

# QA+: Um Jogo Para o Ensino de Teste e Qualidade de Software

Vladimir Belinski

Universidade Tecnológica Federal do Paraná (UTFPR)  
Curitiba, Paraná, Brasil

## ABSTRACT

Quality is a vital aspect in the context of software development and the testing activity is essential for its assurance. Despite the fact that testing is present in reference curricula for undergraduate courses as Computer Science, Software Engineering, etc, several studies point out to a series of factors that end up causing undergraduate students to generally demonstrate knowledge gaps on the subject, as well as to develop inappropriate testing habits. Among them can be mentioned the lack of motivation of students and the difficulty of improving their interest using only traditional teaching techniques. Thus, aiming to solve such problems, studies related to the use of serious games and gamification in the area have been carried out in recent years. With the same main objective, this work propose a serious game named QA+, which aims to support the teaching of testing and software quality, contributing to the list of existing games related to the theme and introducing new functionalities and dynamics that can be helpful.

## CCS CONCEPTS

- Applied computing → Education.

## KEYWORDS

Education; gamification; software testing; quality assurance; prototype.

### ACM Reference Format:

Vladimir Belinski. 2021. QA+: Um Jogo Para o Ensino de Teste e Qualidade de Software. In . ACM, New York, NY, USA, 15 pages.

## 1 INTRODUÇÃO

A qualidade é um aspecto vital no contexto de desenvolvimento de software [22, 29], sendo a atividade de teste fundamental para sua garantia [13, 32]. Devido a importância do teste de software este assunto é abordado em todos os currículos de referência para cursos de Sistemas e Tecnologia de Informação, Ciência da Computação e Engenharias de Computação e de Software propostos pela Sociedade Brasileira de Computação (SBC) e pela Association for Computing Machinery (ACM) juntamente a outras associações [8].

Contudo, apesar da presença desta temática nos currículos de referência, diversos estudos apontam para uma série de fatores que acabam fazendo com que alunos de graduação costumeiramente

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PPGCA - Testes, Verificação e Validação de Sistemas, 2021, Curitiba, Paraná  
© 2021 Association for Computing Machinery.

apresentem lacunas de conhecimento sobre o assunto, bem como desenvolvam hábitos de teste inadequados, o que se mostra prejudicial tanto às suas carreiras profissionais quanto às áreas de qualidade e desenvolvimento de software em si [26]. Dentre as causas comumente associadas a tais observações destacam-se: a falta de motivação dos alunos, a percepção destes de que a atividade de teste é tediosa e trivial (ou muito difícil, dependendo do enfoque dado ao assunto), a abordagem tardia e/ou superficial do tema nos cursos de graduação, a existência de diferenças entre o que é ensinado academicamente e o que é necessário na indústria, a falta de atividades práticas, a complexidade dos materiais e, considerando o emprego de abordagens tradicionais de ensino-aprendizagem, a dificuldade tanto de ensinar de modo eficaz quanto de despertar o interesse dos alunos acerca do tema [5, 13, 15, 26, 32].

Tendo conhecimento desta problemática, nos últimos anos alguns estudos têm sido realizados com o objetivo de apoiar o ensino de teste de software por meio de abordagens alternativas aos métodos e técnicas educativas tradicionalmente empregadas. Seguindo o que já vinha sendo investigado e aplicado em outras áreas de conhecimento [16] estes estudos sugerem a utilização de jogos sérios e gamificação, conceitos que se relacionam à ideia do emprego de jogos e de seus elementos visando a criação de um contexto envolvente e motivador de ensino-aprendizagem em relação a um domínio de conteúdo específico [30].

Sabendo que o emprego de jogos e gamificação na educação pode ser eficiente, aprimorando, por exemplo, a motivação e a experiência de aprendizagem dos alunos [13, 25, 26], no presente artigo será apresentada a proposta de um jogo sério nomeado QA+, o qual busca contribuir com o rol de jogos voltados ao ensino de teste e qualidade de software, de modo com que os problemas apresentados no início desta seção possam ser sanados ou, ao menos, reduzidos. Como diferencial em relação aos demais jogos já existentes pode-se destacar o fato da proposta compreender um ambiente com dinâmicas novas/próprias e que buscam possibilitar ao aluno tanto aprender quanto ensinar sobre os assuntos, disponibilizando materiais pré-definidos, mas ao mesmo tempo permitindo a criação e compartilhamento ilimitado de conteúdo pelo próprio jogador, além da aplicação de dinâmicas do jogo sobre este conteúdo personalizado. Ainda, cabe destacar que o conjunto de recursos foi planejado de modo a apoiar também atividades desempenhadas por quem ensina, o que será exposto no decorrer do trabalho.

As próximas seções deste artigo estão organizadas da seguinte maneira: na seção 2 serão expostos conceitos básicos que podem auxiliar no entendimento da proposta. Por sua vez, na seção 3 será discorrido brevemente acerca dos trabalhos relacionados. Em seguida, nas seções 4 e 5 serão apresentadas a proposta do trabalho e seus resultados preliminares, respectivamente. Por fim, na seção 6 serão destacadas as conclusões do estudo, seguidas de suas referências e apêndices.

## 2 CONCEITOS BÁSICOS

### 2.1 Jogos sérios, gamificação e aprendizagem baseada em jogos

De acordo com [7], jogos sérios consistem em jogos com finalidade educacional explícita e detalhadamente pensada, não possuindo como intenção principal a utilização para diversão. Contudo, isto não significa que a diversão não deva ocorrer, mas sim que ela deve ser utilizada para motivar o jogador a atingir os resultados de aprendizagem planejados [7, 31].

Por sua vez, segundo [14] a gamificação consiste na utilização de elementos de jogos em um dado ambiente (um contexto não-jogo [9]), objetivando motivar e aumentar o envolvimento das pessoas.

Por fim, [7] define a aprendizagem baseada em jogos como uma abordagem que emprega o jogo como ferramenta para o engajamento dos alunos em relação ao aprendizado, sendo que seu processo geralmente ocorre por meio de: (a) uma entrada, representada por um sistema que une conteúdo instrucional e elementos de jogos; (b) um processamento, caracterizado pela utilização deste sistema, o que desencadeia reações e comportamentos nos jogadores, bem como *feedbacks* do sistema; e (c) uma saída, correspondente ao alcance dos objetivos educacionais e atendimento dos resultados de aprendizagem, dados a partir do engajamento do jogador.

### 2.2 Elementos de jogos

Os elementos de jogos correspondem a pequenas partes que unidas (mas não apenas) acabam moldando um jogo, um jogo sério ou um sistema gamificado [34].

Para [34] os elementos mais importantes podem ser divididos em três categorias, sendo elas (em nível decrescente de abstração): (a) dinâmicas – aspectos abstratos do que se pretende no jogo (e.g.: emoções, narrativa, progressão, etc); (b) mecânicas – conduzem as ações dos usuários de modo a engajá-los (e.g.: desafios, recompensas, *feedback*, etc); e (c) componentes – elementos concretos e visíveis aos usuários (e.g.: avatar, nível, ponto, quadro de classificação, etc).

Por fim, de modo a facilitar o entendimento dos conceitos acima, vale apresentar o seguinte exemplo de [9, p. 26]: “recompensar (mecânica) um aluno atribuindo pontos (componente) provê a ele um senso de sua progressão (dinâmica)”. Por meio do exemplo percebe-se que os pontos são visíveis aos jogadores, enquanto o ato de recompensar e sua progressão são apenas perceptíveis. Contudo, os três se relacionam e são importantes para a construção do jogo.

## 3 TRABALHOS RELACIONADOS

Na literatura podem ser encontradas algumas abordagens e jogos voltados ao ensino de teste de software, sendo que estes variam em relação a aspectos como formato, temas abordados, entre outros.

Nos trabalhos de [3, 30, 31], por exemplo, são apresentados dois jogos físicos de cartas. No jogo criado por [3], nomeado *GreaTest Card Game*, os participantes desempenham o papel de analistas de teste, devendo indicar quais tipos de teste são mais adequados aos cenários apresentados nas cartas. Por sua vez, no jogo SOFTTY [30, 31] as cartas propõem a resolução de atividades e questões baseadas em conteúdos do nível *foundation* (básico) de certificações do *International Software Testing Qualifications Board* (ISTQB)<sup>1</sup>.

<sup>1</sup>Página oficial do ISTQB: <https://www.istqb.org/>.

Jogos que igualmente utilizam o corpo de conhecimento do ISTQB e/ou que se relacionam a suas certificações são os criados por: [1], que apresenta o *JoVeTest*, um jogo da velha digital onde o usuário possui o direito de marcar seu símbolo caso responda corretamente uma pergunta de múltipla escolha baseada em questões do ISTQB; [24], que introduz o *iLearnTest*, um jogo para o ensino de teste que explora as matérias a serem ensinadas por meio de diversas atividades (e.g.: jogo da forca, pegar conceitos corretos caindo da tela, etc) e que objetiva treinar os utilizadores para a obtenção da certificação base do ISTQB; e [28], que em seu jogo nomeado *gTest Learning* apresenta diversos questionários ao jogador, focando em testes de software com base na certificação *foundation* do ISTQB.

