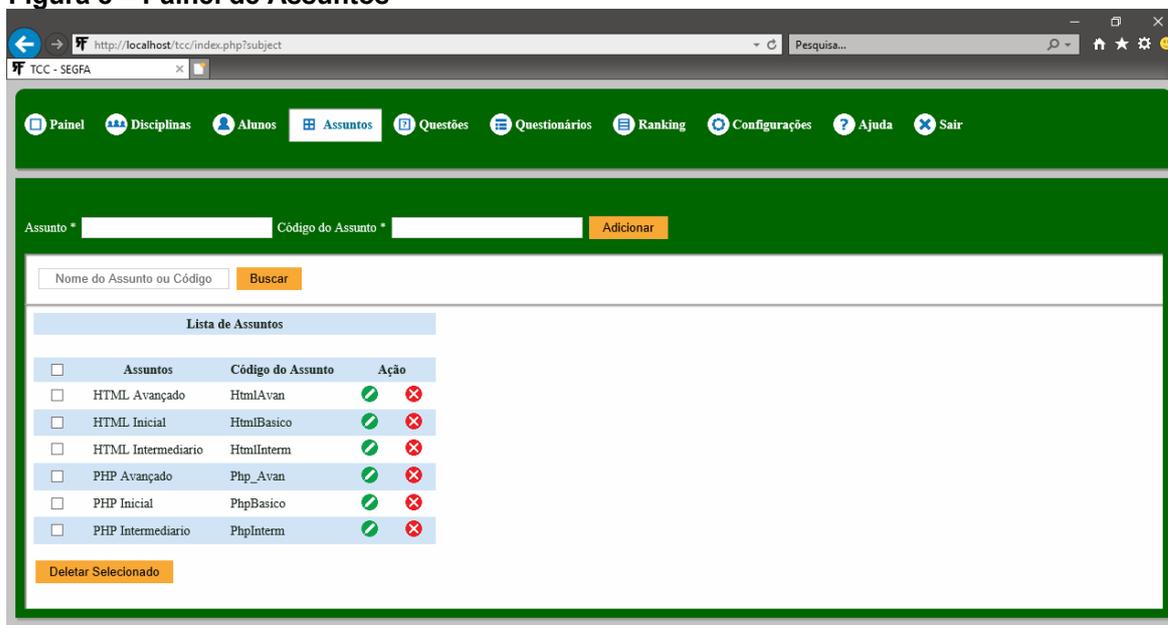
Os botões que correspondem a alguma ação possuem a mesma coloração, como pode-se perceber na Figura 7, o botão verde para edição e o vermelho para exclusão. Dessa forma auxilia-se no processo de aprendizado do *software* pois através da similaridade visual aumenta o aprendizado.

Após o cadastro de uma ou mais disciplinas, é habilitada a possibilidade de acessar a página de Assuntos. Cada disciplina pode possuir um ou mais assuntos, porém em ambos os campos, Código, Disciplina ou Assunto, é único e não poderá ser alterado uma vez cadastrado.

A implementação foi realizada desta forma para evitar uma inconsistência e até mesmo perda no banco de questões que será resultado através da união desses dois códigos

Este código deverá ser definido no ato do cadastro e pode conter letras, números ou símbolos. Visando a reutilização do código a mesma interface é usada para compor a página de Assuntos, a Figura 8 contém o resultado visual.

**Figura 8 – Painel de Assuntos**



**Fonte: Autoria Própria**

É possível realizar uma busca pelos códigos ou nomes dos respectivos assuntos. De tal forma findou-se o segundo *sprint* e conseqüentemente inicia-se o terceiro com a implementação das questões e geração dos questionários.

### 3.8.4 Questões e Questionários

Quando se iniciou a implementação da penúltima parte do sistema o maior desafio encontrado foi a geração de três diferentes tipos de questões, partindo da premissa que após a questão ter sido gerada seria necessário, no futuro, recuperá-la para devolver um *feedback* ao aluno de seu desempenho naquele questionário, bem como compor o *ranking* dos alunos.

Por outro lado, para montar o questionário cada questão deveria pertencer a um único Assunto, mas, uma mesma questão pode ser reaproveitada em outro questionário, um exemplo é o professor que ministra a mesma disciplina para dois cursos diferentes, porém da mesma área do conhecimento.

Isto ocorre em uma disciplina *x* que é ministrada pelo mesmo professor, porém para dois cursos diferentes, como por exemplo, a disciplina de Banco de dados, presente na ementa dos cursos de Bacharelado em Ciência da Computação e Tecnologia em Análise e Desenvolvimento de Sistemas

Optou-se por exibir o tipo da questão e o código da disciplina respectiva para auxiliar no processo de criação do banco de questões pelo professor. Na Figura 9 temos ilustrado o produto da codificação para a tela de questões com cinco questões cadastradas.

