Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

Gamificação aplicada à educação: Ferramenta de apoio ao ensino e aprendizagem de programação.

Autor: Eduardo Júnio Veloso Rodrigues
Orientador: Dr. Wander Cleber Maria Pereira da Silva

Brasília, DF
2020

Eduardo Júnio Veloso Rodrigues

Gamificação aplicada à educação: Ferramenta de apoio ao ensino e aprendizagem de programação.

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

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dr. Wander Cleber Maria Pereira da Silva

Brasília, DF

2020

Eduardo Júnio Veloso Rodrigues

Gamificação aplicada à educação: Ferramenta de apoio ao ensino e aprendizagem de programação./ Eduardo Júnio Veloso Rodrigues. – Brasília, DF, 2020-

60 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Wander Cleber Maria Pereira da Silva

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2020.

1. Palavra-chave01. 2. Palavra-chave02. I. Dr. Wander Cleber Maria Pereira da Silva. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Gamificação aplicada à educação: Ferramenta de apoio ao ensino e aprendizagem de programação.

CDU 02:141:005.6

Eduardo Júnio Veloso Rodrigues

Gamificação aplicada à educação: Ferramenta de apoio ao ensino e aprendizagem de programação.

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

Trabalho aprovado. Brasília, DF, 03 de Julho de 2020 – Data da aprovação do trabalho:

Dr. Wander Cleber Maria Pereira da Silva
Orientador

Titulação e Nome do Professor Convidado 01
Convidado 1

Titulação e Nome do Professor Convidado 02
Convidado 2

Brasília, DF
2020

A dedicatória é opcional. Caso não deseje uma, deixar todo este arquivo em branco.

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (sem negritos, aspas ou itálico.*

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

A epígrafe é opcional. Caso não deseje uma, deixe todo este arquivo em branco.

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)

Resumo

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. O texto pode conter no mínimo 150 e no máximo 500 palavras, é aconselhável que sejam utilizadas 200 palavras. E não se separa o texto do resumo em parágrafos.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Contexto

A tecnologia tem vindo a evoluir rapidamente e, com isto, nota-se o surgimento de diversos desafios. Um destes desafios é a falta de programadores que possuam competências e qualificação necessárias para solucionar os mais diversos problemas presentes nos mais variados projetos. Este fato está relacionado com a metodologia adotada no ensino de programação e também com a falta de motivação por parte dos estudantes (PEREIRA; COSTA; APARICIO, 2017). Dai, Zhao e Chen (2010) afirmam que, uma das razões de os alunos não absorverem eficientemente os conceitos relacionados à programação se dá pela falta de concentração e motivação dos mesmos frente a exposição destes conteúdos na forma tradicional.

Thais et al. (2002) destaca que a aprendizagem dos conceitos e mecanismos envolvidos na construção de programas não é trivial, uma vez que requer a utilização de raciocínio na sua forma mais abstrata. Um dos problemas mais comuns segundo os autores são: dificuldades no entendimento de comandos, sintaxe dos comandos, dificuldades em entender os resultados da execução de um determinado comando pela máquina, dificuldades em dar os primeiros passos relativos ao estudo de programação entre outros.

Almeida et al. (2019) diz que, em geral, os alunos têm grandes dificuldades em compreender e aplicar os conceitos relativos à programação . Uma das grandes dificuldades está relacionada a problemas de compreensão e aplicação de noções básicas, como por exemplo o uso de estruturas de controle e estruturas condicionais.

Dificuldades como estas apresentadas, encorajam o desenvolvimento de soluções que auxiliem no ensino e aprendizagem de programação de forma diferente ao atual modelo de ensino. Diversas abordagens de ensino são estudadas para facilitar o aprendizado dos alunos, algumas delas são: gamificação, programação imperativa, programação funcional e etc. Neste trabalho, é abordado o uso da gamificação na construção de uma ferramenta de apoio ao ensino e aprendizagem de programação desenvolvida com base nos requisitos identificados a partir da interação com ex alunos da disciplina de Algoritmos de Programação de Computadores, ofertada pela Universidade de Brasília, campus Gama.

Segundo BARATA et al. (2013), jogos bem projetados representam bons motivadores, uma vez que passa a sensação de satisfação e recompensa fazendo com que os jogadores persistam e fiquem engajados em realizar sua missões. Neste contexto, este poder motivacional dos jogos, passou a ser utilizado em outros contextos que não estão

relacionados diretamente aos jogos, uma prática conhecida atualmente como Gamificação do inglês *gamification* .

Para Deterding et al. (2011), o termo gamificação pode ser definido como a utilização de elementos e mecânica de jogos em contextos não relacionados a jogos. De acordo com Baruque (2015) a utilização destes elementos tornam tarefas reais em atividades mais atrativas e lúdicas e, consequentemente, aumentam a motivação e engajamento. Há uma grande variedade de ambientes que possuem elementos semelhantes a características de jogos, muitos deles contendo: sistema de pontuação, feedbacks constantes e etc (BARATA et al., 2013). São exemplos de ambientes com características semelhantes a de jogos: Uri, Datacamp, Edx entre outras.

A aprendizagem baseada na gamificação, preocupa-se em utilizar de mecanismos de jogos não para o entretenimento, mas para o ensino. Os interessados no campo da gamificação trabalham para identificar o cenário e as condições que possam apoiar a integração de jogos aos ambientes de aprendizado. Vários cientistas e estudiosos no campo da gamificação apontaram uma diversidade de elementos de jogos que permitem que eles sejam utilizados como ferramentas de apoio ao aprendizado. Por exemplo: os jogos são bastante envolventes (DICKEY, 2005) e motivadores (PRENSKY, 2003). Além destas características, jogos são excelentes fontes para se adquirir experiência que são dificeis de serem fornecidas por meio de instruções tradicionais (ARENA; SCHWARTZ, 2014).

Os ambientes online gamificados de apoio ao ensino podem fornecer diversas ferramentas, entre elas: classificações, batalhas, fórums de discussões e etc. De forma a incentivar os usuários a participarem das atividades propostas. Durante as competições e batalhas, os estudantes têm a possibilidade de aprender com outros jogadores e comparar suas habilidades, tornando o aprendizado mais prazeroso (COMBÉFIS; BERESNEVIČIUS; DAGIENĖ, 2016).

1.2 Justificativa

De acordo com Souza (2009), a abordagem de ensino tradicional de programação, que é aquela onde o professor apresenta uma série de conceitos aos alunos e os mesmos têm a tarefa de entender como se aplicam na resolução de problemas, para a maioria dos estudantes, se revelam muito abstratas.

Em seu trabalho de conclusão de curso pela Universidade de Brasília , campus Gama (FGA), Calixto (2016) apresenta uma pesquisa baseada nos dados referentes aos índices de aprovação, trancamento e reprovação dos alunos na disciplina de computação básica (atualmente a disciplina recebe o nome de Algoritmos e Programação de Computadores, ofertada no campus Gama).

Os dados utilizados por Calixto (2016), são referentes a um total de 60 turmas e 3286 alunos matrículados nas disciplinas de Computação Básica entre os anos de 2009 e 2013. Segundo o autor, dos 3286 alunos matriculados, 1659 alunos (50,48%) foram reprovados, 262 alunos (8%) trancaram a disciplina e apenas 1364 alunos (41,5%) obtiveram aprovação. O autor ainda explica que esses resultados, quando comparados à média nacional de aprovação de alunos em disciplinas de programação são bem preocupantes, a taxa apresentada como média nacional é de cerca de 67% de aprovação, ou seja, a taxa de aprovação na FGA, é mais de 30% menor em relação a média apresentada (67%).

Contando com a cooperação dos funcionários da secretaria da FGA, obteve-se dados atuais referentes às aprovações, reprovações e trancamentos dos alunos entre os períodos 2017/1 e 2019/1. Ao analisar estes dados, notou-se que do total de alunos matriculados (2225 alunos divididos em 37 turmas) no decorrer de 5 semestres, 890 alunos (40%) foram reprovados, 112 alunos realizaram o trancamento (5%) e apenas 1223 alunos foram aprovados (54,96%). Estes dados são preocupantes e demonstram a atual situação do aprendizado dos alunos na disciplina de Algoritmos e Programação de Computadores na FGA.

Um estudo realizado pela Universidade Federal da Paraíba (UFPB) que analisou por seis períodos acadêmicos os índices de reprovação na disciplina de Introdução à programação, apontou que em nenhum dos seis períodos analisados houvera um índice de aprovação superior a 34%. Além disso, os índices de reprovação e trancamento da disciplina giraram em torno dos 64% e 6% respectivamente (DANTAS, 2016).

Um outro estudo, realizado na Faculdade de Educação Tecnológica do Estado do Rio de Janeiro (FAETERJ-Paracambi) por Vieira, Junior e Vieira (2015), envolvendo 663 alunos da disciplina de Algoritmos 1 mostrou que, destes alunos, apenas 511 cursaram a disciplina até o final, ou seja, cerca de 152 alunos desistiram da disciplina. Do total restante (511), apenas 30% foram diretamente aprovados (cerca de 153 alunos), 40% foram reprovados diretamente (204 alunos) e 30% foram para o exame final (recuperação,

153). Dos 153 alunos que foram para a recuperação, apenas 55% foram aprovados (84 alunos), nota-se então que, dos 663 alunos que foram matriculados na disciplina, apenas 288 alunos foram aprovados ao seu término, cerca de 43%.

1.3 Problema

Como desenvolver uma ferramenta gamificada para apoiar o ensino e aprendizagem de programação.

1.4 Objetivos

Desenvolver uma ferramenta gamificada para apoiar o ensino e aprendizagem de programação.

2 Referencial teórico

2.1 Aprendizagem de programação

De acordo com Koliver, Dorneles e Casa (2004), disciplinas introdutórias de programação são, em sua maioria, problemáticas e costumam apresentar altos índices de desistência e reprovações. Um dos motivos para tal ocorrência se dá pela falta de preparo que se espera que alunos ingressantes nestas disciplinas possuam.

Disciplina de algoritmos são normalmente oferecidas no início da grade curricular dos cursos. Isso faz com que a maioria dos alunos cursantes sejam calouros que ainda estão acostumados com a forma "mecanisada" de ensino que os habituam a somente aplicar fórmulas sem qualquer tipo de análise mais profunda dos problemas (KOLIVER; DORNELES; CASA, 2004).