Como exemplos de abordagens e jogos que focam em temáticas específicas de teste pode-se mencionar: [6], que visa o ensino de testes exploratórios por meio de um *framework* gamificado no estilo caça ao tesouro; [4], que apresenta o *Bug Hide-and-Seek*, um jogo para a exposição dos alunos a testes unitários (envolve a inserção de *bugs* em programas e a criação de suítes de testes unitários); [27], que introduz o *U-TEST*, um jogo de simulação focado em teste de unidade e em técnicas de caixa-preta; [23], que visa apoiar o ensino de teste funcional por meio das três missões que compõem o jogo *BlackBox*; [13], que por meio do “Jogo das 7 falhas” igualmente busca ensinar testes de caixa-preta (nele o jogador precisa identificar 7 falhas em uma determinada funcionalidade); [26], que apresenta o *CoverBot*, um jogo que objetiva apoiar o ensino de conceitos de cobertura de testes (nele a sobrevivência do jogador depende de sua habilidade em executar todas as linhas de códigos fornecidos com a menor quantidade de entradas possível); [16], que no jogo *iTestLearning* foca na atividade de planejamento de testes (nele o jogador deve realizar o planejamento de teste de software a partir da especificação de um projeto); e [5, 17, 18, 25], que apresentam o *Code Defenders*, um jogo no qual os utilizadores competem sobre um trecho de código, gerando testes e introduzindo defeitos por meio da utilização dos princípios do teste de mutação.

Por sua vez, jogos que abordam uma quantidade maior de conceitos são os apresentados por: [32, 33], que introduz o jogo *Testing Game*, composto por três níveis e vinte e três fases que contemplam as técnicas de teste funcional e estrutural, bem como o teste baseado em defeitos (especificamente o teste de mutação); e [19], que no jogo “As aventuras de Jack Test” apresenta 4 fases (uma para cada um dos seguintes temas: teste unitário, de integração, validação e sistema) e 4 desafios (sobre garantia de qualidade de software, métodos de teste de software, processo de teste de software e depuração).

Ainda, dentre os jogos que simulam contextos empresariais se encontram os expostos em: [7, 8], que introduz o “Jogo da Equipe de Teste de Software (JETS)”, o qual aborda o tópico estratégia de teste e visa simular as interações de uma equipe de teste (em cada fase o jogador assume o papel de um membro da equipe); e [21], que no jogo *TestEG* imerge o jogador em um ambiente empresarial simulado (nele precisam ser respondidas perguntas de múltipla escolha acerca do tema).

Igualmente relacionados ao tema são os trabalhos de: [14], que propõe uma abordagem gamificada para o ensino de testes (principalmente teste funcional); [15], que apresenta o *Bug Hunt*, um tutorial web que visa estimular os estudantes a aprenderem sobre estratégias de teste de software (o aluno passa por cinco aulas onde precisa utilizar estratégias de teste específicas para encontrar erros

nos programas que lhe são apresentados; aborda conceitos de teste, técnicas de caixa-branca e caixa-preta e testes automatizados); [2], que busca ensinar a atividade de teste de software por meio da metodologia *Secret Ninja*, na qual os jogadores participam de *quests* (individuais ou em grupo), mas sem ter conhecimento do objetivo específico de aprendizagem; e [9, 11] que apresenta o *Bug Hunter*, uma abordagem gamificada para o ensino do teste composta por *quizes*, desafios e outros elementos de gamificação.

Por fim, é importante destacar que a listagem apresentada nesta seção não é exaustiva, ainda existindo outros jogos neste contexto, tais como *Mutation*, *Rings*, *EMVille* (relacionado à análise e classificação de mutantes equivalentes), *Code Hunt* e *Bug Catcher* (relacionados à criação de casos de teste), trabalhos mencionados em [14, 26], bem como abordagens adicionais, a exemplo das descritas no estudo de caracterização realizado por [10].

## 4 PROPOSTA

Nesta seção será apresentada a proposta de um jogo sério, digital e online, nomeado QA+, o qual busca apoiar o ensino de teste e qualidade de software principalmente para alunos de graduação (mas não se restringindo a este público).

Inicialmente cabe justificar a escolha do nome do jogo. QA+ deriva da associação de dois conceitos: o primeiro, a sigla QA (do inglês *Quality Assurance*), remete à “garantia de qualidade”, a respeito da qual a atividade de teste associa-se; o segundo, A+, ao conceito (no sentido de nota) que costuma denotar um desempenho excelente. Logo, a nomenclatura QA+ visa representar um jogo que auxilia os alunos a aprender e a otimizar seus conhecimentos e habilidades acerca dos assuntos propostos (testes e qualidade de software). Complementarmente destaca-se que, buscando incluir elementos alusivos à temática do jogo em seu logotipo, a letra Q foi representada graficamente como uma lupa em busca de *bugs*.

A respeito do planejamento do jogo, este foi realizado com foco em dois aspectos principais: o conteúdo a ser ensinado e as dinâmicas/elementos de jogos a serem utilizados, de modo com que seja provido um ambiente de aprendizado interessante aos alunos, mas sem desviar dos objetivos de ensino.

Acerca do conteúdo do jogo e de modo a garantir seus objetivos educacionais planejou-se: (a) a abordagem de assuntos diversos da área, não se limitando a conceitos e/ou técnicas específicas; (b) a utilização de referências bibliográficas conhecidas para a extração de conceitos e formulação de atividades tanto teóricas quanto práticas (como [12, 20, 22, 29], mas não se restringindo a elas); (c) a organização dos assuntos em unidades compostas por lições e atividades em níveis progressivos de dificuldade; (d) a estruturação das mecânicas de recompensa e das dinâmicas de progresso do jogo com base no acesso aos materiais disponibilizados e na resolução das atividades propostas; (e) o provimento constante de *feedback* e a possibilidade do jogador progredir de acordo com o seu próprio ritmo de aprendizagem; e (f) a disponibilização de recursos que permitam, além da utilização dos materiais e atividades padrão, a construção de uma base de conhecimento colaborativa, de modo com que possam ser exploradas nuances diversas dos assuntos relacionados à área.

Para a definição das dinâmicas, mecânicas e componentes de jogos a serem empregados no jogo utilizou-se principalmente a obra de [34], sendo planejado o uso de elementos como:

- Pontos/estatísticas: no jogo expressos por pontos de *bugs* (representam numericamente o progresso do jogador e são obtidos conforme este realiza atividades, nomeadas de desafios, corretamente), pontos de vitória (estatística de duelos ganhos, funcionalidade a ser exposta adiante neste trabalho), derrotas (estatística de duelos não ganhos), refis (necessários para a utilização de funcionalidades que permitem o ganho de pontos; são análogos ao conceito de “vidas”, sofrendo decréscimos caso o jogador cometa erros) e moedas (obtidas por meio da resolução correta de desafios; podem ser utilizadas para a compra de refis);
- Conquistas: relacionam-se ao alcance de objetivos predefinidos. No jogo estão associadas à visualização de lições, finalização de unidades, vitórias em duelos, etc. Ainda, de modo a estimular sua obtenção, as conquistas mais próximas de serem alcançadas pelo jogador são destacadas em áreas estratégicas, como pode ser visualizado nas Figuras 1 e 6;
- Badges: representações visuais das conquistas (no jogo expressas por quadros, como verificável na Figura 8);
- Quadro de líderes/classificação: exibição visual do progresso de todos os jogadores (disponibilizado na página principal a fim de estimular os jogadores a buscarem se destacar no *ranking*, o que é possível por meio da utilização das funcionalidades do jogo e de um bom desempenho neste);
- Desafios: tarefas que exigem esforço para resolução (visando estimular a utilização diária do jogo, disponibiliza-se diariamente uma atividade extra com viés de desafio, por meio da qual o jogador pode ganhar até 3x sua pontuação comum);
- Chance/sorte: presença de elemento randômico (nos desafios diários parte da pontuação é definida aleatoriamente, respeitando uma faixa de valores);
- Competição: mecânica onde um jogador ganha e outro perde (presente nos duelos da funcionalidade “Arena”);
- Combate: uma disputa definida e geralmente curta (presente por meio de duelos existentes na funcionalidade “Arena”);
- Feedback: informações acerca do desempenho do jogador (no QA+, após a submissão de uma resposta pelo jogador este é informado sobre a corretude ou não de sua ação, lhe sendo igualmente apresentada a resposta esperada);
- Avatar: representação visual do personagem do jogador (passível de escolha na página de perfil);
- Desbloqueio de conteúdo: realizado quando o jogador atinge objetivos (no jogo são desbloqueadas unidades e conquistas de acordo com o progresso do jogador); e
- Níveis: etapas definidas na progressão do jogador (no jogo proposto são explicitados nas conquistas).