**Figura 9 – Painel de Questões**

Questão	Tipo da Questão	Código da Disciplina	Ação
<input type="checkbox"/> Assinale a alternativa correta:	Múltipla Escolha	HtmlBasico	<input checked="" type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Acerca de tecnologias, ferramentas e procedimentos...	Verdadeiro ou Falso	HtmlBasico	<input checked="" type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HTML é uma linguagem de programação...	Verdadeiro ou Falso	HtmlBasico	<input checked="" type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Entre os novos elementos do HTML5, o elemento	Múltipla Escolha	HtmlBasico	<input checked="" type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> PHP é uma linguagem de tipagem forte.	Verdadeiro ou Falso	PhpBasico	<input checked="" type="checkbox"/> <input type="checkbox"/>

**Fonte: Autoria Própria**

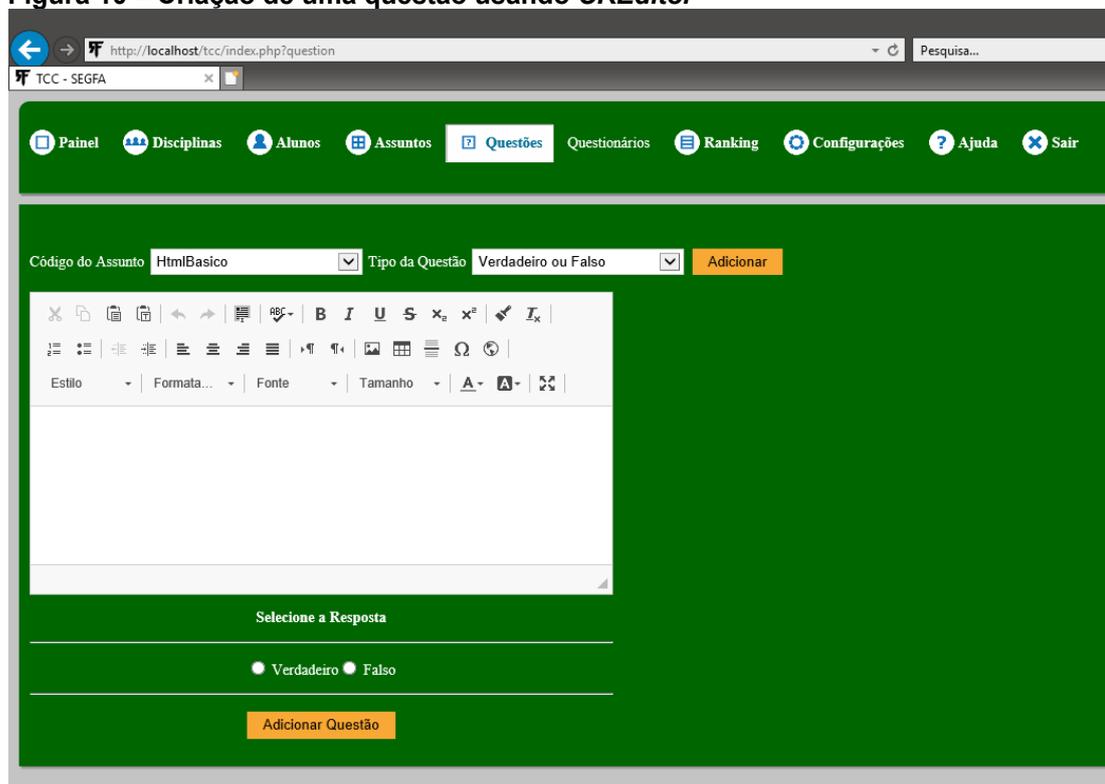
Cada questão pode ser do tipo verdadeiro ou falso, múltipla escolha e descritiva. Na opção de várias alternativas ela podem ter de três até seis alternativas, e sua exibição será por questão em relação ao código da disciplina.

Da mesma maneira como ocorreu em outras páginas do *software* utiliza-se do reaproveitamento dos códigos pelos mesmos motivos citados anteriormente. Sobre a elaboração de cada questão fez-se uso do *plugin CKEditor* para compor a tela de criação.

Seu uso está presente para edição da pergunta de cada questão, gerando possibilidade de personalizar o texto com negrito, itálico, sublinhado, inserir e formatar imagens ou até mesmo uma tabela, por exemplo.

A Figura 10 ilustra o *CKEditor* e a tela para inserção de uma questão do tipo verdadeiro ou falso, com um dos tipos de questão que foram implementados, além do resultado da implementação do *plugin*.

**Figura 10 – Criação de uma questão usando *CKEditor***



**Fonte: Autoria Própria**

O desafio aqui demonstrou-se em realizar a integração entre o *plugin* e o código PHP de modo que ao carregar a página de Questões todo código fosse interpretado e exibido corretamente conforme ilustrado anteriormente.