De acordo com Borges (2019), o modo tradicional de ensino (figura 1) não é suficiente para motivar os alunos a se interessarem por disciplinas de programação. Não fica claro para a maioria dos alunos, principalmente para aqueles não possuem nenhum tipo de conhecimento em informática, a importância de certos conteúdos.

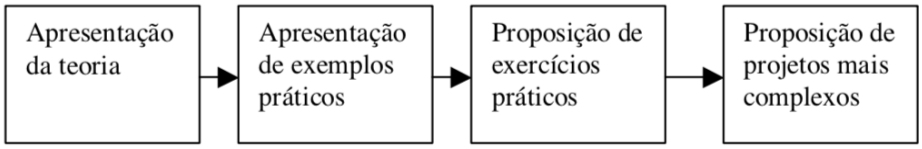

Figura 1 – Sequência de passos típicos na apresentação de uma disciplina

Fonte: (BORGES, 2019)

Ao se apresentar uma nova liguagem de programação, é comum as aulas serem realizadas em laboratórios com recursos computacionais. Apesar dessas aulas apresentarem um formato diferenciado em relação ao modo tradicional, os professores não exploram a diversidade dos equipamentos disponíveis com práticas que apoiem o desenvolvimento de habilidades por parte dos estudantes (BORGES, 2019).

2.2 Gamificação

Jogos são uma construção humana que envolvem em seu contexto fatores sociais, culturais e econômicos (TOLOMEI, 2017). Como apresentado no livro "Gamificação na Educação" por Silva et al. (2014), a interação com *games*, apesar dos custos altos dos consoles, foram ocupando cada vez mais o tempo das pessoas que notaram que os jogos poderiam ser boas fontes de prazer e entretenimento.

O notável crescimento do mercado dos *games* tem atraido diferentes olhares de estudiosos que se dedicam ao estudo de seu uso na educação, comunicação, marketing, psicologia, computação, entre outras áreas (SILVA et al., 2014).

O termo gamificação, segundo Silva et al. (2014), consiste na utilização de elementos de jogos em atividades que por natureza de sua criação não são jogos. Raposo e Dantas (2016) dizem que o objetivo da gamificação, consiste em resolver problemas práticos ou despertar interesse e engajar um público específico para a realização de uma determinada atividade.

Para Chou (2017), a gamificação é uma arte que é capaz de derivar elementos divertidos e envolventes encontrados em jogos e utiliza-los em outras atividades. Para o autor, o foco deve estar centrado no ser humano e na sua motivação.

2.2.1 Gamificação na Educação

Embora o termo gamificação tenha sido apresentado pela primeira vez em 2010, a idéia de se utilizar elementos de jogos em atividades que não são jogos, tem sido utilizado há muito tempo. Na educação de crianças, por exemplo, as mesmas podiam ter seus esforços e trabalhos reconhecidos por meio de estrelinhas ou outros tipos de recompensas dadas por seus educadores, como explica Silva et al. (2014).

De acordo com Silva et al. (2014), no Brasil existem diversas instituições públicas e privadas que apoiam o desenvolvimento e uso de ambientes gamificados. A exemplo disto, o Ministério da Educação visa fornecer suporte para o ambiente gamificado de apoio ao ensino *Geekie games* que possibilita aos estudantes se prepararem para o Exame Nacional de Ensino Médio (ENEM). Os resultados da utilização da ferramenta, segundo Silva et al. (2014), foram considerados positivos e o Ministério da Educação levanta a possibilidade de extender o uso da ferramenta gamificada para outros sistemas de avaliação.

Segundo Lee e Hammer (2011), somente a utilização de elementos de jogos no ensino não resolve a falta de empatia no processo de aprendizagem dos alunos uma vez que a utilização de mecanismos como por exemplo o sistema de pontuação já estão presentes no cotidiano escolar há anos. De acordo com as autoras, a utilização de elementos e característica de jogos devem provocar impactos tanto emocionais quanto sociais nos indivíduos para que eles tenham um aprendizado efetivo.

2.3 *Framework Octalysis*

No livro *"Actionable Gamification – Beyond Points, Badges, and Leaderboards"*, Chou (2017) apresenta uma série de características que, segundo ele, atraem e motivam pessoas a tomarem decisões e a realizarem determinadas atividades. Chou (2017) ressalta

ainda que essas características estão presentes na maioria dos jogos bem sucedidos.

Ao estudar uma variedade de técnicas de jogos que fazem com que os mesmos sejam tão atraentes, Chou (2017) notou que estas técnicas impulsionam os jogadores de maneiras diferentes. Algumas estimulam os jogadores a permanecerem no jogo por meio da inspiração e capacitação, outras estimulam por meio da manipulaçãoe e obsessão.

Com o aprofundamento de sua pesquisa e a partir da observação de como as técnicas de jogos influenciam nos jogadores, Chou (2017) propôs uma estrutura de design de gamificação que foi apelidada de *Framework Octalysis*.

O *Framework Octalysis* une elementos de *design* de jogos com a psicologia objetivando motivar os envolvidos no processo de gamificação a realizarem determinadas ações. O modelo de Chou (2017) apresenta oito eixos principais que são chamados de *core drives*, cada *core* apresenta características que por sua vez apresentam técnicas que influenciam diretamente os jogadores. Nas subseções a seguir, é apresentado cada um dos oito *core drives* definidos por Yu-Kai Chou.

2.3.1 Significado épico e chamado

Significado épico e chamado, segundo Chou (2017), é um *core* que provoca a sensação no jogador de que ele está envolvido em algo grande e que foi especialmente escolhido para executar uma determinada ação. Ao realizar uma contribuição para a plataforma Wikipedia, por exemplo, as pessoas acreditam que estão ajudando a proteger e disseminar algo que é maior que elas, o conhecimento humano (CHOU, 2017). O próprio autor relata em seu livro a experiência que teve com a criação de uma página na Wikipedia com informações de sua empresa, página criada por ele permaneceu no ar por pouco minutos. Vários contribuintes da Wikipedia relataram que a página não era significante o suficiente para merecer estar na Wikipedia, vários membros concordaram com a insignificância da página e, após aproximadamente dez minuto a página foi retirada do ar. Para Chou (2017), isso ocorreu devido ao fato de as pessoas que constituiam a "comunidade Wikipedia", ao realizarem o trabalho voluntário de inspecionar os conteúdos da plataforma, sentirem que estão fazendo parte de algo que é maior que elas e que de certa forma estão protegendo o conhecimento humano.

Em seguida são apresentadas algumas técnicas de “Significado épico e chamado” segundo Chou (2017). Demais técnicas podem ser apreciadas em seu livro presentes no livro *"Actionable Gamification – Beyond Points, Badges, and Leaderboards"*.

- Narrativa: A maioria dos jogos iniciam com uma história que mostra aos jogadores o contexto sobre o qual está inserido o jogo e o quão importante o jogador é para solucionar os desafios propostos. Uma das maneiras mais eficazes em utilizar este núcleo é por meio de uma narrativa envolvente.

- Herói da humanidade: Em seu livro, Chou (2017) apresenta exemplos que demonstram a natureza desta técnica. A empresa *"Tom's Shoes"*, por exemplo, envia um par de sapatos para uma criança em país de terceiro mundo sempre que um de seus clientes realiza um novo pedido. O site *"Free Rice"* doa dez grãos de arroz para cada resposta correta dadas para as perguntas educacionais postadas no site. Mecanismos como este, passam a sensação de "heroismo" aos envolvidos, fazendo com que os mesmos se sintam motivados em participar das atividades propostas;
- Sorte de iniciante;
- Criança predestinada;
- Almoço grátis;
- Elitismo;
- Significado superior.

2.3.2 Desenvolvimento e Conquista

Desenvolvimento e conquista é o segundo *core* do *framework* e está relacionado à motivação interna de cada pessoa em desenvolver suas habilidades e superar desafios. Neste núcleo, o desafio para se conquistar, por exemplo, um troféu ou distintivo é muito importante, uma vez que conquistar estes artefatos sem um desafio se torna insignificante. De acordo com Chou (2017), essa é a unidade mais fácil de ser projetada.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

- Pontos;
- Lista de desafios;
- Luta contra chefes;
- Tutorial passo a passo;
- Prêmios;
- Barra de progresso;
- Medalhas;
- Emblemas;
- Ranking.

2.3.3 Empoderamento e feedback

Empoderamento e feedback é o terceiro *core* e está ligado ao uso de táticas visando promover a realização pessoal do usuário de forma a aumentar seu potencial. Geralmente é expresso quando os usuários se envolvem em um processo criativo onde os mesmos descobrem coisas novas e, a partir disso, passam a tentar combinações novas, recebendo *feedbacks* de acordo com os resultados destas novas combinações.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

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

2.3.4 Propriedade e posse

Propriedade e posse é o quarto *core*. É onde os usuários são motivados a permanecerem nas atividades por sentirem que possuem controle sobre alguma coisa. De acordo com Chou (2017), quando uma pessoa sente que tem propriedade sobre alguma coisa, a atitude dela é de querer aumentar e melhorar o que possui. Se uma pessoa gasta muito tempo personalizando seu avatar, automaticamente sente mais propriedade e controle sobre ele ou quando ela é recompensada com uma moeda virtual ou bens virtuais, a postura dela é realizar mais atividades objetivando receber e acumular mais bens.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

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

2.3.5 Influência e relacionamento social

Influência e relacionamento social é o quinto *core* e envolve todos os elementos e características sociais que motivam as pessoas. São alguns destes elementos: aceitação social, competição, feedback social e até inveja. Em seu livro, Chou (2017) diz que, quando vemos um amigo que possui uma ótima habilidade em alguma coisa, logo nos sentimos motivados a alcançar o mesmo. Quando alguém nos fala que alcançou um determinado nível em um jogo, por exemplo, a nossa reação, na maioria das vezes, é de querer superar a pessoa e, portanto nos engajamos para realizar tal feito.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

- Amizades;
- Reconhecimento social;
- Atividades em grupo;
- Orientações.