Complementarmente, cabe destacar que planejou-se para o QA+ a existência de:

- Uma tela de cadastro e login no jogo, com recursos comuns a telas com esta finalidade;
- Uma tela principal (verificável na Figura 1), onde seria exposta a próxima conquista do jogador, suas estatísticas individuais, um resumo do quadro de líderes, bem como opções de acesso ao desafio diário e às unidades de ensino-aprendizagem (as quais são compostas por: (a) lições com

- conteúdo multimídia e referências; e (b) conjuntos de atividades/desafios;
- Uma funcionalidade de “Turmas” (verificável na Figura 4), permitindo ao jogador criar, buscar e participar de turmas adicionais (além da disponibilizada por padrão no jogo). Ao criar uma turma seria possível definir uma imagem para ela, seu nome e conteúdo (sendo possível inclusive importar o conteúdo da turma padrão do QA+). Já ao acessar uma turma criada por outro usuário seria apresentada uma página semelhante à da Figura 1, mas com os conteúdos personalizados, a identificação de qual turma está sendo exibida e uma opção para sair. Por sua vez, ao acessar uma turma criada pelo próprio usuário este seria direcionado para uma página de gestão (a exemplo do que é apresentado na Figura 5), onde seriam disponibilizadas funcionalidades para a manutenção da turma, bem como para o acompanhamento do desempenho dos participantes (tanto na turma QA+ quanto na turma adicional, permitindo, assim, que docentes possam acompanhar o processo de aprendizado de seus alunos de modo individualizado). Ao possibilitar a utilização de recursos do jogo sobre conjuntos personalizados de conteúdo esta funcionalidade expande as possibilidades de obtenção de conhecimento;
  - Uma funcionalidade intitulada “Arena” (verificável na Figura 6), por meio da qual seria possibilitado aos jogadores competir entre si em disputas de cinco minutos (nomeadas de duelos) envolvendo a resolução de desafios e com o objetivo de acumular pontos e desbloquear conquistas;
  - Uma tela de perfil (verificável na Figura 8), na qual o jogador poderia manter sua conta, alterar seu avatar, visualizar suas estatísticas completas, bem como suas conquistas.

Ainda, ressalta-se que foi planejado um *design* simples e intuitivo para o jogo, tendo sido utilizado como referência para tal o jogo Duolingo, o qual é voltado ao aprendizado de novos idiomas e pode consistir em um exemplo prático de gamificação [9].

## 5 RESULTADOS PRELIMINARES

Como resultados preliminares pode-se destacar a criação de um protótipo executável do jogo e de artefatos (como diagramas de fluxo de negócios) que buscam auxiliar na compreensão de sua lógica.

A seguir serão expostas algumas telas elaboradas para o jogo, de modo a apresentar uma visão geral deste. Por sua vez, o protótipo executável do QA+ pode ser encontrado clicando aqui. Ainda, nos Apêndices A, B e C podem ser encontrados os requisitos do sistema (funcionais e não funcionais), os diagramas de fluxo de negócio relacionados aos desafios e duelos do jogo e as principais telas prototipadas, respectivamente.

Inicialmente, na Figura 1 pode ser visualizada a tela principal do jogo. Complementando o que foi apresentado na Seção 4, cabe mencionar que as unidades são representadas com símbolos de lupas nas cores: (a) cinza: representa que a unidade está bloqueada; (b) amarelo e branco: representa que a unidade está desbloqueada (a borda vai sendo colorida em dourado conforme é progredido no material disponibilizado; caso o centro da lupa apresente uma “rachadura” na cor branca significa que a unidade precisa ser reforçada, pois sua conclusão ocorreu há mais de 30 dias); e (c) amarelo e dourado (somente): representa que a unidade foi finalizada.

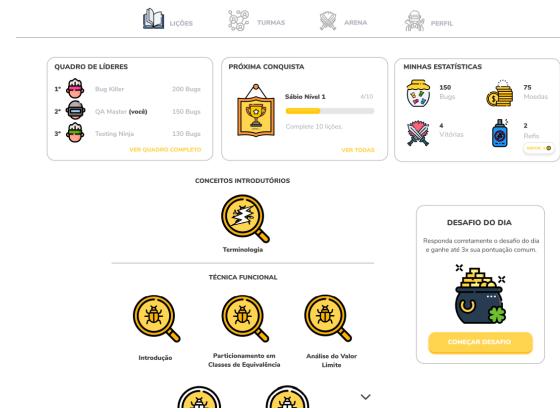


Figura 1: Tela principal do jogo QA+.

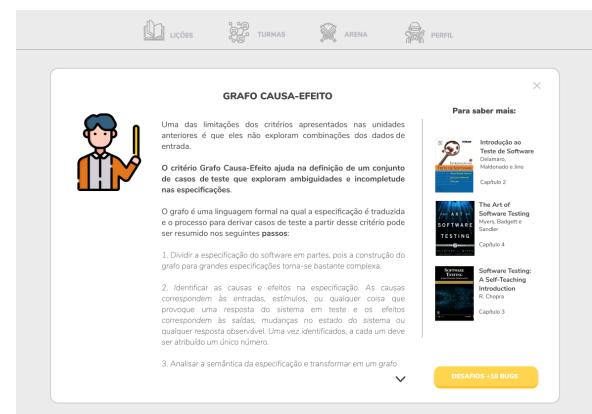


Figura 2: Exemplo de lição no jogo QA+.



Figura 3: Exemplo de atividade/desafio no jogo QA+.

Ao clicar sobre uma unidade desbloqueada são apresentadas opções para: (a) visualização da lição associada: um conjunto de conteúdo multimídia e referências, como pode ser observado na

Figura 2; e (b) realização das atividades propostas: uma sequência de questões teóricas e práticas (no estilo múltipla escolha, *drag and drop*, verdadeiro e falso, etc, de modo a permitir correções automáticas) por meio das quais o jogador pode ganhar pontos de *bugs* e moedas. Para evitar chutes cada resposta incorreta acarreta na perda de um refil, sendo que a cada resposta o jogador obtém *feedback* acerca de seu desempenho, bem como é informado acerca da opção correta (como pode ser verificado na Figura 3).

Figura 4: Tela geral da funcionalidade “Turmas”.

Figura 5: Exemplo de tela de gestão de turma.

Conforme já mencionado na Seção 4, nas Figuras 4 e 5 podem ser visualizadas a página geral da funcionalidade de turmas (com opções de busca, criação e acesso à turmas) e o exemplo de uma página de gestão de turma (com links para funcionalidades diversas de gestão, bem como um quadro de estatísticas dos inscritos), respectivamente.

Por sua vez, nas Figuras 6 e 7 podem ser visualizadas a tela principal da funcionalidade “Arena” (com explicações, estatísticas do jogador, opção de duelar e próxima conquista relacionada) e um exemplo de duelo (destacando o placar, a contagem de tempo, um exemplo de desafio proposto e o *feedback* do sistema), nesta ordem, as quais seguem a dinâmica exposta na Seção 4.

Figura 6: Tela geral da funcionalidade “Arena”.

Figura 7: Exemplo de parte de um duelo no jogo QA+.

Por fim, na Figura 8 pode ser visualizada a tela de perfil do usuário, onde é perceptível a existência de seus dados, estatísticas, conquistas, bem como opções de manutenção da conta e realização de logout do jogo.

Figura 8: Tela de perfil no jogo QA+.

## 6 CONCLUSÕES

Neste trabalho foi apresentada a proposta de um jogo sério nomeado QA+, o qual busca apoiar o ensino de teste e qualidade de software principalmente para alunos de graduação.

No artigo focou-se especialmente no planejamento do jogo (tanto a respeito de seu conteúdo quanto das dinâmicas, mecânicas e componentes de jogos a serem aplicados) e nos artefatos já elaborados (a exemplo do protótipo executável). Espera-se que com a implementação e posterior uso do jogo problemas tais como os apresentados na Seção 1 possam ser superados ou, ao menos, reduzidos, de modo com que os alunos aprendam efetivamente sobre as temáticas abordadas e consigam empregar adequadamente o conhecimento adquirido em projetos reais de desenvolvimento de software.

Por fim, como trabalhos futuros poderão ser realizados: (a) a prototipação de telas para todos os recursos previstos para a funcionalidade de turmas; (b) a validação da proposta com as partes interessadas (alunos e docentes da área); (c) a implementação do jogo; e (d) a realização de estudos de aplicação do QA+ de modo a verificar sua eficácia/atendimento dos objetivos propostos.