O *CKEditor* é composto por diversos arquivos que constituem a parte visual, logo as configurações que se mostraram necessárias foram relacionadas a parte lógica do *plugin*. Utilizando-se então de um vetor de elementos foi unido o conteúdo inserido dentro da caixa de edição com as opções de verdadeiro ou falso, múltipla escolha ou a resposta descritiva de cada questão.

Através da imagem vê-se o ícone “Exportar para .doc” é possível exportar as questões em um formato bem similar ao de prova tradicional, isto é, o modelo impresso. De outra forma, se uma questão é múltipla escolha ela aparecerá com alternativas de “a” até “e” no formato do *Microsoft Word*, com um cabeçalho simples.

A Figura 11 a seguir contém 2 exemplos da exportação gerada, vale ressaltar que ao clicar no botão ele exporta e realiza o *download* das questões presentes salvando em um arquivo com extensão.doc.

### Figura 11 – Exportação para Arquivo .doc

Nome: \_\_\_\_\_ Matricula: \_\_\_\_\_  
Data: \_\_\_\_\_

---

1. Assinale a alternativa correta:

- a. CSS é uma linguagem de programação poderosa e requer HTML para seu funcionamento.
- b. Ambos são os conceitos fundamentais aplicados em dados abertos conectados.
- c. HTML é um programa que favorece o desenvolvimento de arquivos CSS.
- d. HTML e CSS são as tecnologias utilizadas na construção de páginas da Web: HTML (html e xhtml) para a estrutura, CSS para o estilo e layout, incluindo WebFonts.
- e. Nenhuma das Anteriores

2. Acerca de tecnologias, ferramentas e procedimentos associados à Internet e à intranet, julgue os itens subsequentes. O hipertexto consiste em um sistema de base de dados que integra eficientemente objetos de diferentes tipos, tais como texto, imagem, vídeos, sons e aplicativos.

- a. Falso
- b. Verdadeiro

**Fonte: Autoria Própria**

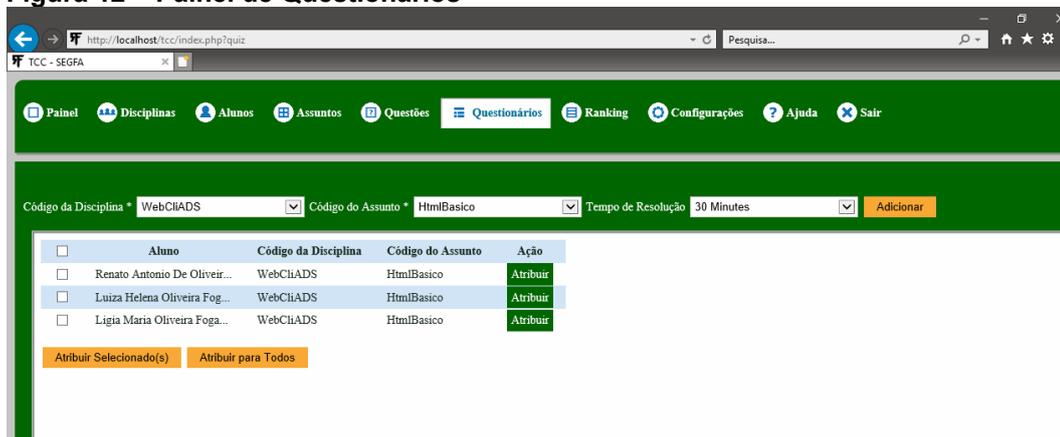
Observa-se que o resultante é bem próximo a uma prova escrita. Como o objetivo é ajudar o professor em relação aos conteúdos que precisam ser elaborados para estimular os alunos tal exportação pode ser usada como prova ou composição da mesma.

A próxima etapa se deu pela união das questões cadastradas em um questionário, este composto pelos códigos de uma das disciplinas e o código do assunto, para que, futuramente, componham o banco de questões do professor.

Atendendo uma das demandas, no que diz respeito a gamificação, no ato de gerar um questionário também se deve configurar o tempo para resolução do mesmo. Cada disciplina poderá possuir um ou mais questionários e é possível a reutilização.

Na Figura 12 está ilustrado o painel de questionário como parte integrante do processo de criação da ferramenta de autoria que também compõe, para módulo do aluno, uma parte essencial do processo de gamificação do *software*.

**Figura 12 – Painel de Questionários**



**Fonte: Autoria Própria**

Após clicar em “Adicionar”, como ilustrado, realiza-se uma busca por todos os alunos associados aquele código da disciplina. Logo é possível atribuir o questionário ao aluno, que, ao acessar o sistema pode perceber que existem questionários não respondidos, finalizando assim a entrega dessa *sprint*.