2.3.6 Escassez e impaciência

A escassez e a impaciência é o sexto *core* e consiste na urgência de se obter algo só porque é raro, exclusivo ou inatingível. De acordo com Chou (2017), o facebook, por exemplo, na época em que foi lançado era exclusivo para estudantes de *Harvard*, depois foi aberto para outras Universidades de grande prestígio e finalmente, quando aberta para todos os públicos, universitários ou não, houve um grande aumento no número de pessoas que queriam participar da rede social pelo simples fato de que anteriormente não podiam.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

- Dinâmica de nomeação;
- Intervalos fixos;
- *Feedback paciente*;
- Fossos;
- Aceleradores;
- Contagem regressiva.

2.3.7 Imprevisibilidade e curiosidade

Imprevisibilidade e curiosidade é o sétimo *core* e exige o envolvimento constante, uma vez que não se tem nenhuma previsão do que acontecerá. Chou (2017) explica que, quando algo não se enquadra nos nossos ciclos regulares de reconhecimento de padrões, nosso cérebro entra em ação e passa a focar no inesperado, prendendo nossa atenção ao que pode vir a ocorrer. Yu-Kai Chou explica ainda que esse é o principal núcleo por trás dos vícios nos jogos de azar.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

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

2.3.8 Prevenção e perda

Prevenção e perda é o oitavo e último *core* do *framework* e está relacionado à motivação em se evitar que algo negativo aconteça. Exemplo: realizar alguma ação em um jogo para evitar que pontos sejam perdidos por falta de atividade.

Alguns exemplos de técnicas utilizadas neste *core* são apresentadas em seguida.

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

2.4 Elementos motivacionais: Octógono

De acordo com Chou (2017), tudo o que se faz em relação à gamificação é baseado em um ou mais *core drives*. Quando não há nenhum dos oito núcleos, a motivação é zero e nenhuma atividade é realizada.

2.4.1 Divisão dos motivadores de acordo com sua natureza

Cada um dos oito *core drives* possuem naturezas diferentes. Alguns fazem os usuários se sentirem poderosos mas não criam a sensação de urgência, outros já criam essa sensação nos usuários, além de obsessão e vício. Como resultado destas diferentes características, as oito unidades do *framework* são mapeadas em um octógono onde a posição de cada um determina a natureza da motivação. Na figura 2 é apresentado a distribuição de cada *core drive* no octógono.

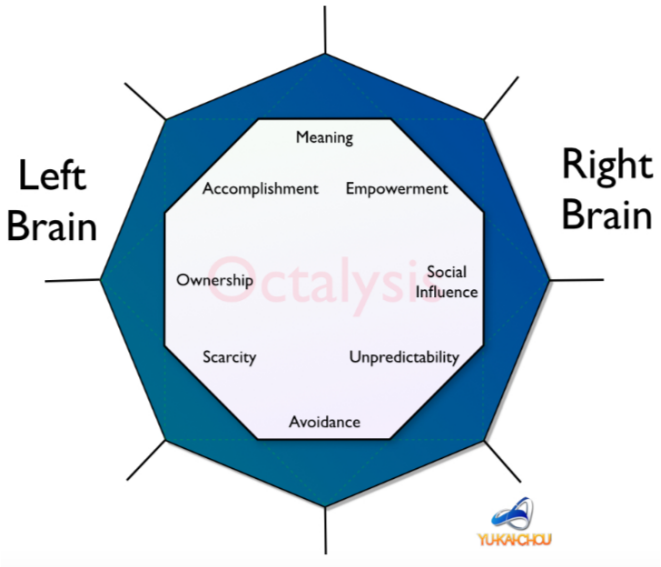

Figura 2 – Divisão do octógono

Fonte: (CHOU, 2017)

A estrutura do *Octalysis* é organizada de forma que os núcleos relacionados à criatividade, auto expressão e dinâmica social são representados dos lado direito do octógono ("*Right Brain*"). Os núcleos relacionados à logica, pensamento analítico e propriedade são representados graficamente no lado esquerdo (*"Left Brain"*). É importante ressaltar que "*Right Brain*" e *"Left Brain"* não são literais no que se refere à regiões do cérebro humano, a utilização destes termos é apenas um simbolismo.

2.4.2 Gamificação *White Hat* e *Black Hat*

Uma característica importante na representação dos núcleos motivacionais dentro do octógono é a forma como eles estão agrupados em relação ao impacto que causam sobre os participantes. Os núcleos localizados na parte superior do octógono são considerados "motivadores positivos" enquanto os núcleos localizados na parte inferior do octógono são considerados "motivadores negativos". Os *cores* apresentados na parte superior são chamados de *"White hat"* e os inferiores são conhecidos como *"Black hat"*. Na figura 3 é apresentado a divisão dos núcleos levando em consideração o tipo de motivador em que cada um é classificado.

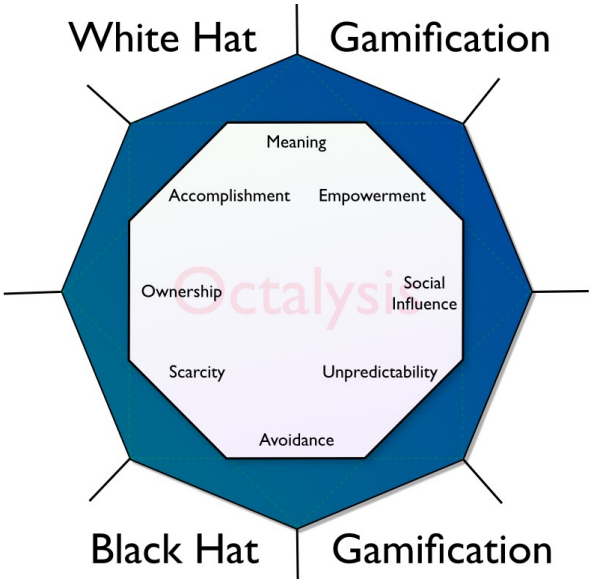

Figura 3 – Gamificação *White hat* e *Black hat*

Fonte: (CHOU, 2017)

Chou (2017) apresenta um exemplo do que vem a ser os núcleos *"White hat"* e *"Black hat"*. Se algo é envolvente por permitir que se expresse a criatividade, faz com que os indivíduos envolvidos se sintam bem sucedidos e poderosos, com certeza há um *core "White hat"* sendo utilizado. Por outro lado, se os envolvidos realizam determinadas ações por não saber o que acontecerá logo em seguida ou está constantemente com medo que algo ruim possa acontecer, neste caso, há um *core "Black hat"* sendo utilizado.

Como apresentado por Chou (2017), não é porque alguma coisa é categorizada como *"black"* que ela seja ruim. Se utilizadas corretamente podem gerar resultados saudáveis e positivos. De acordo com Chou (2017), um profissional de gamificação deve sempre considerar todos os *core drive* objetivando obter resultados positivos e produtivos.

2.4.3 Níveis *Octalysis*

O *framework Octalysis* é dividido em vários níveis que, quanto mais alto forem, mais domínio por parte do "gamificador" é requerido (CHOU, 2017). Nas subseções que se seguem, são apresentadas as características relacionadas aos três primeiros níveis do *octalysis*.

2.4.3.1 Nível 1

Para ser aplicado, este nível deve levar em consideração pontos fortes e fracos de vários produtos, este primeiro método permite identificar que tipos de motivação são fracas para que se possa introduzir novas melhorias (CHOU, 2017).

O segundo método consiste na criação de uma nova experiência baseada nos *core drives* através de um processo sistemático.

Como citado em seções anteriores, neste nível sempre há a necessidade de se utilizar ao menos um *core drive*, caso não seja utilizado nenhum dos oito núcleos, a motivação é zero e nenhum progresso relativo à gamificação é alcançado (CHOU, 2017).

2.4.3.2 Nível 2

Este nível envolve todas as quatro fases da jornada de um jogador. Em seguida são apresentadas cada uma das quatro fases, bem como suas características.

- Descoberta: Experimentações e ganhos de experiências por parte dos jogadores;
- Entrada: Aprendizado das regras e ferramentas necessárias para jogar o *game*.
- Dia a dia: Jornada regular de ações e comportamentos que os jogadores devem realizar para completar os objetivos;
- Fim de jogo: Manter os jogadores motivados a continuarem jogando.

Chou (2017) argumenta que o motivo pelo qual as pessoas utilizam um produto no dia 1 muitas das vezes é bastante diferente do motivo de uso no dia 100. Por este motivo, é importante se planejar cada uma das quatro fases envolvidas na trajetória dos jogadores. Caso não haja nenhum *core drive* envolvido em alguma das fases, o jogador não possui uma razão para passar para a próxima fase e ele simplesmente desiste.

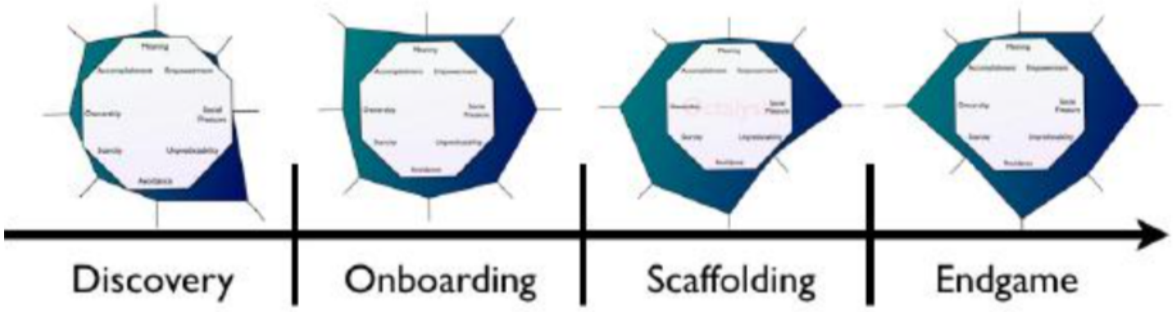

Figura 4 – As fases da jornada de um jogador.

Fonte: (CHOU, 2017)

Observando a figura 4, é possível notar os *core drives* mais proeminentes durante cada fase de experiência da jornada dos jogadores.

2.4.3.3 Nível 3

Este nível está relacionado com a identificação dos perfis de jogadores e como eles se comportam quando estão envolvidos em jogos. Mesmo que seja praticamente impossível, este nível permite aos projetistas de gamificação buscar atender as necessidades e motivações dos mais diferentes perfis.