## REFERÊNCIAS

- [1] Ana Karoline T. Barbosa, Larissa L. E. Neves, and Arilo C. Dias-Neto. 2016. JoVeTest - Jogo da Velha para Auxiliar no Ensino e Estudo de Teste de Software. *IX Fórum de Educação em Engenharia de Software* (2016). <http://experts.icomp.ufam.edu.br/jovetest/>
- [2] Jonathan Bell, Swapneel Sheth, and Gail Kaiser. 2011. Secret Ninja Testing with HALO Software Engineering. In *Proceedings of the 4th International Workshop on Social Software Engineering* (Szeged, Hungary) (*SSE '11*). Association for Computing Machinery, New York, NY, USA, 43–47. <https://doi.org/10.1145/2024645.2024657>
- [3] Thiago A. Beppe, Ítalo Linhares de Araújo, Bruno Sabóia Aragão, Ismayle de Sousa Santos, Davi Ximenes, and Rossana M. Castro Andrade. 2018. GreaTest: A Card Game to Motivate the Software Testing Learning (*SBES '18*). Association for Computing Machinery, New York, NY, USA, 298–307. <https://doi.org/10.1145/3266237.3266254>
- [4] Kevin Buffardi and Pedro Valdivia. 2018. Bug Hide-and-Seek: An Educational Game for Investigating Verification Accuracy in Software Tests. In *2018 IEEE Frontiers in Education Conference (FIE)*. 1–8. <https://doi.org/10.1109/FIE.2018.8658748>
- [5] Benjamin S. Clegg, Jose Miguel Rojas, and Gordon Fraser. 2017. Teaching Software Testing Concepts Using a Mutation Testing Game. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. 33–36. <https://doi.org/10.1109/ICSE-SEET.2017.1>
- [6] Igor Ernesto Ferreira Costa and Sandro Ronaldo Bezeria Oliveira. 2020. The use of gamification to support the teaching-learning of software exploratory testing: an experience report based on the application of a framework. In *2020 IEEE Frontiers in Education Conference (FIE)*. 1–9. <https://doi.org/10.1109/FIE44824.2020.9273943>
- [7] Tarcila Gesteira da Silva. 2012. *Jogos sérios em mundos virtuais: uma abordagem para o ensino-aprendizagem de teste de software*. Dissertação, Universidade Federal de Santa Maria.
- [8] Tarcila Gesteira da Silva, Giliane Bernardi, and Felipe Müller. 2011. Abordagem de Apoio ao Ensino e Aprendizagem de Teste de Software Baseada em Jogos Sérios e Mundos Virtuais. *Anais do XXII Simpósio Brasileiro de Informática na Educação - XVII WIE* (01 2011), 538–541.
- [9] Gabriela Martins de Jesus. 2019. *Explorando Gamificação no Ensino de Teste de Software*. Dissertação, Universidade Federal de São Carlos.
- [10] Gabriela Martins de Jesus, Fabiano Cutigi Ferrari, Daniel de Paula Porto, and Sandra Camargo Pinto Ferraz Fabbri. 2018. Gamification in Software Testing: A Characterization Study. In *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing* (SAO CARLOS, Brazil) (*SAST '18*). Association for Computing Machinery, New York, NY, USA, 39–48. <https://doi.org/10.1145/3266003.3266007>
- [11] Gabriela Martins de Jesus, Leo Natan Paschoal, Fabiano Cutigi Ferrari, and Simone R. S. Souza. 2019. Is It Worth Using Gamification on Software Testing Education? An Experience Report. In *Proceedings of the XVIII Brazilian Symposium on Software Quality* (Fortaleza, Brazil) (*SBQS'19*). Association for Computing Machinery, New York, NY, USA, 178–187. <https://doi.org/10.1145/3364641.3364661>
- [12] Márcio Eduardo Delamaro, José Carlos Maldonado, and Mario Jino. 2007. *Introdução ao teste de software* (1ª ed.). Elsevier.
- [13] Lucio L. Diniz and Rudimar L. S. Dazzi. 2011. Jogo para o Apoio ao Ensino do Teste de Caixa-Preta. In *Jogo para o Apoio ao Ensino do Teste de Caixa-Preta*.
- [14] Eduardo Henrique Rocha do Nascimento. 2019. *Aplicando Gamificação no Ensino de Teste de Software*. Dissertação, Universidade Federal do Rio Grande do Norte.
- [15] Sebastian Elbaum, Suzette Person, Jon Dokulil, and Matt Jorde. 2007. Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable. In *29th International Conference on Software Engineering (ICSE'07)*. 688–697. <https://doi.org/10.1109/ICSE.2007.23>
- [16] Virginia Farias, Carla Moreira Bezerra, Emanuel Coutinho, and Ismayle de Sousa Santos. 2012. iTest Learning: Um Jogo para o Ensino do Planejamento de Testes de Software.
- [17] Gordon Fraser. 2017. Gamification of Software Testing. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. 2–7. <https://doi.org/10.1109/AST.2017.20>
- [18] Gordon Fraser, Alessio Gambi, and José Miguel Rojas. 2020. Teaching Software Testing with the Code Defenders Testing Game: Experiences and Improvements. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 461–464. <https://doi.org/10.1109/ICSTW50294.2020.00082>
- [19] Kely Santos Macêdo. 2014. *As aventuras de Jack Test: jogo educacional para o apoio ao ensino de teste de software*. Trabalho de Conclusão de Curso, Universidade Estadual do Sudoeste da Bahia.
- [20] Glenford J. Myers, Tom Badgett, and Corey Sandler. 2012. *The Art of Software Testing* (3rd ed.). John Wiley & Sons, Inc.
- [21] Bruno Oliveira, Paulo Afonso, and Heitor Costa. 2016. TestEG – A computational game for teaching of software testing. In *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*. 1–10. <https://doi.org/10.1109/SCCC.2016.7836022>
- [22] Roger S. Pressman. 2015. *Software engineering: a practitioner's approach* (8th ed.). McGraw-Hill Education, New York, USA.
- [23] Nayara Ribeiro, Rosiane Vasconcelos, Davi Viana, and Luis Rivero. 2017. Avaliando a Viabilidade do BlackBox em Sala de Aula: Um Jogo Sério para Ensino de Teste Funcional de Software. In *Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE 2017)*. <https://doi.org/10.5753/cbie.sbie.2017.817>
- [24] Tânia P. B. Ribeiro and Ana C. R. Paiva. 2015. iLearnTest: Educational game for learning software testing. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–6. <https://doi.org/10.1109/CISTI.2015.7170608>
- [25] José Miguel Rojas and Gordon Fraser. 2016. Code Defenders: A Mutation Testing Game. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 162–167. <https://doi.org/10.1109/ICSTW.2016.43>
- [26] Eman Sherif, Andy Liu, Brian Nguyen, Sorin Lerner, and William G. Griswold. 2020. Gamification to Aid the Learning of Test Coverage Concepts. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE T)*. 1–5. <https://doi.org/10.1109/CSEE49119.2020.9206224>
- [27] Antonio Carlos Silva. 2010. *Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade*. Dissertação, Universidade do Vale do Itajai.
- [28] Cleibson Gomes Silva, Ivaldir de Farias Junior, Marcelo Teixeira, and Cristiane de Aquino. 2016. gTest Learning: Um Jogo para Ensino Básico de Teste de Software. In *Congresso Regional sobre Tecnologias na Educação*.
- [29] Ian Sommerville. 2011. *Software Engineering* (9th ed.). Addison-Wesley, Massachusetts, USA.
- [30] Alexander Soska, Jürgen Mottok, and Christian Wolff. 2016. An experimental card game for software testing: Development, design and evaluation of a physical card game to deepen the knowledge of students in academic software testing education. In *2016 IEEE Global Engineering Education Conference (EDUCON)*. 576–584. <https://doi.org/10.1109/EDUCON.2016.7474609>
- [31] Alexander Soska, Jürgen Mottok, and Christian Wolff. 2017. Pattern oriented card game development: SOFTTY – A card game for academic learning of software testing. In *2017 IEEE Global Engineering Education Conference (EDUCON)*. 1166–1173. <https://doi.org/10.1109/EDUCON.2017.7942996>
- [32] Pedro Henrique Valle, Rafaela Rocha, and José Maldonado. 2017. Testing Game: An Educational Game to Support Software Testing Education. In *SBES: Brazilian Symposium on Software Engineering*. 289–298. <https://doi.org/10.1145/3131151.3131182>
- [33] Pedro Henrique Dias Valle. 2017. *Jogos educacionais: uma contribuição para o ensino de teste de software*. Dissertação, Universidade de São Paulo.
- [34] Kevin Werbach and Dan Hunter. 2012. *For the Win: How Game Thinking Can Revolutionize Your Business* (1st ed.). Wharton Digital Press, Pennsylvania, USA.