### 3.8.5 Aluno

De modo sequencial para que seja possível realizar um questionário faz-se necessária existência do mesmo, portanto, o módulo do aluno é descrito nesta seção, embora fosse uma segunda entrega em relação as *sprints* estabelecidas. Exemplificando o painel do aluno, depois do login, o aluno se depara com a mesma página ilustrada na Figura 13 a seguir.

**Figura 13 – Painel do Aluno sem Questionário**



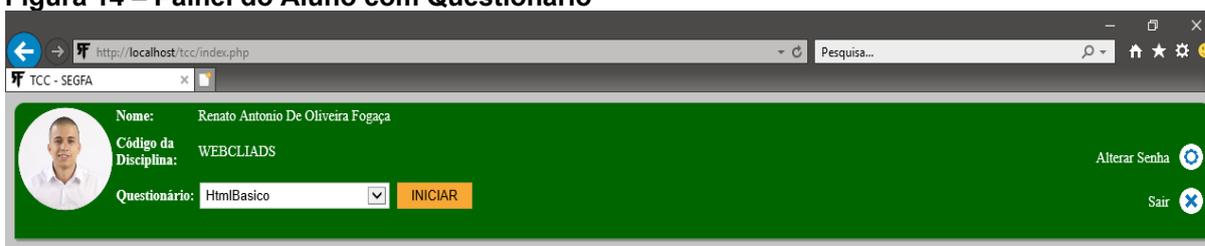
**Fonte: Autoria Própria**

Percebe-se que a imagem do aluno não foi alterada e não está atribuído um questionário a ele. Não existindo nenhuma atribuição uma mensagem é mostrada; é válido ressaltar que um ou mais questionários podem aparecer para o aluno caso professor tenha atribuído.

A alteração da imagem atende um dos critérios de gamificação e o aluno pode definir uma imagem personalizada para ser exibida no seu perfil. A imagem não pode ultrapassar 500kb de tamanho, sendo um tamanho genérico e padrão da web usado apenas por uma questão de sobrecarga em relação ao servidor, mas, esse valor pode ser alterado caso seja necessário.

A Figura 14 ilustra a página do aluno que já fez alteração na sua imagem de perfil e possui um questionário atribuído.

**Figura 14 – Painel do Aluno com Questionário**



**Fonte: Autoria Própria**

Assim que o aluno inicia um questionário aparece para o professor que o aluno está respondendo o questionário, mudando na página de questionários seu status, auxiliando no acompanhamento.

Após clicar em “INICIAR” o aluno é redirecionado para página de respostas do questionário, estes podem ou não possuir um tempo. Se houver um tempo estipulado, 15 minutos por exemplo, implementou-se uma função de pausa, caso o aluno não queira responder todas questões de uma vez.

O contador de tempo é decrescente e é exibido ao aluno, estimulando assim a habilidade de resolver problemas com limitações, neste caso o tempo, esse ponto é outro critério de gamificação aplicado no *software*.

A contabilização do tempo foi implementada baseando-se nas divisões simplificadas de uma hora em quartas partes, sendo o menor tempo quinze minutos, correspondente a um quarto de hora, e o maior quatro horas.

Simulando linhas de um caderno usou-se duas imagens repetidas criando uma interface mais amigável ao aluno que estará respondendo. Ambas informações podem se perceber na Figura 15:

**Figura 15 – Página de Resposta do Questionário**

Nome: Isis Luise Oliveira Fogaça 00:07:42

Código da Disciplina: WEBCLIADS

Questionário: HtmlBasico

1. Assinale a alternativa correta:

Nenhuma das Anteriores

HTML e CSS são as tecnologias utilizadas na construção de páginas da Web: HTML (html e xhtml) para a estrutura, CSS para o estilo e layout, incluindo WebFonts.

CSS é uma linguagem de programação poderosa e requer HTML para seu funcionamento.

HTML é um programa que favorece o desenvolvimento de arquivos CSS.

Ambos são os conceitos fundamentais aplicados em dados abertos conectados.

2. O HTML é uma linguagem de Programação?

Verdadeiro

Falso

3. Um usuário de um computador com Windows 7 está utilizando o navegador Google Chrome, versão 65, para realizar a busca por informações sobre cursos de informática. Ao localizar uma página com informações interessantes, o usuário decide por salvar a página. Nessa situação, um dos tipos de salvamento da página é: Página da web,

somente texto.

formato impressão.

formato PDF.

Enviar seu questionário

**Fonte: Autoria Própria**

Outra informação importante é que cada vez que o aluno pausa e retorna às questões ou inicia, uma função randômica mistura tanto as alternativas de cada questão quanto a ordem das questões.