2.4.3.3.1 *Design* de gamificação: Octalysis e Bartle

Embora a classificação dos perfis dos jogadores seja uma proposta criada por Bartle (1996), Chou (2017) apresenta um modelo que une os perfis apresentados por Bartle com os *core drives* apresentados no *framework Octalysis* de forma a criar um *design* de gamificação mais apropriado para cada tipo de perfil.

Como apresentado por Vianna et al. (2013) no livro "Gamification: reinventando empresas a partir de jogos" e com base nos estudos de Bartle (1996), existem quatro perfis de jogadores que possuem características e comportamentos específicos quando envolvidos em jogos. Cada um destes quatro perfis são apresentados na subseção a seguir.

2.4.3.4 Perfis de Jogadores

2.4.3.4.1 Predadores (*Killers*)

São aqueles jogadores que participam de uma competição pelo prazer de derrotar os adversários. O objetivo desse tipo de jogador é ser o melhor, não se importam com o que está em disputa. Possuem comportamentos agressivos e focados em conquistar a posição de liderança. Seus desejos de liderança se sobrepõem à cooperação.

2.4.3.4.2 Realizadores (*Achievers*)

A motivação destes jogadores consiste na realização de todas as atividades que o jogo apresenta. Esses jogadores apreciam a sensação constante de vitória mesmo que os objetivos a serem alcançados não sejam tão relevantes ou significativos. Atuam de maneira cordialmente competitiva, mesmo que não estejam liderando algum placar. São caracterizados por se destacarem dos demais jogadores de maneira leal, isto é, por meio de suas próprias conquistas.

2.4.3.4.3 Exploradores (*Explorers*)

Este é um grupo que se caracteriza pela busca e interesse em desvendar todas as possibilidades dos jogos. São indivíduos curiosos que se dedicam em estudar e desenvolver habilidade que lhes permitam solucionar os mais diversos desafios. Para este tipo de perfil, a trejetória é mais importante que a conquista.

2.4.3.4.4 Socializadores (*Socializers*)

São, no geral, os indivíduos que enxergam o contato com jogos como um meio de interação social. Mais significativo que atingir os objetivos propostos ou realizar derterminadas tarefas é a oportunidade de reforçar e estabelecer vínculos sociais. Os socializadores, em sua maioria, preferem jogos cooperativos que necessitam de trabalho conjunto. Estes representam 80% do total de jogadores.

2.4.3.5 Relação entre *Core drives* e Perfis de jogadores

Como apresentado por Chou (2017) em seu livro *"Actionable Gamification – Beyond Points, Badges, and Leaderboards"*, para cada perfil de jogador, exitem um conjunto de *core drives* que os motivam a engajarem no desenvolvimento e realização de determinadas ações dentro de um jogo. Na tabela a seguir , são apresentados os *core drives* predominantes em cada perfil que, se utilizados de maneira correta, podem aumentar o interesse e engajamento dos perfis identificados na seção anterior.

Perfil	**CD 1 - Significado épico e chamado**	**CD 2 - Desenvolvimento e Conquista**	**CD 3 - Empoderamento e feedback**	**CD 4 - Propriedade e posse**	**CD 5 - Influência e relacionamento social**	**CD 6 - Escassez e impaciência**	**CD 7 - Imprevisibilidade e curiosidade**	**CD 8 - Prevenção e perda**
Predadores (*Killers*)		x			x			
Realizadores (*Achievers*)		x				x		
Exploradores (*Explorers*)							x	
Socializadores (*Socializers*)					x			

Tabela 1 – Principais *core drives* envolvidos em cada perfil de jogador

Fonte: (CHOU, 2017)

Chou (2017) apresenta um gráfico relacionando os quatro perfis dos jogadores com os *core drives* característicos de cada um deles. O gráfico é apresentado na figura 5.

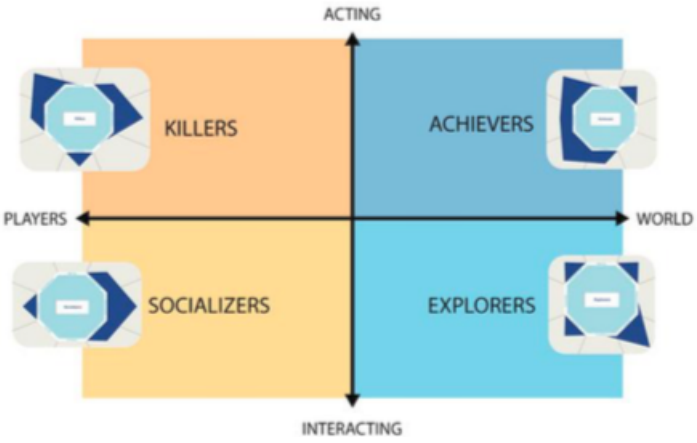

Figura 5 – Perfil de jogadores

Fonte: (CHOU, 2017)

Na figura 6, são apresentados os *core drives* mais utilizados em cada uma das quatro fases da jornada dos jogadores levando em consideração os quatro tipos de perfis identificados por Bartle (1996).

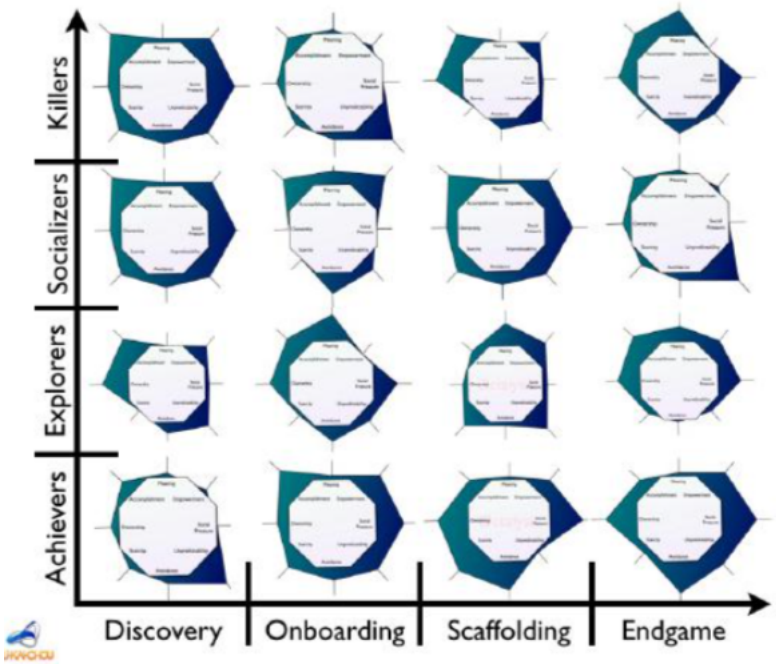

Figura 6 – *Core drives* mais utilizados de acordo com as fases e perfis de jogadores.

Fonte: (CHOU, 2017)

2.5 Modelagem de banco de dados

Existem diversas formas de se representar graficamente a forma como uma base de dados deverá estar consolidada em um banco de dados após sua implementação. Nesta seção, apresenta-se uma forma específica de modelagem de base de dados, bem como suas características.

2.5.1 Diagrama entidade-relacionamento

Segundo Heuser (1998) , a abordagem mais utilizada e conhecida é a entidade-relacionamento (ER) onde o modelo de dados é geralmente representado graficamente através de um diagrama entidade-relacionamento (DER). Esta abordagem foi criada em 1976 por Peter Chen e é considerada como um padrão para a modelagem conceitual.

A abordagem entidade-relacionamento é baseada em dois principais pilares que são apresentados em seguida.

2.5.1.1 Entidade

De acordo com Heuser (1998), uma entidade, no modelo conceitual, representa um conjunto de objetos da realidade modelada. Seu principal objetivo é modelar de forma abstrata um banco de dados, onde se te interesse somente nos objetos sobre os quais deseja-se manter informações. No DER, uma entidade é representada por meio de um retângulo contendo o nome da entidade.

2.5.1.2 Relacionamento

Como apresentado por Heuser (1998), o DER permite a especificar as propriedades dos objetos que serão armazenados no banco de dados, como por exemplo o relacionamento/associação entre os objetos. No DER, um relacionamento é representado por meio de um losango que são ligados por linhas aos retângulos que representam as entidades que participam de um determinado relacionamento.

Na figura 7 é apresentado um exemplo simples de um diagrama entidade-relacionamento (DER). É possível notar no diagrama a existência de atributos e cardinalidade, estes dois são apresentados logo em seguida.

Os atributos correspondem às características/qualidades que descrevem uma entidade, são representados por elipses ou círculos acompanhados por seus respectivos nomes (CARDOSO; MARA, 2013).

As cardinalidades representam a restrição do número de objetos que podem participar do relacionamento. Na notação de Peter Chen, as cardinalidade são apresentadas próximas às ligações de relacionamento e são compostas da quantidade mínima e máxima

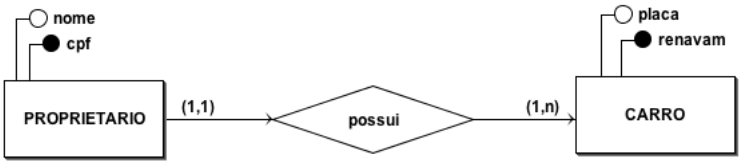

Figura 7 – Exemplo de DER

Fonte: Autor

de objetos que podem participar do relacionamento. Tais características podem ser vistas na figura 7 (CARDOSO; MARA, 2013).

2.5.2 Diagrama Lógico

O diagrama lógico leva em consideração as limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual (CARDOSO; MARA, 2013).

2.6 Metodologia cascata

De acordo com Semedo et al. (2012), metodologia pode ser definida como um conjunto de procedimentos, técnicas, documentação e ferramentas que auxiliam os responsáveis ou interessados no desenvolvimento e implementação de um sistema de informação. Na maioria das vezes, o sucesso de um projeto de software depende de vários fatores que vão desde o planejamento à escolha mais adequada de uma metodologia.