## A REQUISITOS DO SISTEMA

### A.1 Requisitos Funcionais

- RF1: O sistema deve permitir que um usuário crie uma conta no jogo por meio do informe de nome, usuário, e-mail e senha.
- RF2: O sistema deve apresentar uma página de login que permita o acesso ao jogo por meio do informe de: (1) e-mail ou nome de usuário e (2) senha.
- RF3: O sistema deve disponibilizar na página de login um recurso para recuperação de senha.
- RF4: O sistema deve apresentar um esquema de recompensas aos usuários baseado em pontos (de tipos nomeados bugs, moedas e vitórias/derrotas) e badges (nomeados de conquistas).
- RF5: O sistema deve apresentar uma tela principal de lições contendo um resumo do quadro de líderes (RF6, RF7, RF8), a indicação da próxima conquista a ser alcançada pelo usuário (RF9, RF10), um resumo das estatísticas do usuário (RF11, RF12), as unidades de ensino-aprendizagem do jogo (RF14, RF15, RF16) e um quadro relacionado à funcionalidade de desafio diário (RF19).
- RF6: O quadro de líderes deve apresentar um ranking (em ordem decrescente) dos usuários com a maior quantidade de pontos de bugs no momento, devendo ser apresentadas nele as informações de ordem de classificação, avatar, nome e total de pontos de bugs do usuário.
- RF7: A posição em que o usuário logado no sistema se encontra no quadro de líderes deve ser destacada para facilitar sua localização.
- RF8: No quadro de líderes apresentado na tela principal deverão ser expostas minimamente as informações: do usuário classificado na 1ª posição, do usuário logado, do usuário que o antecede no ranking (se houver) e do usuário que o sucede no ranking (se houver).
- RF9: Para cada conquista possível de obtenção pelo usuário devem ser apresentados: título e imagem ilustrativa da conquista, descrição do que é necessário para obtê-la, indicadores numéricos e ilustrativos do progresso do usuário em relação à obtenção da conquista.
- RF10: A próxima conquista geral a ser alcançada pelo usuário deve ser apresentada em um quadro no topo da tela de lições, sendo igualmente disponibilizado um botão de acesso à visualização de todas as conquistas (ao clicar neste o usuário deve ser redirecionado à sua página de perfil (RF29)).
- RF11: O resumo de estatísticas do usuário deve apresentar para cada estatística (Pontos de bugs, moedas, vitórias e refis) seu nome, imagem ilustrativa e saldo atual do usuário.
- RF12: Abaixo da estatística de refis deve ser apresentado um botão para reposição. Quando este for acionado deverão ser atribuídos 5 refis e consumidas 5 moedas do usuário.
- RF13: Uma vez ao dia, caso o saldo de refis do usuário seja inferior a 5, 1 refil será acrescentado automaticamente ao seu saldo de refis.
- RF14: As unidades de ensino-aprendizagem do jogo devem ser organizadas em seções, sendo que cada seção deve ser identificada por um título.
- RF15: Cada unidade de ensino-aprendizagem deve ser identificada por um título e um símbolo de lupa em uma das seguintes

**combinações de cores:** (a) cinza: se a unidade estiver bloqueada; (b) amarelo e branco: se a unidade estiver desbloqueada (a borda deve ser colorida em dourado conforme o usuário progride no material disponibilizado; deve ser apresentado no centro da lupa uma “rachadura” na cor branca caso a conclusão da unidade tenha ocorrido a mais de 30 dias); ou (c) amarelo e dourado (somente): se a unidade tiver sido concluída pelo usuário em período menor ou igual a 30 dias.

- RF16: Cada unidade de ensino-aprendizagem deve ser composta por uma lição (RF17) e um conjunto de atividades (nomeados desafios; RF18). Ao se clicar em uma unidade desbloqueada deverão ser apresentadas opções de acesso à sua lição e atividades relacionadas.
- RF17: Cada lição deve ser composta por um título, conteúdos em formato multimídia, uma lista de referências e uma opção para acesso aos desafios da unidade.
- RF18: Cada conjunto de desafios deve apresentar uma série de atividades ao usuário (com correção automática) relacionadas ao conteúdo da unidade. Por meio da resolução das atividades o usuário poderá acumular pontos e moedas (proporcionais à sua quantidade de acertos). O fluxo dos desafios deve seguir o disposto na Figura 11.
- RF19: O quadro relacionado à funcionalidade de desafio diário deve apresentar um título, explicação, ícone e opção para começar o desafio (habilitada somente caso o usuário ainda não tenha clicado nela no dia em questão). Nesta funcionalidade deve ser possibilitado que o usuário responda um desafio extra (intitulado de “desafio diário”) por dia. Caso o usuário acerte a resposta do desafio, além de sua pontuação normal ele poderá ganhar um adicional de pontos (nomeado “Pontos da sorte”) de valor até três vezes superior à pontuação normal do desafio (pontuação gerada aleatoriamente). O fluxo do desafio diário deve seguir o disposto na Figura 9.
- RF20: O sistema deve disponibilizar uma página nomeada “Turmas”, a qual deve permitir com que o usuário crie, busque e participe de turmas adicionais (além da turma padrão QA+). Por meio desta página deve ser possível acessar funcionalidades de: (a) criação de turma (RF21); (b) gerenciamento de turma criada pelo próprio usuário (RF22); (c) acesso à turma adicional em que o usuário já participa (RF25); (d) busca de turmas por termo; e (e) solicitação de acesso/ingresso em turma criada por outro usuário (RF26).
- RF21: A funcionalidade de criação de turma deve disponibilizar recursos para informe de nome, imagem e situação da turma, bem como para importação dos dados padrão da turma QA+, exportação dos dados da turma, adição de nova seção, nova unidade, nova lição, novo desafio, edição e remoção de dados.
- RF22: A funcionalidade de gerenciamento de turma deve apresentar um painel de controle (RF23) e um quadro de estatísticas dos inscritos (RF24).
- RF23: O painel de controle de turma deve apresentar opções para acesso a funcionalidades de: (a) manutenção dos dados da turma (edição dos mesmos campos do RF21); (b) exclusão da turma; (c) manutenção das solicitações pendentes de ingresso na turma (visualização, aceite e negativa); (d) manutenção dos inscritos (busca, visualização e remoção da turma); (e) visualização da turma como inscrito (visão descrita no RF25);

- (f) manutenção de seções, unidades, lições e desafios; e (g) manutenção de conquistas.
- RF24: O quadro de estatísticas de inscritos em uma turma deve apresentar campos para busca de inscrito, ordenação dos resultados por tipo de estatística (total de bugs, total de vitórias ou total de conquistas) e sentido (crescente ou decrescente), bem como uma grid na qual devem ser listados os nomes dos inscritos, seus usuários, links para seus perfis nas turmas QA+ e minha turma, estatísticas nas turmas QA+ e minha turma e totalizadores gerais das estatísticas.
- RF25: Ao acessar uma turma adicional será disponibilizado ao usuário seus recursos, sendo que esta difere da turma padrão QA+ ao passo que: (a) apresenta conteúdo personalizado; (b) apresenta uma barra com o nome da turma e seu criador, bem como opção para deslogar; e (c) não apresenta acesso à funcionalidade de turmas.
- RF26: Ao realizar uma solicitação de acesso em turma o usuário será listado na funcionalidade “Solicitações Pendentes de Ingresso na Turma” na página de gestão da turma em questão.
- RF27: O sistema deve apresentar uma página nomeada “Arena”, a qual deverá conter explicações sobre esta funcionalidade, opção para iniciar duelo (RF28) e quadros de estatísticas e próxima conquista do usuário na funcionalidade.
- RF28: O sistema deve apresentar uma funcionalidade de duelos, por meio da qual os jogadores podem competir entre si em disputas de cinco minutos que envolvam a resolução de desafios e acarretem na acumulação de pontos e desbloqueio de conquistas. O fluxo/comportamento de um duelo deve seguir o disposto na Figura 10.
- RF29: O sistema deve apresentar uma página de perfil que possibilite ao usuário atualizar seus dados de cadastro, escolher um avatar, visualizar suas estatísticas de desempenho e conquistas obtidas no jogo (RF9), excluir sua conta e realizar logout.

## A.2 Requisitos Não Funcionais

- RNF1: O sistema deve atender o que se encontra disposto na Lei nº 13.709/2018 (Lei Geral de Proteção de Dados - LGPD).
- RNF2: O sistema deve estar disponível na internet, sendo acessível minimamente a partir dos navegadores Google Chrome, Mozilla Firefox, Microsoft Edge e Safari.
- RNF3: O sistema deve ser disponibilizado minimamente na língua portuguesa.
- RNF4: O sistema deve possuir uma interface intuitiva e de fácil utilização, permitindo novos usuários navearem com o mínimo de esforço ou treinamento.
- RNF5: O acesso ao jogo deve ser restrito a usuários autenticados no sistema.
- RNF6: O sistema deve estar disponível para acesso 24 horas por dia e 7 dias por semana.
- RNF7: O tempo de resposta para a apresentação do quadro de estatísticas dos inscritos (em uma turma) não pode ser superior a 15 segundos.

## B DIAGRAMAS DE FLUXO DE NEGÓCIO DAS FUNCIONALIDADES PRINCIPAIS

Neste apêndice serão apresentados os diagramas de fluxo de negócios relacionados aos desafios e funcionalidade de duelo do jogo.