Atendendo à mais 2 critérios de gamificação tem-se a página de Resultados e Ranking, na qual, após respondido um ou mais questionários tem-se o *feedback*, como primeiro critério aplicado, e o *ranking*, como segundo critério.

O primeiro resultado corresponde apenas a relação de quantidades de questões, as respostas certas e a porcentagem em relação ao total. Melhorando a resposta ao aluno implementou-se um *feedback* mais elaborado que ao clicar em “Revisão” exibe o mesmo questionário realizado anteriormente pelo aluno, porém com a descrição de cada questão como correta ou incorreta.

Ressalta-se que o *software* permite mais de um questionário por disciplina, basta que o professor no ato de criação das questões designar corretamente o assunto ao qual ele pertence. Os resultados e o *feedback* são ilustrados na Figura 16 a seguir.

**Figura 16 – Painel de Ranking e Resultados do Aluno**

The screenshot shows a web browser window with the URL 'http://localhost/tcc/index.php?score'. The page title is 'TCC - SEGFA'. The user profile is for 'Luiza Helena Oliveira Fogaça' with discipline code 'WEBCLIADS' and a status of 'Ruim'. There are three navigation links: 'Resultados/Ranking', 'Alterar Senha', and 'Sair'. Below the profile are two tables:

Resultados e Revisão						
Código do Assunto	Qtd Questões	Respostas Corretas	Porcentagem	Status	Ação	Realizado em:
HtmlBasico	5	1	20	Ruim	Revisão	2018-10-23

Ranking							
Aluno	Código da Disciplina	Código Assunto	Qtd Questões	Resposta Correta	Porcentagem	Status	Realizado em
Renato Antonio De Oliveir...	WebCliADS	HtmlBasico	4	3	75	Ótimo	2018-10-23
Luiza Helena Oliveira Fog...	WebCliADS	HtmlBasico	5	1	20	Ruim	2018-10-23

**Fonte: Autoria Própria**

No canto direito da imagem temos três opções para o aluno, verificar os resultados das questões, alterar sua senha e sair do sistema. De modo simples ao clicar em sair ele somente será desconectado, retornando para tela de login, ao clicar em “Alterar Senha” será direcionado à página correspondente a tal ação.

Se o aluno clicar em “Revisão” o mesmo questionário é carregado, porém com distinção das alternativas corretas e incorretas, aonde é informado ao aluno em vermelho as questões erradas e em verdes as corretas. Dessa forma o *feedback* ao aluno é mais preciso pois auxilia no aprendizado já que ele pode aferir quais questões ele tem conhecimento e quais ele ficou em dúvida.

Como o *layout* é o mesmo torna-se simplificada a verificação, bem como auxilia na fixação do conteúdo referente ao assunto abordado no questionário. Já que trabalha com elementos de memorização através da similaridade das telas e, o aluno, facilmente consegue verificar a questão e sua resposta correta.

Uma possibilidade discutida no decorrer da implementação seria imputar a resolução para cada resposta, porém não será tratado neste trabalho. Através da figura 17 ilustrou-se a resolução de duas questões respondidas de forma incorreta durante a simulação de um teste.

**Figura 17 – Página de Revisão do Questionário**

The screenshot displays a web interface for reviewing questionnaire results. It features two questions, each with a text input field for the user's answer and a list of radio button options. The first question asks if HTML is a programming language, with 'Falso' (False) selected. The second question asks to select the correct alternative, with 'HTML e CSS são as tecnologias utilizadas na construção de páginas da Web...' selected. Both questions are marked as 'Incorreta' (Incorrect) in red text, with the correct answer provided below each question.

2. O HTML é uma linguagem de Programação?

Sua resposta foi:

Falso  
 Verdadeiro

A resposta está: **Incorreta**. Reposta Correta: verdadeiro

3. Assinale a alternativa correta:

Sua resposta foi:

Nenhuma das Anteriores  
 CSS é uma linguagem de programação poderosa e requer HTML para seu funcionamento.  
 HTML é um programa que favorece o desenvolvimento de arquivos CSS.  
 HTML e CSS são as tecnologias utilizadas na construção de páginas da Web: HTML (html e xhtml) para a estrutura, CSS para o estilo e layout, incluindo WebFonts.  
 Ambos são os conceitos fundamentais aplicados em dados abertos conectados.

A resposta está: **Incorreta**. Reposta Correta: html é um programa que favorece o desenvolvimento de arquivos css.

**Fonte: Autoria Própria**

Para que fosse possível realizar o retorno das respostas ao aluno, para cada questão criada o sistema armazena no banco de dados sua resposta correta. Após o questionário ser respondido o *software* realiza uma verificação daquilo que foi respondido em relação ao que está armazenado e salva em variáveis temporárias para exibir ao aluno.