O modelo cascata, também conhecido como modelo tradicional, é uma das metodologias mais antigas e conhecidas. A utilização desta metodologia consiste em seguir o desenvolvimento do projeto de forma sequencial, onde só se deve passar para os níveis seguintes após a conclusão do nível anterior. Esta metodologia envolve duas grandes fases: levantamento de requisitos/necessidades e design. (SEMEDO et al., 2012)

Muitos autores desencorajam a utilização da metodologia cascata no desenvolvimento de grandes sistemas, como é o caso do pesquisador Gilb e Finzi (1988 apud SEMEDO et al., 2012) e Brooks Jr (2019 apud SEMEDO et al., 2012). Este tipo de metodologia deve ser usada apenas em situações onde os requisitos do software são estáveis e requisitos que possam vir a surgir no futuro sejam previsíveis. (SEMEDO et al., 2012)

No modelo cascata apresentado por Pressman (2011) no livro "Engenharia de Software: Uma abordagem profissional", o autor apresenta cinco fases sequenciais que devem

ser satisfeitas no desenvolvimento de um sistema obedecendo esta metodologia(cascata). Cada uma das cinco fases podem ser vistas na figura 8 apresentada logo em seguida.

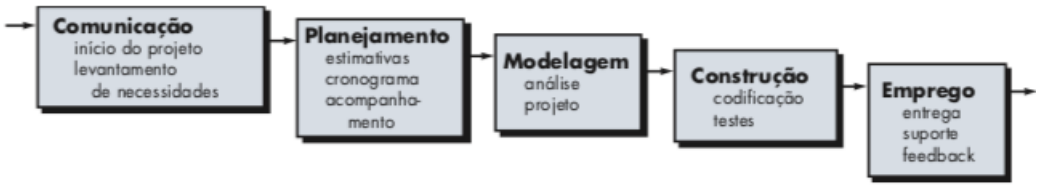

Figura 8 – Modelo cascata

Fonte: (PRESSMAN, 2011)

3 Metodologia

Neste capítulo, são apresentados com detahes o passo a passo realizado para a construção de uma ferramenta web de apoio ao ensino e aprendizagem de programação gamificada.

3.1 *Core drives*

Os *core drives* implementados no desenvolvimento da aplicação, foram definidos a partir de um questionário que teve como objetivo identificar as características em jogos que mais atraem os alunos/jogadores.

A partir de análises realizadas no conjunto de dados obtidos, identificou-se predominantemente os *core drives* apresentados na tabela 2.

Core drive	**Técnicas**
Desenvolvimento e conquista	Pontos, *ranking*, lista de desafios e barra de progresso.
Empoderamento e *feedback*	Desbloqueio de marcos e *feedback*
Significado épico e chamado	Narrativa e elitismo
Prevenção e perda	Perda de pontos e perda de progresso

Tabela 2 – *Core drives* implementados

Fonte: Autor

3.1.1 Desenvolvimento e conquista

3.1.1.1 Pontos

A medida que os alunos/jogadores realizam alguma entre a variedade de atividades disponíveis na ferramenta, o mesmo adquire pontos. As atividades possuem diferentes pontuações de acordo com sua natureza e dificuldade. Na tabela 4, são apresentados as atividades e suas respectivas pontuações.

Atividade	**Pontuação**
Questões Quiz	10
Desafios (jogador vs PC)	100
Criar um desafio para outtro jogador (jogador vs jogador)	100
Responder desafios de outros jogadores (jogador vs jogador)	100
Avaliar resposta do desafiado (jogador vs jogador)	100

Tabela 3 – Pontuações

Fonte: Autor

3.1.1.2 *Ranking*

A ferramenta fornece uma página dedicada à apresentar as pontuações dos jogadores bem como seus nomes, matrículas e posições, desta forma, todos os jogadores/alunos têm a noção do seu progresso em relação aos demais usuários do sistema. Tornando, desta forma, a busca por pontuações e usp dos módulos disponíveis mais competitivos.

3.1.1.3 Lista de desafios

O sistema possui um módulo destinado inteiramente para desafios que podem ser cadastrados pelo administrador do sistema ou em casos de desafios jogador vs jogador, os próprios usuários da ferramenta podem criar seus próprios desafios.

3.1.1.4 Barra de progresso

A ferramenta possui uma página de perfil contendo diversas informações do jogador, isso inclui uma barra de progresso que mostra a evolução do jogador na utilização da ferramenta e seu avanço nos doze níveis disponíveis na mesma.

3.1.2 Empoderamento *feedback*

3.1.2.1 Desbloqueio de marcos

A ferramenta possui um sistema de divisão de conteúdos por níveis. No total existem 12 níveis que vão sendo desbloqueados a medida que os jogadores conquistam determinadas pontuações.

Na tabela 4 são apresentadas as pontuações necessárias para desbloquear cada nível.

Nível	**Pontuação**
1	< 800
2	> 800
3	> 1600
4	> 3200
5	> 6400
6	> 12800
7	> 25600
8	> 51200
9	> 102400
10	> 204800
11	> 409600
12	> 819200

Tabela 4 – Pontuações

Fonte: Autor

3.2 Requisitos

Tendo como base a pesquisa realizada na primeira parte deste trabalho, nesta seção, são apresentados os requisitos implementados na construção da ferramenta.

Como apresentado no referencial teórico deste trabalho, o *framework octalysis* é dividido em vários níveis. Neste trabalho, a solução tecnológica foi construída sobre o nível 1.

Na tabela a seguir, é apresentado uma comparação entre os requsitos levantados em relação aos implementados.

Identificador	**Requisito**	**Implementado?**
RF 01	CRUD de jogadores	Sim
RF 02	Ativar/desativar perfil de jogadores	Sim
RF 03	Rankeamento de jogadores	Sim
RF 04	CRUD de disciplinas	Não
RF 05	CRUD de turmas	Não
RF 06	CRUD de conteúdos para estudo	Sim
RF 07	CRUD de desafios	Sim
RF 08	Sistema de recompensas (pontuação)	Sim
RF 09	Dashboard pessoal de desempenho (jogadores)	Sim
RF 10	CRUD de professores de disciplinas	Não
RF 11	Painel de acompanhamento de desempenho dos alunos (professores)	Não
RF 12	Narrativa temática	Sim
RF 13	Personagens e guerreiros da narrativa	Sim
RF 14	Lógica de progressão de jogadores (*levels*)	Sim
RF 15	CRUD de linguagens de programação	Sim
RF 16	Desafios entre jogadores (individual)	Sim
RF 17	CRUD de questões	Sim
RF 18	Desafios entre times/grupos	Não
RF 19	Torneios/campeonatos entre turmas	Não

Tabela 5 – Requisitos implementados

Fonte: Autor

4 Solução proposta

4.1 Ferramentas de desenvolvimento

A escolha das ferramentas de desenvolvimento e demais tecnologias, foram realizadas com base em experiências prévias, disponibilidade de documentações e comunidades de apoio ao desenvolvedor.

A seguir são apresentadas as principais tecnologias utilizadas no desenvolvimento da ferramenta bem como uma breve descrição de suas características.

4.2 *Backend*

- Java: Linguagem de programação orientada a objetos desenvolvida na década de 90.Diferente das linguagens de programação modernas que são compiladas para código nativo, a linguagem java é compilada para um *bytecode* que é interpretado por uma máquina virtual (*java virtual machine*). Por conta disso, a linguagem torna-se altamente portável, isto é, independente de plataforma (SCHOOLS, 1999b).
- MySql: Sistema de gerenciamento de banco de dados que utiliza linguagem sql como interface. Voltado para o armazenamento e manutenção de registros de um banco de dados (MYSQL, 2019).
- Maven: Ferramenta de automação de compilação muito utilizada em projetos que envolvem a linguagem java. O maven utiliza um arquivo *xml* (POM) para descrever o projeto de software sendo implementado, suas dependências, componentes externos, *plugins* necessários, diretórios e ordem de compilação (DEVMEDIA, 2012).
- Jersey: *Framework* que facilita o desenvolvimento de APIs (Application Programming Interface) RESTful (ECLIPSE, 2019).
- JDBC: *Driver* que possui um conjunto de classes e interfaces java que permitem o envio/comunicação com qualquer banco de dados relacional. Permitindo interligar a aplicação com um banco de dados (DEVMEDIA, 2007).
- Tomcat: Consiste em um servidor de aplicações java que permite que o mesmo funcione na *web* (APACHE, 2013).

4.2.1 Segurança dos dados

Visando prover maior segurança das informações dos jogadores/usuários, informações críticas como senhas de usuários, são persistidas no banco de dados de forma criptografada. Desta forma, nem os desenvolvedores/manetedores ou possíveis tentativas de ataque *hacker* têm acesso a essa informação.

Nessa solução, utilizou-se como algoritmo de criptografia o MD5.

4.3 *Frontend*

- Vue.js: *Framework* javaScript voltado para o desenvolvimento de interfaces de usuário (VUE, 2014).
- Axios: É um cliente HTTP que permite consumir e exibir dados de uma API (Application Programming Interface) de forma facilitada e ágil (VUE, 2019).
- Html: Linguagem de marcação bastante utilizada na construção de páginas web. Que podem ser interpretados por qualquer navegador (SCHOOLS, 1999a).
- CSS: Mecanismo que permite adicionar cores, fontes, espaçamentos e efeitos em um documento *web* (html). (HOSTINGER, 2019)
- Bootstrap:*Framework web* voltado para o desenvolvimento de interfaces para aplicações e sites usando HTML, CSS e JavaScript, melhorando a experiência do usuário, tornando um site amigável e responsível (BOOTSTRAP, 2011).
- Json: *JavaScript Object Notation* é uma formatação de troca de dados. Possui formato textual e é completamente independente de linguagem. É consituido por conjuntos de chave e seu respectivo valor (CROCKFORD, 2019).

4.4 Arquitetura

No desenvolvimento da ferramenta, tem-se como modelo arquitetural a ser utilizado o "n camadas". Esse tipo de arquitetura consiste na separação de responsabilidades específicas para cada uma das camadas presentes na construção do sistema (AZURE, 2018). A letra “n” representa a quantidade de camadas em que a construção do sistema está dividida, nesta solução, o desenvolvimento do sistema está dividido em 4 camadas que são: apresentação, serviço, modelo e armazenamento.