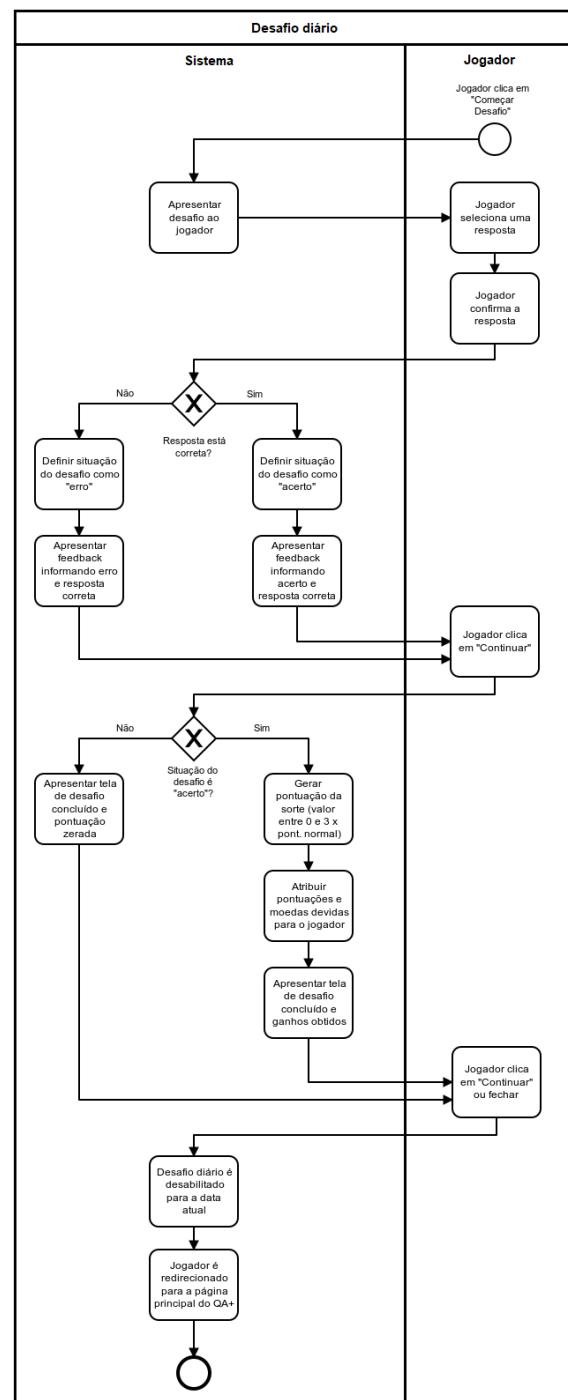


Figura 9: Fluxo de um desafio diário.

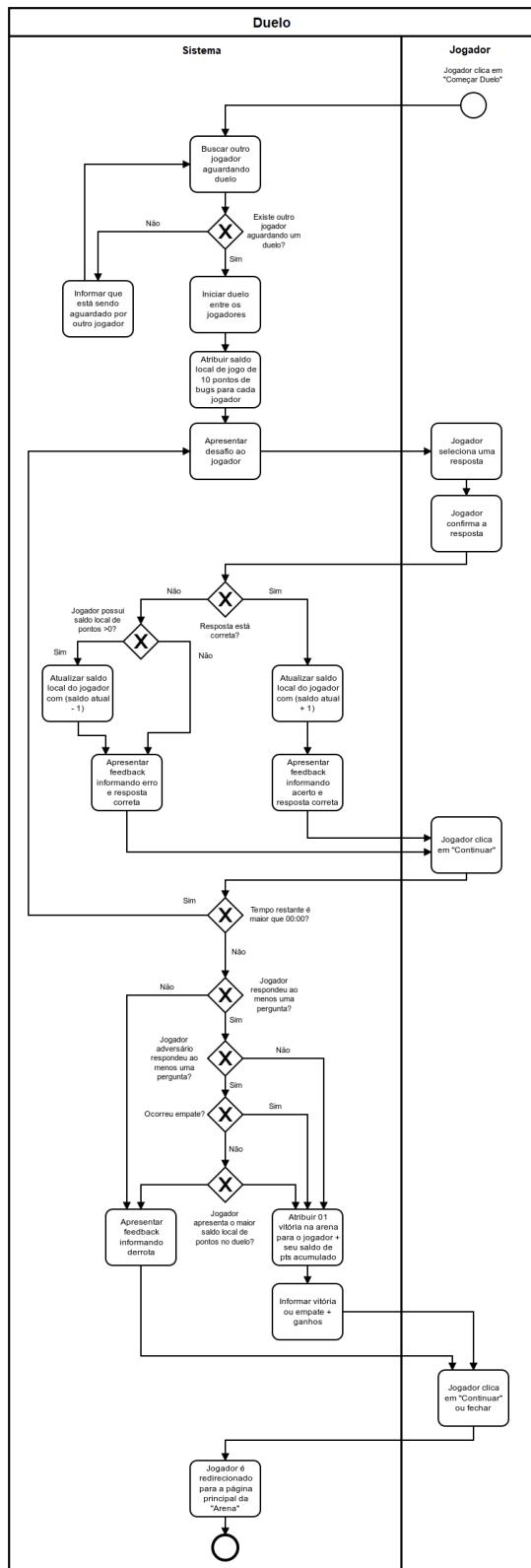


Figura 10: Fluxo de um duelo.

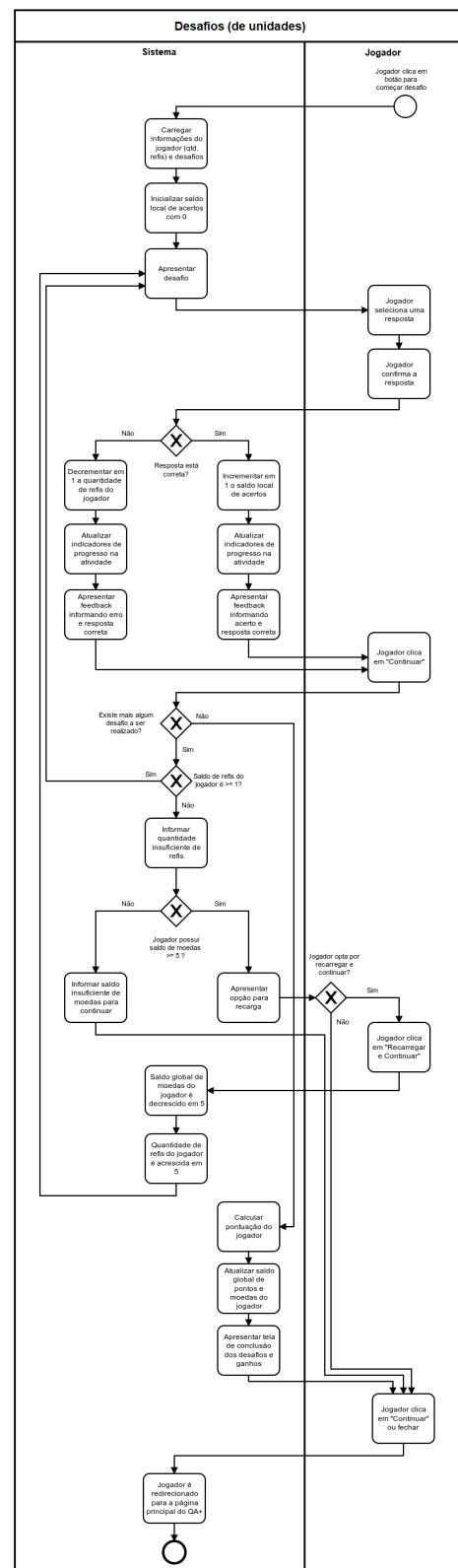


Figura 11: Fluxo de desafios de unidades.

## C PROTÓTIPOS CRIADOS

Neste apêndice serão apresentadas as principais telas criadas para o jogo (utilizadas no protótipo executável).



Figura 12: Tela de criação de conta.



Figura 13: Tela de login.

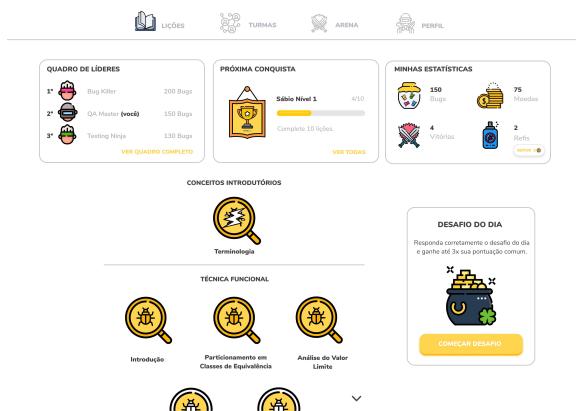


Figura 14: Tela principal do QA+.

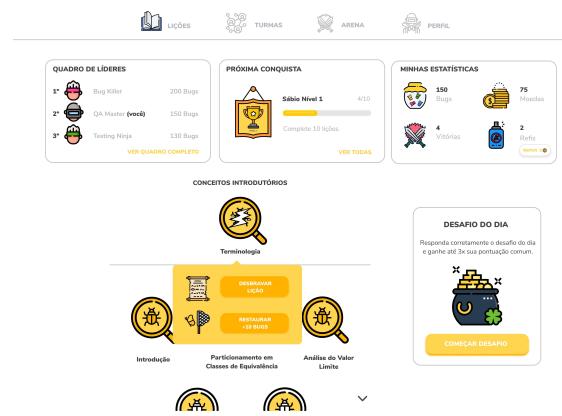


Figura 15: Tela principal - Restaurar unidade.