Julgou-se interessante fornecer um *feedback* ao docente do desempenho de seus alunos, para tanto, no painel do professor existe uma opção nomeado como “*Ranking*” onde ele pode aferir o desempenho resumido de cada aluno e quem está com maior nota, bem como quantidade de questões acertadas e de qual disciplina ele pertence

O diferencial deste *ranking* é a existência da coluna “Ação” o docente pode realizar a exclusão do questionário, de tal forma que deixa de constar ao aluno, tanto para *feedback* quanto para *ranking*.

Justifica-se essa opção porque, por exemplo, no decorrer do semestre podem existir mais questionários, logo, outros *rankings* são gerados. As essas opções citadas estão visíveis na Figura 18 ilustrada na sequência.

**Figura 18 – Painel de *Ranking* e Resultados do Aluno para Professor**

Ranking									
<input type="checkbox"/>	Aluno	Código da Disciplina	Código Assunto	Qtd Questões	Resposta Correta	Porcentagem	Status	Realizado em	Ação
<input type="checkbox"/>	Renato Antonio De Oliveir...	WebCliADS	HtmlBasico	4	3	75	Ótimo	2018-10-23	<input type="checkbox"/>
<input type="checkbox"/>	Luiza Helena Oliveira Fog...	WebCliADS	HtmlBasico	5	1	20	Ruim	2018-10-23	<input type="checkbox"/>

Deletar Selecionado

**Fonte: Autoria Própria**

Um dos motivos dessa implementação é permitir ao professor, caso seja possível, bonificar os alunos com melhor desempenho, utilizando-se de outro questionário mais desafiador como recompensa, por exemplo. Esse tipo de recompensa constitui uma técnica de gamificação, porém não foi contabilizada na implementação por ter dependência com uma ação do professor.

Dessa forma findou-se a entrega da última *sprint*, atendendo as solicitações do *product owner* em relação aos requisitos. Encerrou também a entrega o *software* dentro daquilo que foi definido, alcançou-se os objetivos definidos e foi entregue na data de entrega prevista anteriormente.

## 4 CONSIDERAÇÕES FINAIS

O trabalho apresentado foi uma ideia que surgiu através da leitura e análise da tese de mestrado e doutorado, parcialmente, da orientadora, se definiu conforme necessidade o escopo do projeto.

Através da implementação de cada um dos módulos, o objetivo do desenvolvimento do software educacional foi alcançado. A principal dificuldade foi abstrair todos os requisitos necessários para composição de cada parte constituinte e em seguida aplicá-la.

Aplicou-se os conceitos de gamificação em relação ao *feedback* dado ao aluno após a realização do questionário, a inclusão do tempo para resolução das questões e a geração do *ranking* dos melhores resultados e a personalização da imagem de perfil do aluno.

O *XAMPP* possibilitou a execução dos códigos em PHP, e, através do *phpmyadmin*, permitiu a implementação do banco, sem nenhum por apresentar uma interface de fácil interação aos usuários e aceitar a execução de *scripts*, para criar um tratamento de erro por exemplo, dentro dessa interface.

O *Microsoft Visio* facilitou muito a modelagem dos diagramas tanto o modelo relacional quanto o diagrama de classes, pois para o primeiro ele relaciona os elementos já presentes e o banco, permitindo que se trabalhe visualmente com a modelagem das tabelas, atributos e relações.

Uma possível aplicação deste projeto seria para instituições que precisem resolver questionários *on-line* sendo necessário a existência do tempo de resolução e ao findar retorne um *feedback* ao participante. Um exemplo de aplicação seria no Projeto de Inclusão Informacional da Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa, pois como os alunos participam de apenas uma turma facilmente os instrutores poderiam construir questões e disponibilizar questionários aos alunos.

Embora existam plataformas com funcionalidades iguais ou parecidas com os presentes no *software*, como a integração entre assunto, disciplina e professor, ou utilização de questionário, o *Moodle* é um exemplo. Porém o *Moodle*, tomando como exemplo específico, é exigido maior conhecimento do docente em relacionado à plataforma de ensino, além da configuração prévia de todo o sistema.

Os diferenciais do *software* são a facilidade na construção das questões pelos professores, tomando como base um docente que não tenha contato com outras plataformas e ferramentas de autoria, a possibilidade de gerar questões para o formato .doc, usando a mesma formatação utilizada no programa, e a possibilidade de o professor utilizar o mesmo banco de questões para diferentes disciplinas.

Finalmente pode-se afirmar que este projeto contribui no aprendizado sobre softwares educacionais gamificados. Possibilitou uma maior compreensão dos reais desafios existentes quando é necessário deixar uma plataforma gamificada e desafios em elaborar uma ferramenta de autoria que pudesse proporcionar mais de um tipo de questão ao professor.