O modelo simplificado apresentado na figura 9 demonstra como ocorre o fluxo de comunicação entre as principais tecnologias usadas no desenvolvimento.

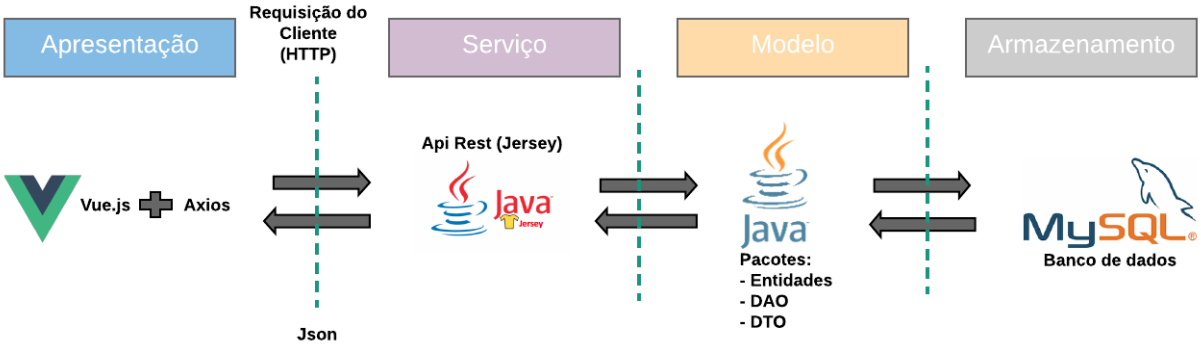

Figura 9 – Representação da arquitetura do sistema.

Fonte: Autor

4.4.1 Camada 1: Apresentação

Esta camada envolve todas as tecnologias responsáveis pela apresentação de informações ao usuário e realização de requisições para obter as informações a serem apresentadas. As principais tecnologias utilizadas para a consolidação desta camada foram: HTML, CSS, Vue.js, Axios (requisições) e bootstrap.

4.4.2 Camada 2: Serviço

A camada de serviço é responsável pelo preparo e disponibilização de informações obedecendo uma certa estrutura (json para esta solução) quando estas forem requisitadas. Cabe a ela iniciar a comunicação com as subcamadas de forma a solicitar que determinadas informações sejam recuperadas e organizar estas informações em uma estrutura que seja compatível com o formato de troca de dados estabelecido. Para a construção do serviço, utilizou-se o *framework* de construção de APIs REST, Jersey.

4.4.3 Camada 3: Modelo

Nesta camada são organizados em pacotes todos os arquivos necessários para realizar o mapeamento de objetos para os tipos de dados aceitos pelo banco de dados adotado e obter uma conexão com o banco de dados. Esses dois itens citados são feitas por classes DAO (Objeto de Acesso a Dados).

Além das responsabilidades citadas no parágrafo anterior, cabe a esta camada a responsabilidade de definir a estrutura de transferência de dados entre os subsistemas do software. Este padrão é conhecido como DTO (Objeto de Transferência de Dados) e geralmente são utilizados em conjunto com objetos de negócio de forma a se obter maior flexibilidade no acesso a determinados dados.

4.4.3.1 Camada 4: Armazenamento

Esta camada consiste na utilização de um sistema gerenciador de banco de dados (SGBD), mySql para esta solução, que passa a ser responsável pelo armazenamento de forma segura e consistente dos dados do sistema. Além de realizar o armazenamento, o SGBD é responsável por recuperar e manipular informações sempre que for necessário e fornece-las sempre que forem solicitadas.

4.5 Desenvolvimento cascata

No desenvolvimento da ferramenta, escolheu-se a metodologia cascata devido a natureza de desenvolvimento deste trabalho ser compatível com as catacterísticas propostas pela metodologia. O desenvolvimento da ferramenta partiu da definição dos requisitos, planejamento, desenvolvimento, testes e entrega final.

As características desta metodologia podem ser apreciadas em detalhes no referencial teórico deste trabalho.

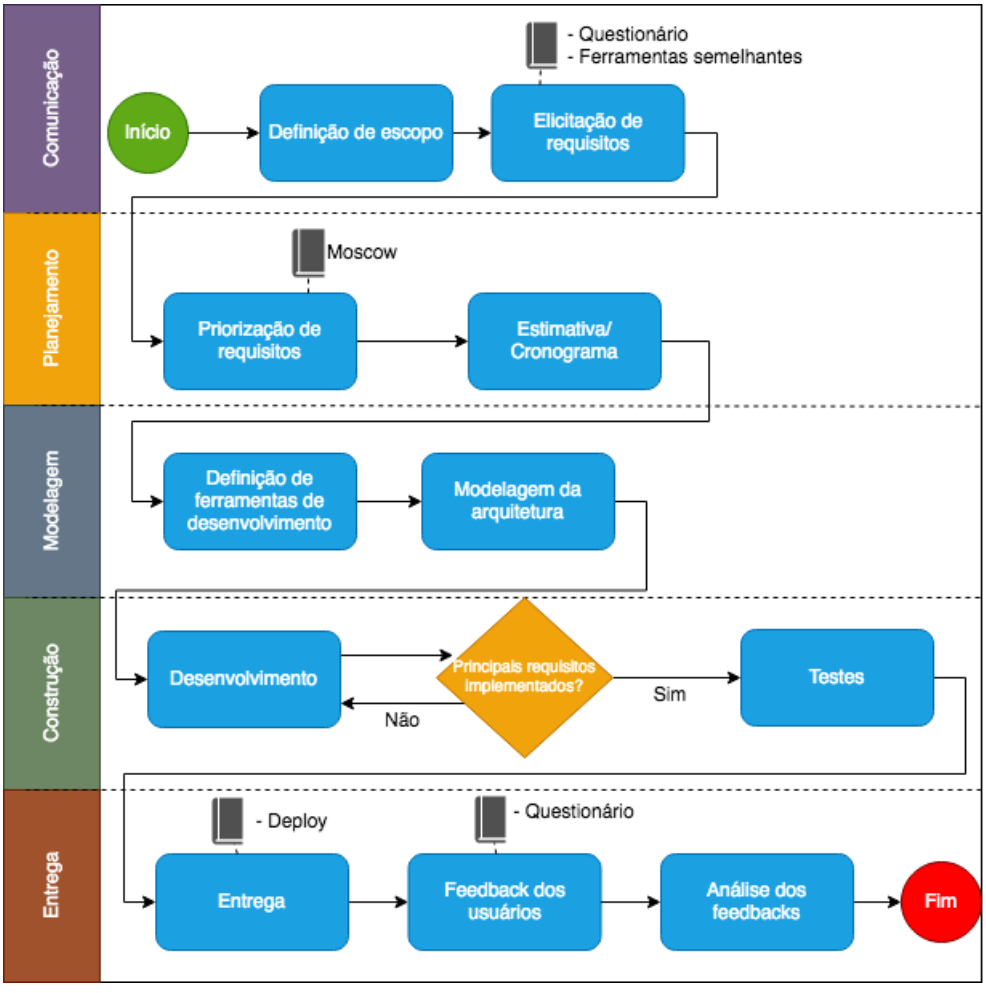

Figura 10 – Modelo de processos com principais atividades

Fonte: Autor

No modelo apresentado na figura 10, é possível identificar em cada fase as principais atividades, iniciando pela definição do escopo do projeto, elicitação de requisitos por meio da aplicação de um questionário e extração de algumas características de *design* de outras ferramentas e jogos, priorização dos requisitos por meio da ferramenta *moscow*, definição das tecnologias de desenvolvimento utilizadas, modelagem arquitetural da ferramenta construída, desenvolvimento da ferramenta utilizando ferramentas e seguindo metodologia definida, testes, disponibilização da versão a ser utilizada como foco de análise, coleta de *feedback* por meio da aplicação de questionário e análise dos dados.

4.6 Hospedagem

Após a conclusão da construção da ferramenta, iniciou-se um estudo em cima de serviços de hospedagem. Nesta fase também foi levado em consideração, assim como todas as tecnologias utilizadas no desenvolvimento desta ferramenta, os conhecimentos prévios e experiências com o serviço.

Com relação ao banco de dados, utilizou-se o próprio heroku como hospedagem. Embora aplicação e o banco de dados estejam hospedados neste serviço, os dois são independentes um do outro, isto é,Em caso de indisponibilidade de um destes, o outro não é afetado no que se refere a disponibilidade do serviço.

4.7 Testes

4.8 Modelagem de dados

Como apresentado no referencial teórico de trabalho, existem diversas formas de representar em forma de diagrama banco de dados. Neste trabalho utilizou-se a diagramação de Peter Chen.

Neste capítulo são apresentados os elementos característicos da ferramenta como: personagens, narrativa, temática entre outros elementos que foram pensados como forma de gamificar as atividades de aprendizagem de programação.