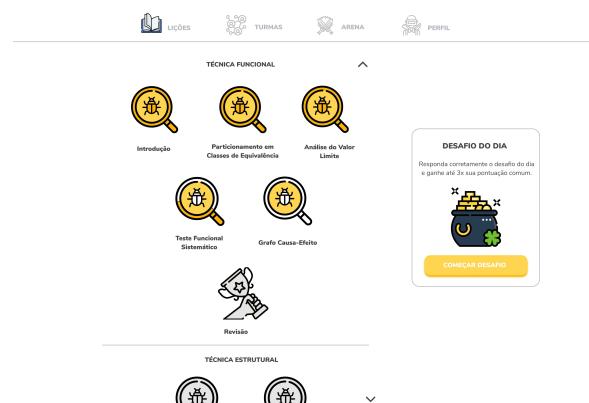


Figura 16: Tela principal - Unidades bloqueadas.

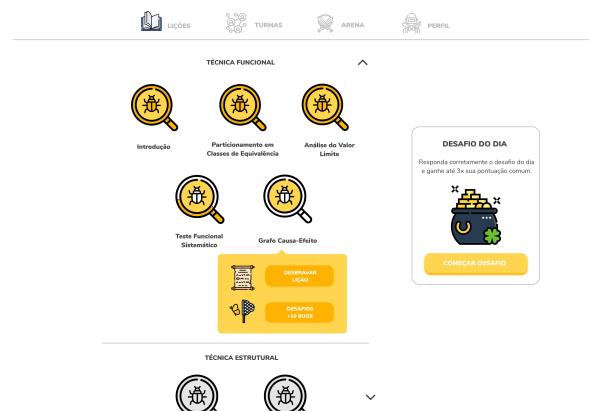


Figura 17: Tela principal - Acessar lição e desafios.

**GRAFO CAUSA-EFEITO**

Uma das limitações dos critérios apresentados nas unidades anteriores é que eles não exploram combinações dos estados de entradas.

O critério Grafo Causa-Efeito ajuda na definição de um conjunto de casos de teste que exploram ambiguidades e incompletude nas especificações.

O grafo é uma linguagem formal na qual a especificação é traduzida e o processo para derivar casos de teste a partir desse critério pode ser resumido nos seguintes passos:

1. Dividir a especificação do software em partes, pois a construção do grafo para grandes especificações torna-se bastante complexa.
2. Identificar as causas e efeitos na especificação. As causas correspondem às entradas, estados, ou qualquer coisa que provoca uma reação no sistema, e os efeitos correspondem às saídas mutantes ou teste do sistema, ou qualquer resposta observável. Uma vez identificados, a causa é atribuída um único número.
3. Analisar a semântica da especificação e transformar em um grafo.

**PARA SABER MAIS:**

- Introdução ao Teste de Software (Defenses, Matérias e Projetos)
- The Art of Software Testing (Software Testing: A Craft of Quality Assurance)
- Software Testing: A Craft-Teaching Introduction (R. Churas)

**DESAFIOS > 10 BUGS!**

Figura 18: Exemplo de lição.

**Escolha a opção correta:**

O processo para derivar casos de teste a partir do critério Grafo Causa-Efeito pode ser resumido em alguns passos, cuja ordem de execução é:

- | Convert o grafo em uma tabela de decisão, na qual cada coluna representa um caso de teste.
- | Dividir a especificação do software em partes.
- | Analisar a semântica da especificação e tirar um grafo booleano.
- | Converter as colunas da tabela de decisão em casos de teste.
- | Identificar as causas e efeitos na especificação e a cada uma delas atribuir um número único.
- | Adicionar anotações ao grafo descrevendo combinações de causas e efeitos impossíveis devido restrições sintáticas ou do ambiente.

**Ops! Resposta incorreta!**  
A resposta correta é 5 - 1 - 3 - 6 - 2 - 4.

**CONTINUAR**

Figura 21: Exemplo de feedback para resposta incorreta.

**Escolha a opção correta:**

O processo para derivar casos de teste a partir do critério Grafo Causa-Efeito pode ser resumido em alguns passos, cuja ordem de execução é:

- | Convert o grafo em uma tabela de decisão, na qual cada coluna representa um caso de teste.
- | Dividir a especificação do software em partes.
- | Analisar a semântica da especificação e tirar um grafo booleano.
- | Converter as colunas da tabela de decisão em casos de teste.
- | Identificar as causas e efeitos na especificação e a cada uma delas atribuir um número único.
- | Adicionar anotações ao grafo descrevendo combinações de causas e efeitos impossíveis devido restrições sintáticas ou do ambiente.

**VERIFICAR**

Figura 19: Exemplo 01 de desafio - Pergunta.

**Ligue os pares:**

Ligue cada operador lógico à sua respectiva notação:

<b>OR</b>	<b>AND</b>
<b>NOT</b>	<b>Idiomatic</b>

(a) (b)

(c) (d)

**VERIFICAR**

Figura 22: Exemplo 02 de desafio - Pergunta.

**Escolha a opção correta:**

O processo para derivar casos de teste a partir do critério Grafo Causa-Efeito pode ser resumido em alguns passos, cuja ordem de execução é:

- | Convert o grafo em uma tabela de decisão, na qual cada coluna representa um caso de teste.
- | Dividir a especificação do software em partes.
- | Analisar a semântica da especificação e tirar um grafo booleano.
- | Converter as colunas da tabela de decisão em casos de teste.
- | Identificar as causas e efeitos na especificação e a cada uma delas atribuir um número único.
- | Adicionar anotações ao grafo descrevendo combinações de causas e efeitos impossíveis devido restrições sintáticas ou do ambiente.

**VERIFICAR**

Figura 20: Exemplo 01 de desafio - Seleção de resposta.

**Ligue os pares:**

Ligue cada operador lógico à sua respectiva notação:

<b>OR</b>	<b>AND</b>
<b>NOT</b>	<b>Idiomatic</b>

(a) (b)

(c) (d)

**VERIFICAR**

Figura 23: Exemplo 02 de desafio - Seleção de resposta.

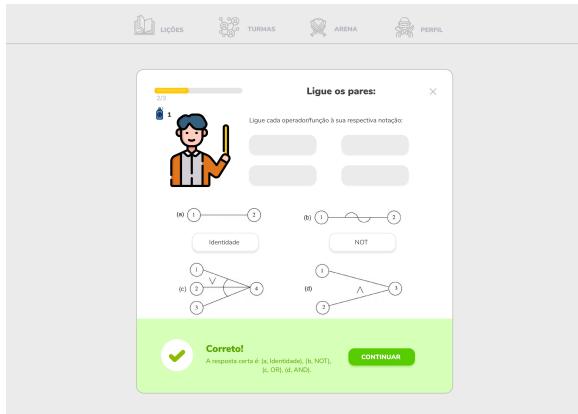


Figura 24: Exemplo de feedback para resposta correta.

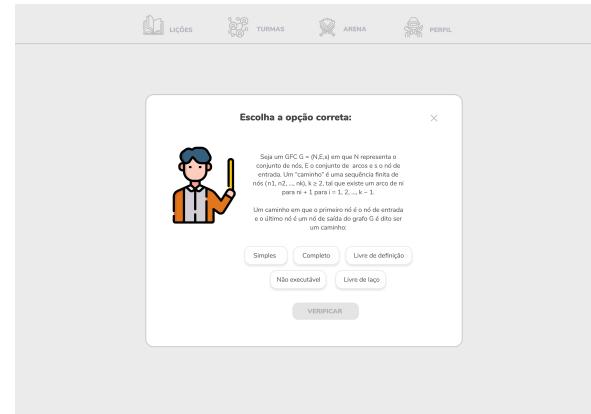


Figura 27: Exemplo de desafio diário - Pergunta.

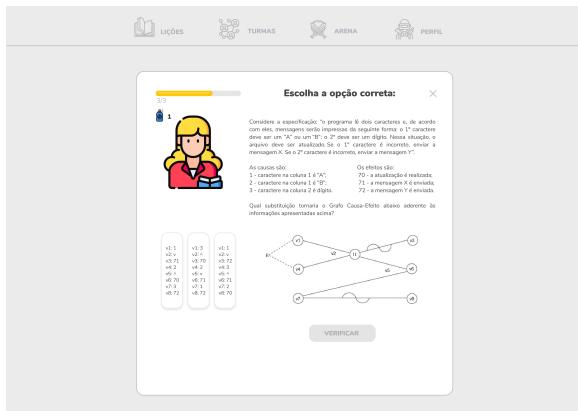


Figura 25: Exemplo 03 de desafio - Pergunta.

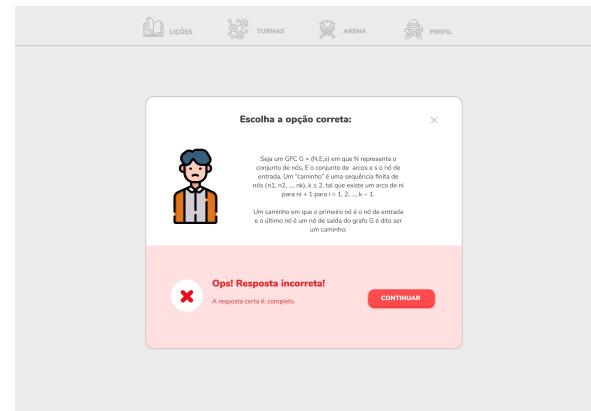


Figura 28: Desafio diário - Feedback resposta incorreta.