O trabalho permitiu ganhar conhecimento necessário para o desenvolvimento de novos projetos ágeis ou não, preparando melhor para o mercado de trabalho, garantindo assim um futuro profissional na área de análise e desenvolvimento de sistemas.

#### 4.1 TRABALHOS FUTUROS

São indicados como trabalhos futuros a implementação da participação do aluno em mais de uma disciplina, refatorar o código tornando o aplicativo móvel no módulo aluno, aplicação de um framework para melhoria da interface para todos os módulos do programa.

Além desta melhoria outros tipos de questões poderiam ser implementados como questões de somatória, auto complete entre outras. Outro ponto é a aplicação dos testes de usabilidade e aceitação tanto para os alunos quanto para professores que poderão vir a utilizar.

Como consta na seção de 2.5 deste trabalho, existem outras ferramentas de autoria, um dos tipos dessas ferramentas a plataforma *Moodle* faz uso; logo um possível trabalho seria adaptar o *software* tornando-o um possível *plugin* para esta plataforma por alguns motivos:

- O *Moodle* também utiliza a linguagem PHP para tratar da sua lógica;
- Também faz uso do CCS e HTML;
- Alguns eventos são manipulados através do *JavaScript*.

Uma justificativa para tornar o *software* em um *plugin* seria o fato de a plataforma não permitir que sejam reutilizadas as mesmas questões para disciplinas diferentes, onerando ao professor que ele insira mais de 1 banco de questões, mesmo as disciplinas possuindo assuntos iguais e/ou semelhantes.

Outra possibilidade também realizar uma melhoria possibilitando a exportação das questões para um formato que o Moodle ou demais plataformas possa reutilizar, como formato *Json* ou *XML*.

Como melhoria no *feedback* seria possível disponibilizar após a criação de cada questão a inserção da resolução da questão, evitando que seja apenas um retorno com “Correto” ou “Incorreto”, mas que, no caso de uma questão da disciplina de Matemática seja fornecida a resolução da questão, por exemplo.

## REFERÊNCIAS

ALVES, L.R.A. **Game Over: jogos eletrônicos e violência**. Salvador: UFBA, 2004. Disponível em: < [http://www.comunidadesvirtuais.pro.br/game-studies/files/gs\\_submission/trabalho\\_27/trabalho\\_27.pdf](http://www.comunidadesvirtuais.pro.br/game-studies/files/gs_submission/trabalho_27/trabalho_27.pdf) >. Acesso em: 07 ago. 2018.

BALASUBRAMANIAN, N; WILSON, B. G. Games and simulations. In: SOCIETY FOR INFORMATION TECHNOLOGY AND TEACHER EDUCATION INTERNATIONAL CONFERENCE, 17., 2006, Estados Unidos. **Anais eletrônicos**. Estados Unidos, 2006 Disponível em:< <http://homepages.utoledo.edu/jlamber4/5550/GamesAndSimulations.pdf> >. Acesso em:12 ago. 2018.

BATTAIOLA, A. Jogos por computador – Histórico, Relevância Tecnológica e Mercadológica, Técnicas de Implementação. In: XIX JORNADA DE ATUALIZAÇÃO EM INFORMATICA, 2., 2000, Curitiba. **Anais eletrônicos**. Curitiba, 2000. Disponível em:< <http://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr031.pdf>>. Acesso em: 12 ago. 2018.

BATTAIOLA, A. L.; ELIAS, N. C.; DOMINGUES, R. G. Desenvolvimento de um software educacional com base em conceitos de jogos de computador. In. XII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1., 2002. **Anais eletrônicos**. 2002. Disponível em:< <http://www.br-ie.org/pub/index.php/sbie/article/view/189/175> >. Acesso em: 12 ago. 2018

CASTRO, Elizabeth; HYSLOP, Bruce. **HTML5 e CSS3**. 7. ed. Rio de Janeiro: Alta Books, 2013. 552 p.

CKEDITOR. **Ecosystem Documentation**. 2018. Disponível em: <<https://ckeditor.com/docs/>>. Acesso em: 01 out. 2018.

CONVERSE, Tim; PARK, Joyce. **PHP: a bíblia**. 2. ed. Rio de Janeiro: Gulf Professional Publishing, 2003. 868 p.

DUCKETT, Jon. **Javascript e jquery**. Desenvolvimento de Interfaces Web Interativas. 1. ed. Elsevier/Alta Books, 2013. 640 p.

DETERDING, S.; KHALED, R.; NACKE, L. E. **Gamification**: Toward a Definition. Vancouver, 2011. Disponível em: < <http://gamification-research.org/wp-content/uploads/2011/04/02-Deterding-Khaled-Nacke-Dixon.pdf> >. Acesso em: 02 set. 2018.