4.9 Narrativa

Com o objetivo de aumentar o engajamento dos estudantes/jogadores, fora desenvolvida uma história que se passa em um mundo onde criaturas (Orcs) invadem o vilarejo do jogador que, motivado pelo desejo de vingança e tendo sido escolhido entre uma legião de outros guerreiro, dá início à jornada onde o mesmo deve cumprir com desafios como: quiz, desafiar outros jogadores entre outras atividades que dão ao jogador pontos

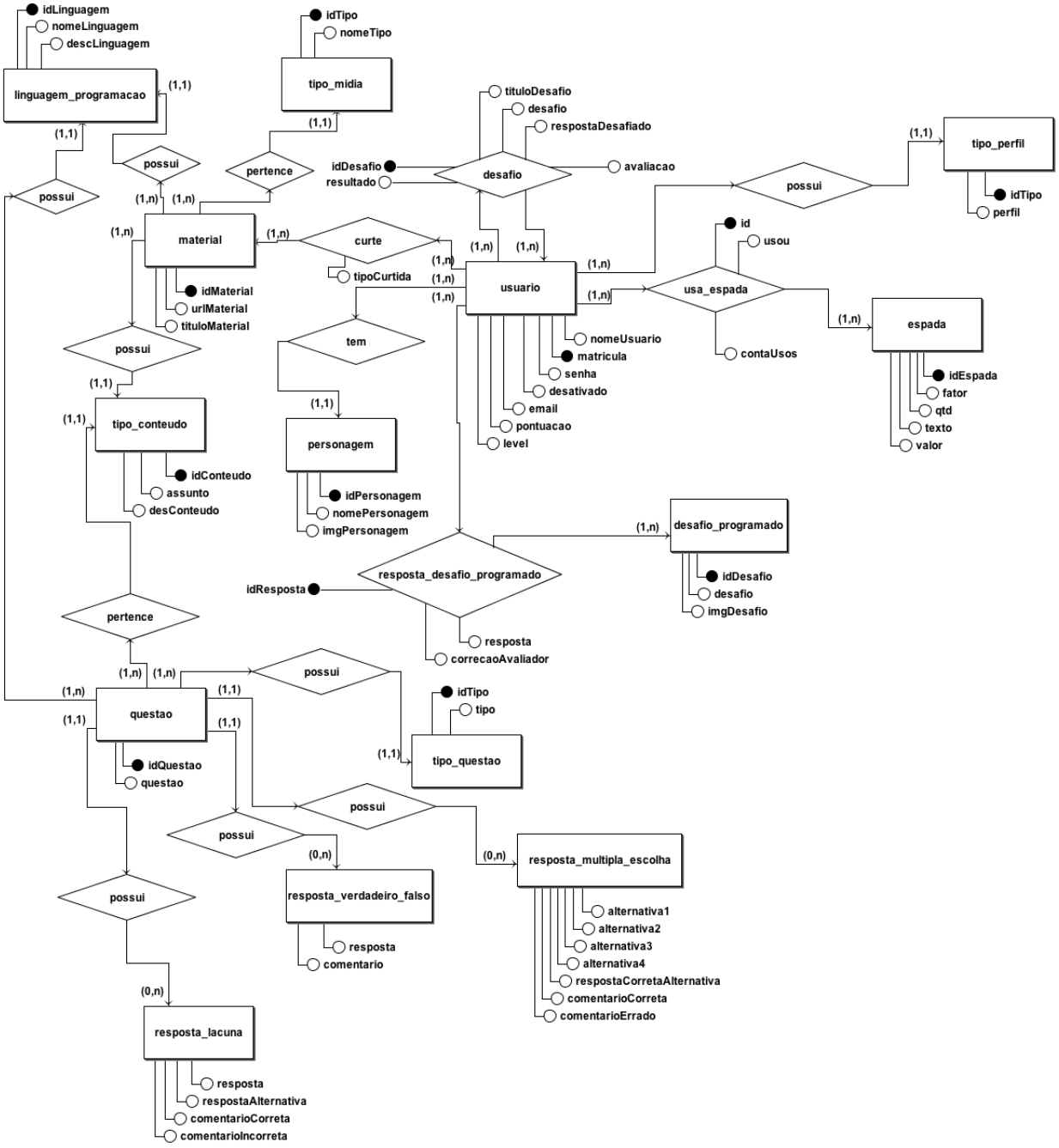

Figura 11 – DER da ferramenta

Fonte: Autor

que podem ser trocados por itens ou serem acumulados afim de obter boas posições nos *rankings*.

4.10 Personagens

Ao jogador, após a realização do primeiro *login* na ferramenta, são apresentados seis personagens com diferentes características, onde o jogador deve selecionar um único personagem para iniciar a jornada de aprendizado. Ao selecionar um determinado

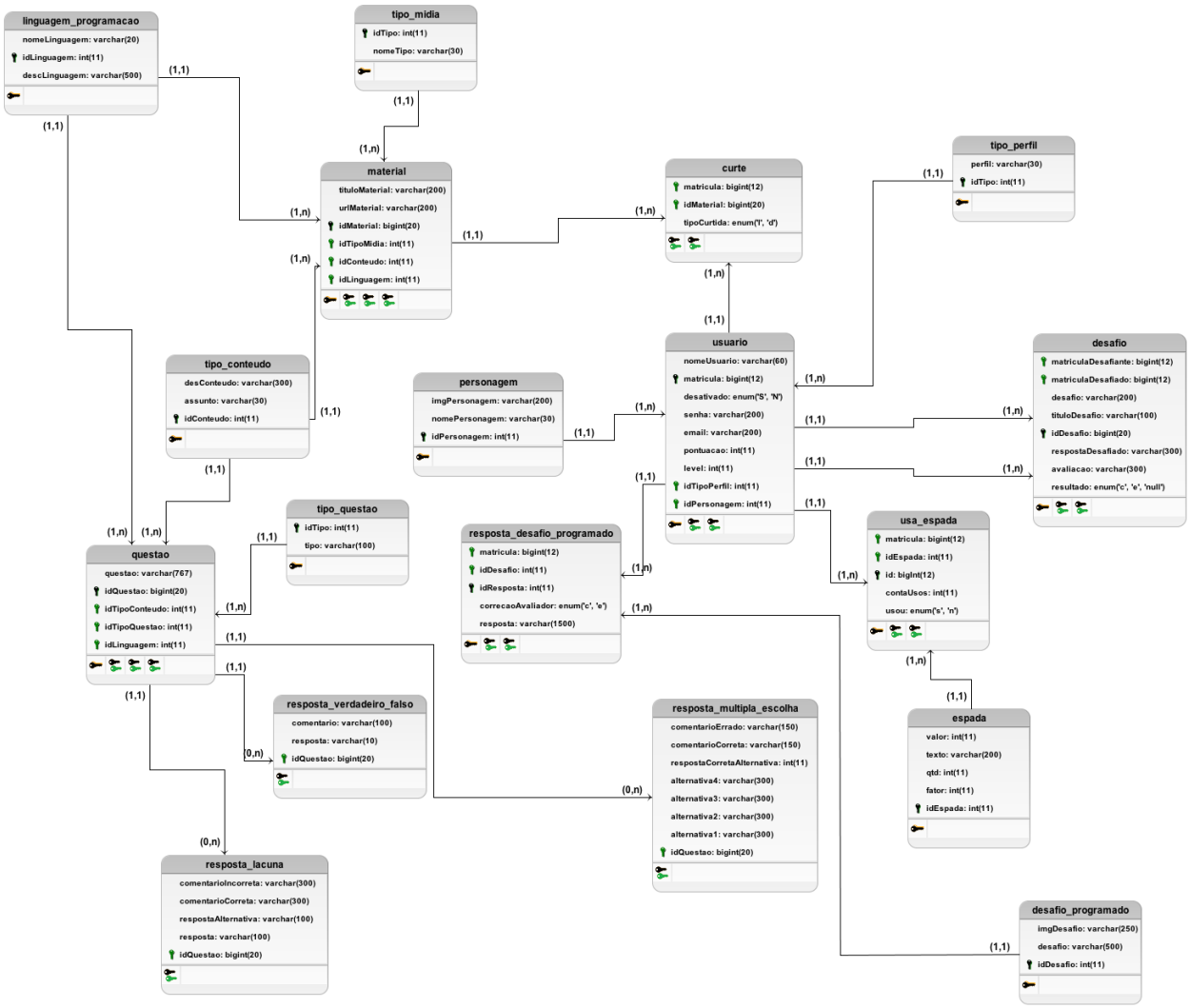

Figura 12 – DL da ferramenta

Fonte: Autor

personagem, não é permitido ao jogador/estudante modifica-lo no decorrer do uso da ferramenta.

Os personagens são apresentados a seguir.

4.10.1 Aisha

Figura 13 – Personagem Aisha

Fonte: Autor

4.10.2 Voxter

Figura 14 – Personagem Voxter

Fonte: Autor

4.10.3 Lince

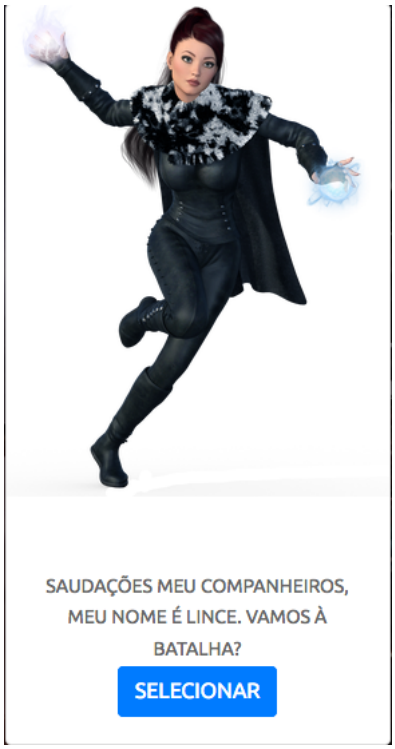

Figura 15 – Personagem Lince

Fonte: Autor

4.10.4 Amazona

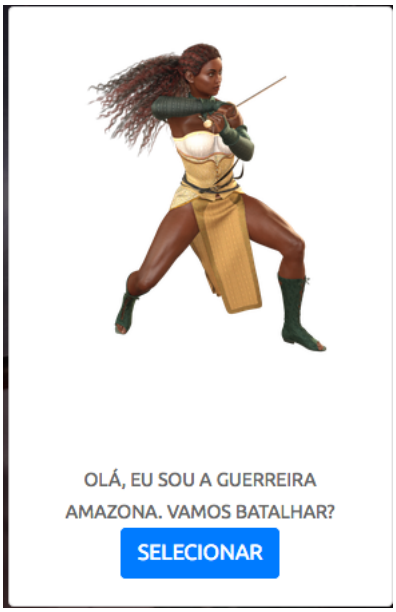

Figura 16 – Personagem Amazona

Fonte: Autor

4.10.5 Tymer

Figura 17 – Personagem Tymer

Fonte: Autor

4.10.6 Scar

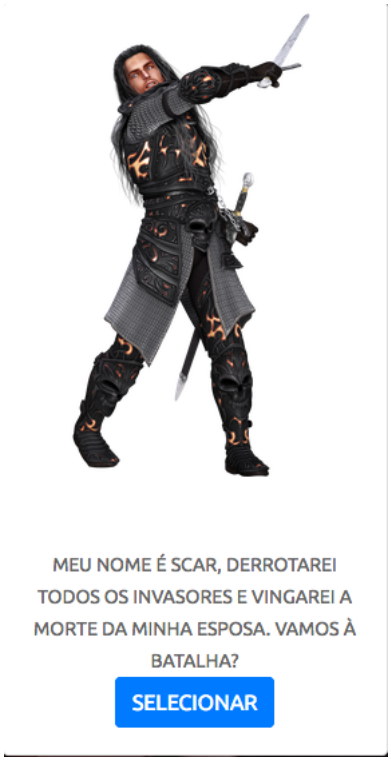

Figura 18 – Personagem Scar

Fonte: Autor

4.11 Questões

Existem diferentes tipos de questões que podem ser solucionados pelo jogador. A correção destas, são feitas automaticamente pelo sistema de acordo com respostas pré cadastradas na base de dados.