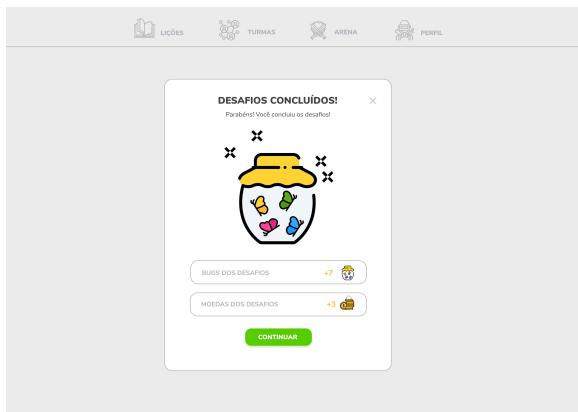


Figura 26: Conclusão de desafio.

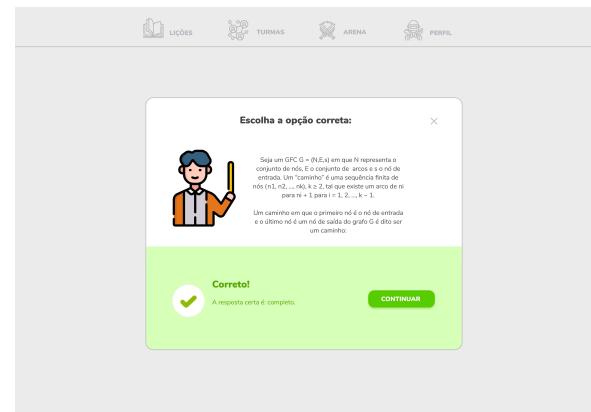


Figura 29: Desafio diário - Feedback resposta correta.

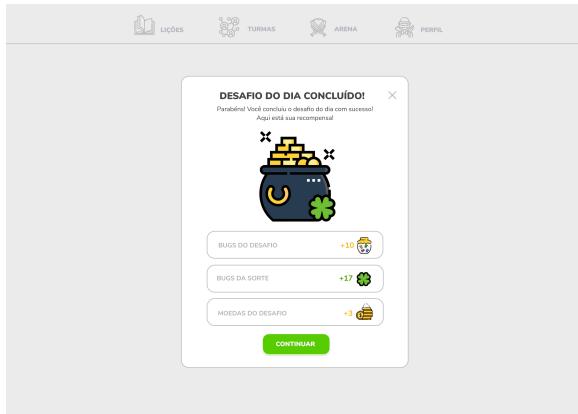


Figura 30: Conclusão de desafio diário - Resposta correta.

Figura 33: Tela geral da funcionalidade “Turmas”.

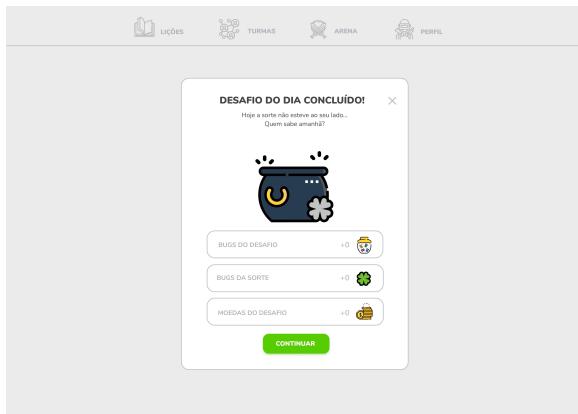


Figura 31: Conclusão de desafio diário - Resposta incorreta.

Figura 34: Criar nova turma.

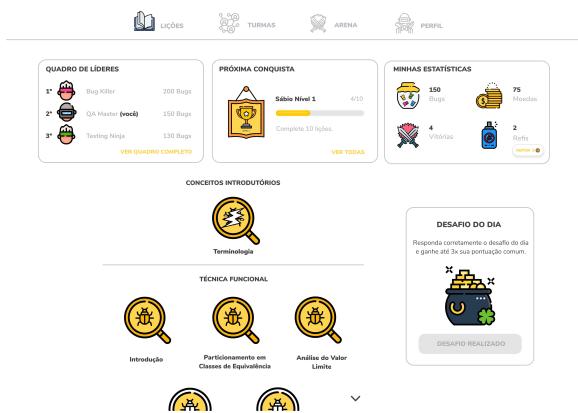


Figura 32: Tela principal - Desafio diário bloqueado.

Figura 35: Criar nova turma - Importar dados QA+.

Inscrito	Usuário	Perfil QA+	Perfil Minha Turma	Estatística	Turma QA+	Minha Turma	Total
Fulano dos Testes	Bug Killer	<a href="#">VER PERFIL</a>	<a href="#">VER PERFIL</a>	Bugs: 530 / 1000 Videos: 5 / 10 Conquistas: 5 / 8	530	520	850
Cidrano dos Testes	Testing Ninja	<a href="#">VER PERFIL</a>	<a href="#">VER PERFIL</a>	Bugs: 410 / 500 Videos: 5 / 5 Conquistas: 2 / 5	410	340	750
Beltrano dos Testes	Testing-Master	<a href="#">VER PERFIL</a>	<a href="#">VER PERFIL</a>	Bugs: 200 / 300 Videos: 3 / 2 Conquistas: 2 / 1	200	200	500

Figura 36: Turmas - Tela de gestão de turma.

Figura 39: Exemplo de duelo - Pergunta.

Figura 37: Turmas - Visão de turma acessada.

Figura 40: Exemplo de duelo - Seleção de resposta.

Figura 38: Tela geral da funcionalidade “Arena”.

Figura 41: Exemplo de duelo - Feedback resposta correta.

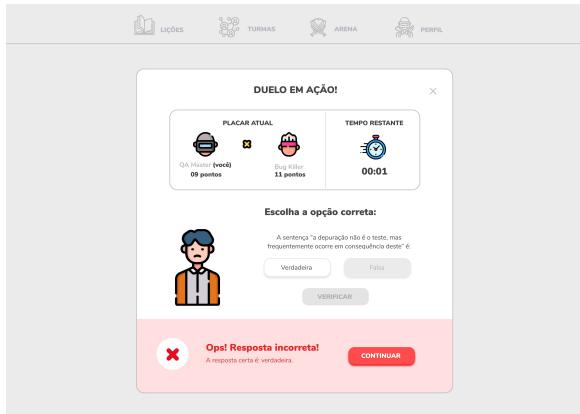


Figura 42: Exemplo de duelo - Feedback resposta incorreta.

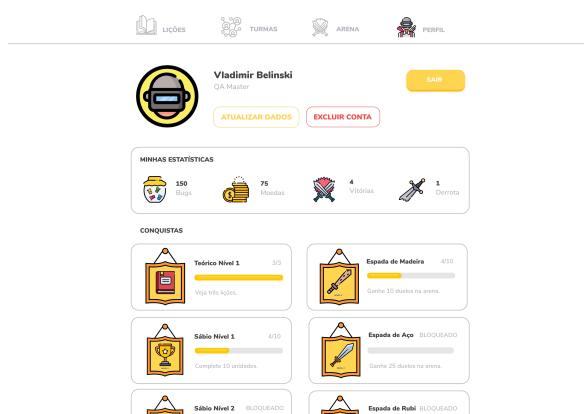


Figura 45: Tela geral da funcionalidade “Perfil”.

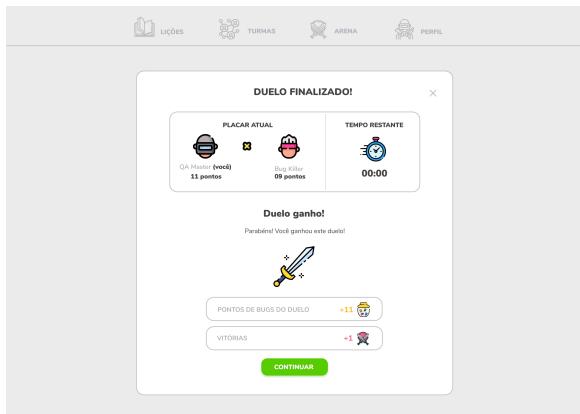


Figura 43: Exemplo de duelo - Tela de vitória.

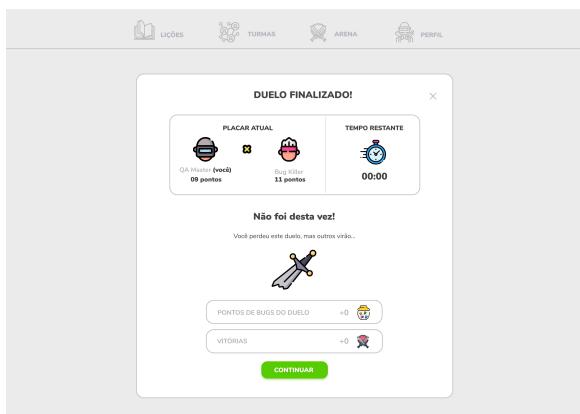


Figura 44: Exemplo de duelo - Tela de derrota.