FALKEMBACH, G. A. M. **O lúdico e os jogos educacionais**. Rio Grande do Sul, UFRGS, 2013. Disponível em:< [http://penta3.ufrgs.br/midiasedu/modulo13/etapa1/leituras/arquivos/Leitura\\_1.pdf](http://penta3.ufrgs.br/midiasedu/modulo13/etapa1/leituras/arquivos/Leitura_1.pdf)>. Acesso em: 15 ago. 2018.

GROH, Fabian. **Gamification**: State of the art definition and utilization. 2012. Disponível em:< [http://hubscher.org/roland/courses/hf765/readings/Groh\\_2012.pdf](http://hubscher.org/roland/courses/hf765/readings/Groh_2012.pdf) >. Acesso em: 03 set. 2018.

KAPP, Karl M. **The Gamification of Learning and Instruction**: Game-based Methods and Strategies for Training and Education. 1 ed. New Jersey:Pfeiffer & Company, 2012.

KISHIMOTO, Tisuko Morchida. **Jogo, brinquedo, brincadeira e a educação**. 7ª edição. São Paulo: Cortez, 2003.

LEE, H.; DOH, Y. Y. **A Study on the relationship between educational achievement and emotional engagement in a gameful interface for video lecture systems**. In: INTERNATIONAL SYMPOSIUM ON UBIQUITOUS VIRTUAL REALITY, IEEE, 2012, p 34-37.

MATTAR, J. **Games em educação**: como os nativos digitais aprendem. 1 ed. São Paulo: Pearson, 2010.

MICROSOFT VISIO. **Software de fluxogramas e diagramação**. Disponível em: <<https://products.office.com/pt-br/visio/flowchart-software>>. Acesso em: 02 out. 2018.

MURRAY, T.; BLESSING, S.; AINSWORTH, S. **Authoring Tools for Advanced Technology Learning Environments**: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software. 1 ed. New York: Springer, 2003.

MURRAY, T.; BLESSING, S.; AINSWORTH, S., **Authoring Tools For Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive And Intelligent Educational Software**. 2 ed. New York: Springer Science & Business Media, 2013.

PEELING, Nick. **Gamification**. 2018. Disponível em: <<http://www.nickpelling.com/>>. Acesso em: 01 out. 2018.

PHP:DOCUMENTATION. **Documentation**. Disponível em: <<http://php.net/docs.php>>. Acesso em: 25 out. 2018.

PIETRUCHINSKI, Mônica Hoeldtke. **ArqMAEC: um modelo arquitetural baseado em Agentes para monitorar, avaliar e estimular a colaboração em ambientes educacionais gamificados**. 2016. 361 p. Tese (doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. São Paulo: Pearson Makron Books, 2011.

RACHA CUCA. **Quebra-cabeças, Problemas de lógica, Testes e Quizzes**. 2018. Disponível em: <<https://rachacuca.com.br/>>. Acesso em: 01 out. 2018

REIS, P. e DAMIÃO, I., **Ferramentas de autoria para a criação de E-conteúdos - Experiência atual. Cibertextualidades**. In: CENTRO DE ESTUDOS CULTURAIS DA LINGUAGEM E DO COMPORTAMENTO. Porto (UFP), 2011, p 1-17.

RESIG, John. **Pro JavaScript Techniques**. Estados Unidos da América: Apress, 2006. 359 p. Disponível em: <<https://doc.lagout.org/programming/JavaScript/Pro%20JavaScript%20Techniques.pdf>>. Acesso em: 02 out. 2018.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. 3 ed. Rio de Janeiro: Casa do Código, 2014.

SCRUM ALLIANCE. **Professional Scrum and Agile Training**. 2018. Disponível em: <<https://www.scrumalliance.org/>>. Acesso em: 01 out. 2018.

SOFFA, M. M. e ALCÂNTARA, P. R. d. C. **O uso do software educativo: reflexões da prática docente na sala informatizada.** In: CONGRESSO NACIONAL DE EDUCACAO (EDUCERE), Curitiba: PUC-PR, 2008, p. 1-13.

SOMMERVILLE, I. **Engenharia de Software.** 6. ed. São Paulo: Pearson Education do Brasil, 2003.

VIANNA, Y.; VIANNA, M.; MEDINA, B.; TANAKA, S. **Gamification, Inc.:** como reinventar empresas a partir de jogos. Rio de Janeiro: MJV Press, 2013.

XAMPP. **Apache + MariaDB + PHP + Perl.** 2018. Disponível em: <[https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)>. Acesso em: 01 out. 2018.

WYSIWYG **Web Builder.** 2018. Disponível em: <<https://www.wysiwygwebbuilder.com/>>. Acesso em: 03 out. 2018.

ZICHERMANN, G.; CUNNINGHAM, C. **Gamification by Design.** Implementing Game Mechanics in Web and Mobile Apps. Canada: O'Reilly Media, 2011.