Os tipos de questões são:

- Verdadeiro ou falso;
- Lacunas;
- Múltipla escolha

Cada resposta contabilizada como correta, são adicionados 10 pontos à pontuação total do jogador. Não há limites de tentativas de solucionar as questões.

Para cada tentativa de resposta julgada como incorreta, é apresentado a alternativa correta para aquela questão.

4.12 *Levels*

A medida que o jogador conquista maiores pontuações, é desbloqueado o novo level que consiste na liberação de novos conteúdos mais avançados em relação aos anteriores. No total, existem 12 levels, os conteúdos abordados em cada um deles são apresentado logo a seguir.

Level	**Conteúdo**
01	Algoritmos
02	Linguagem de Programação
03	Tipos de dados
04	Entrada e saída de dados
05	Operações primitivas
06	Variáveis e expressões
07	Estruturas de decisão
08	Laços de repetição
09	Vetores
10	Matrizes
11	Funções e Procedimentos
12	Arquivos

Tabela 6 – Conteúdos por *level*

Fonte: Autor

Os levels são compostos por questões, como as apresentadas na seção anterior.

4.13 Desafios

4.13.1 Jogador contra Jogador

4.13.2 Desafio individual

Os desafios individuais consistem em resolver diferentes tipos de desafios pré cadastrados no sistema que, uma vez solucionados e avaliados pelo administrador do sistema e, tendo sido julgado por este como correto, é adicionado à pontuação total do jogador mais 100 pontos. Os desafios variam de simples respostas até implementação de códigos.

Referências

ALMEIDA, E. et al. Ambap: Um ambiente de apoio ao aprendizado de programação. 09 2019. Citado na página 14.

APACHE. *Apache Tomcat*. 2013. Disponível em: <http://tomcat.apache.org/index.html>. Citado na página 38.

ARENA, D. A.; SCHWARTZ, D. L. Experience and explanation: Using videogames to prepare students for formal instruction in statistics. *Journal of Science Education and Technology*, v. 23, n. 4, p. 538–548, Aug 2014. ISSN 1573-1839. Disponível em: <https://doi.org/10.1007/s10956-013-9483-3>. Citado na página 15.

AZURE, M. *Estilo de arquitetura de N camadas*. 2018. Disponível em: <https://docs.microsoft.com/pt-br/azure/architecture/guide/architecture-styles/n-tier>. Citado na página 39.

BARATA, G. et al. Engaging engineering students with gamification. In: *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*. [S.l.: s.n.], 2013. p. 1–8. Citado 2 vezes nas páginas 14 e 15.

BARTLE, R. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, v. 1, n. 1, p. 19, 1996. Citado 2 vezes nas páginas 28 e 30.

BARUQUE, A. B. e L. Gamificação aplicada na graduação em jogos digitais. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 26, n. 1, p. 677, 2015. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/5338>. Citado na página 15.

BOOTSTRAP. *Bootstrap*. 2011. Disponível em: <https://getbootstrap.com/>. Citado na página 39.

BORGES, M. Avaliação de uma metodologia alternativa para a aprendizagem de programação. 09 2019. Citado na página 18.

BROOKS JR, F. Essence and accidents of software engineering. *IEEE Computer. April, 1987. vol. 20: pp. 10-19 : ill. includes bibliography. – See also Information Processing '86 edited by H.J. Kugler; and Brooks, F.P. 'The Mythical Man-Month,' Addison-Wesley, 1978*, 10 2019. Citado na página 32.

CALIXTO, G. de L. Fatores relacionados ao ensino de programação: Uma análise das disciplinas introdutórias em cursos de engenharia de softtware. p. 1–82, 2016. Citado na página 16.

CARDOSO, G. C.; MARA, V. C. *Sistema de banco de dados: Uma abordagem introdutória e aplicada*. 1. ed. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 31 e 32.

CHOU, Y.-k. *Actionable gamification: Beyond points, badges, and leaderboards*. [S.l.]: Octalysis Media, 2017. Citado 12 vezes nas páginas 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 e 30.

COMBÉFIS, S.; BERESNEVIČIUS, G.; DAGIENĖ, V. Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, v. 10, 2016. Disponível em: <https://ioinformatics.org/journal/v10_2016_39_60.pdf>. Citado na página 15.

CROCKFORD, D. *Introdução ao Json*. 2019. Disponível em: <https://www.json.org/json-pt.html>. Citado na página 39.

DAI, K.; ZHAO, Y.; CHEN, R. Research and practice on constructing the course of programming language. In: . [S.l.: s.n.], 2010. p. 2033 – 2038. Citado na página 14.

DANTAS, E. R. e V. O desafio da serpente - usando gamification para motivar alunos em uma disciplina introdutória de programação. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 27, n. 1, p. 577, 2016. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/6739>. Citado na página 16.

DETERDING, S. et al. From game design elements to gamefulness: Defining "gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. New York, NY, USA: ACM, 2011. (MindTrek '11), p. 9–15. ISBN 978-1-4503-0816-8. Disponível em: <http://doi-acm-org.ez54.periodicos.capes.gov.br/10.1145/2181037.2181040>. Citado na página 15.

DEVMEDIA. *Instalar e configurar o driver JDBC para Mysql*. 2007. Disponível em: <https://www.devmedia.com.br/instalar-e-configurar-o-driver-jdbc-para-mysql/6719>. Citado na página 38.

DEVMEDIA. *Introdução ao Maven*. 2012. Disponível em: <https://www.devmedia.com.br/introducao-ao-maven/25128>. Citado na página 38.

DICKEY, M. D. Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development*, v. 53, n. 2, p. 67–83, Jun 2005. ISSN 1556-6501. Disponível em: <https://doi.org/10.1007/BF02504866>. Citado na página 15.

ECLIPSE. *Jersey*. 2019. Disponível em: <https://eclipse-ee4j.github.io/jersey/>. Citado na página 38.

GILB, T.; FINZI, S. *Principles of software engineering management*. [S.l.]: Addison-wesley Reading, MA, 1988. v. 11. Citado na página 32.

HEUSER, C. A. *Projeto de banco de dados*. 4. ed. [S.l.: s.n.], 1998. Citado na página 31.

HOSTINGER. *O que é CSS?* 2019. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>. Citado na página 39.

KOLIVER; DORNELES; CASA. p. 949–960, 2004. Citado na página 18.

LEE, J.; HAMMER, J. Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, v. 15, p. 1–5, 01 2011. Citado na página 19.

MYSQL. *About MySQL*. 2019. Disponível em: <https://www.mysql.com/>. Citado na página 38.

PEREIRA, R.; COSTA, C. J.; APARICIO, J. T. *Gamification to support programming learning*. [S.l.], 2017. 1-6 p. Citado na página 14.

PRENSKY, M. Digital game-based learning. *Comput. Entertain.*, ACM, New York, NY, USA, v. 1, n. 1, p. 21–21, out. 2003. ISSN 1544-3574. Disponível em: <http://doi-acm-org.ez54.periodicos.capes.gov.br/10.1145/950566.950596>. Citado na página 15.

PRESSMAN, R. S. *Engenharia de Software: Uma abordagem profissional*. [S.l.]: bookman, 2011. v. 7. Citado 2 vezes nas páginas 32 e 33.

RAPOSO, E. H. S.; DANTAS, V. O desafio da serpente-usando gamification para motivar alunos em uma disciplina introdutória de programação. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 577. Citado na página 19.

SCHOOLS, W. *HTML5 Tutorial*. 1999. Disponível em: <https://www.w3schools.com/html/>. Citado na página 39.

SCHOOLS, W. *Java Introduction*. 1999. Disponível em: <https://www.w3schools.com/java/java_intro.asp>. Citado na página 38.

SEMEDO, M. J. M. et al. Ganhos de produtividade e de sucesso de metodologias ágeis vs metodologias em cascata no desenvolvimento de projectos de software. 2012. Citado na página 32.

SILVA, A. R. L. da et al. *Gamificação na educação*. [S.l.]: Pimenta Cultural, 2014. Citado 2 vezes nas páginas 18 e 19.

SOUZA, C. M. de. Visualg-ferramenta de apoio ao ensino de programação. *Revista Eletrônica TECCEN*, v. 2, n. 2, p. 01–09, 2009. Citado na página 16.

THAIS, H. et al. Utilizando programação funcional em disciplinas introdutórias de computação. p. 3000–69077, 07 2002. Citado na página 14.

TOLOMEI, B. A gamificação como estratégia de engajamento e motivação na educação. *EAD EM FOCO*, v. 7, n. 2, 2017. ISSN 2177-8310. Disponível em: <http://eademfoco.cecierj.edu.br/index.php/Revista/article/view/440>. Citado na página 18.

VIANNA, Y. et al. *Como reinventar empresas a partir de jogos*. [S.l.]: MJV Press, 2013. Citado na página 28.

VIEIRA, C. E. C.; JUNIOR, J. A. T. de L.; VIEIRA, P. de P. Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina de al1 do curso de sistemas de informação da faeterj–campus paracambi. *Cadernos UniFOA*, v. 10, n. 27, p. 5–15, 2015. Citado na página 16.

VUE. *Why Vue.js*. 2014. Disponível em: <https://vuejs.org/>. Citado na página 39.

VUE. *Usando Axios para Consumir APIs*. 2019. Disponível em: <https://br.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>. Citado na página 39.

Apêndices

APÊNDICE A – Primeiro Apêndice

Texto do primeiro apêndice.

APÊNDICE B – Segundo Apêndice

Texto do segundo apêndice.

Anexos

ANEXO A – Primeiro Anexo

Texto do primeiro anexo.

ANEXO B – Segundo Anexo

Texto do segundo anexo.