

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

EDUARDO NEVES DE PINHO
EMANUELLE SCHIAVON
RODRIGO FONSECA BARRETO

CASE GAME: um jogo de estudo de caso

RIO DE JANEIRO

2018

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

EDUARDO NEVES DE PINHO
EMANUELLE SCHIAVON
RODRIGO FONSECA BARRETO

CASE GAME: um jogo de estudo de casos

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Geraldo Bonorino Xexéo

RIO DE JANEIRO

2018

P654c

Pinho, Eduardo Neves de

CASE GAME: um jogo de estudo de caso / Eduardo Neves de Pinho, Emanuelle Schiavon, Rodrigo Fonseca Barreto. – Rio de Janeiro, 2018.

46 f.

Orientador: Geraldo Bonorino Xexéo.

Trabalho de Conclusão de Curso (graduação)-Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2018.

1. Jogos sérios. 2. Estudo de caso. 3. Avaliações escolares. 4. Softwares jogáveis. I. Schiavon, Emanuelle. II. Barreto, Rodrigo Fonseca. III. Xexéo, Geraldo Bonorino (Orient.). IV. Universidade Federal do Rio de Janeiro, Instituto de Matemática. V. Título.

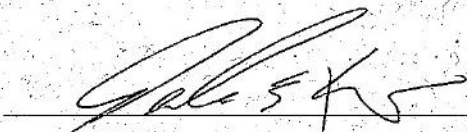
EDUARDO NEVES DE PINHO
EMANUELLE SCHIAVON
RODRIGO FONSECA BARRETO

CASE GAME: Um Jogo de Estudo de Casos

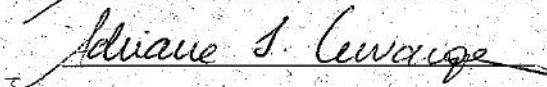
Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 13 de agosto de 2018.

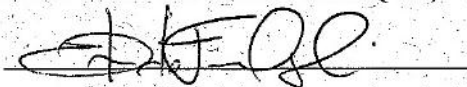
BANCA EXAMINADORA:



GERALDO BONORINO XEXÉO, D.Sc.



ADRIANA SANTAROSA VIVACQUA, D.Sc.



EDUARDO FREITAS MANGELI DE BRITO, M.Sc.

RIO DE JANEIRO – RJ

AGOSTO DE 2018

DEDICATÓRIA

Dedico esse trabalho a todos os amigos que tive oportunidade de conhecer e trocar conhecimento durante minha jornada no curso. A todos que acreditaram no meu vencimento, respeitaram as minhas dificuldades e me apoiaram por mais uma etapa da minha vida concluída.

- EDUARDO NEVES DE PINHO

Quero dedicar esse trabalho à memória de meu pai, apesar de me acompanhar por todo o processo não pode ver eu me formar. Agradeço a minha família, meu irmão, minha avó e minha cunhada, e a todos os amigos que de alguma forma me ajudaram durante estes anos. Gostaria de agradecer ao meu orientador e aos colegas que trabalharam comigo nesse projeto. Por fim, mais do que agradecer, eu quero dividir esse projeto com minha mãe, que esteve ao meu lado quando pensei em desistir e que me fez acreditar.

- EMANUELLE SCHIAVON

Agradeço a minha família que sempre me apoiou nessa jornada independente das dificuldades. Agradeço aos amigos que fiz na faculdade que me deram força para seguir em frente e não desistir. Agradeço também aos professores do curso que muito me ensinaram ao longo dos anos.

- RODRIGO FONSECA BARRETO

AGRADECIMENTOS

Agradecemos ao nosso professor e orientador Geraldo Bonorino Xexéo pela orientação, ideia do projeto e todo seu apoio. Também agradecemos ao Luis Fernando por nos mostrar possibilidades de abordagens no projeto e também ao Eduardo Mangeli pelas dicas de como solucionar certos problemas durante o projeto.

- EDUARDO NEVES DE PINHO

- EMANUELLE SCHIAVON

- RODRIGO FONSECA BARRETO

RESUMO

O objetivo deste trabalho é documentar todos os processos de desenvolvimento do projeto de um jogo para estudo de casos. A exposição do projeto se inicia com a explicação sobre jogos sérios, contexto em que o projeto está inserido, além de falar sobre estudo de caso. Posteriormente, mostra-se as ferramentas e tecnologias utilizadas no projeto, além de detalhar toda a especificação e os requisitos do projetos. Por último, é explicado a utilização da ferramenta, além de todas as dificuldades encontradas ao desenvolvê-la e possíveis melhorias para a mesma.

Palavras-chave: Jogos sérios. Estudo de caso. Avaliações escolares. Softwares jogáveis. Aplicações web

ABSTRACT

The purpose of this paper is to document the entire development process of this project: a game for case study. The project exhibition begins with the explanation of serious games, context that the project is located, as well as explain the case study. Subsequently, it shows the tools and technologies used in the project, in addition to detailing the entire specification and requirements of the projects. Finally, explains the use of the tool, the difficulties encountered in developing it and possible improvements to it.

Keywords: Serious games. Case studies. School tests. Playable softwares. Web applications.

LISTA DE ILUSTRAÇÕES

Figura 1: Benefícios do Ruby on Rails.....	15
Figura 2: Exemplo de repositório no Github.....	16
Figura 3: Quadro Kanban da ferramenta Waffle.....	17
Figura 4: Diagrama de Classes.....	19
Figura 5: Saudação Inicial.....	28
Figura 6: Tela de Login.....	28
Figura 7: Tela de Registro de Usuário.....	29
Figura 8: Tela de Gerenciamento de Usuário.....	30
Figura 9: Tela de Cadastro de Novo Usuário.....	30
Figura 10: Tela de Gerenciamento de Jogos.....	31
Figura 11: Tela de Inserção de Um Jogo.....	31
Figura 12: Tela de um Jogo e Gerenciamento de suas Cenas.....	32
Figura 13: Tela de Inserção de Cenas.....	32
Figura 14: Tela de uma Cena e Gerenciamento de suas Decisões.....	33
Figura 15: Tela de Inserção de uma Decisão.....	33
Figura 16: Tela de Gerenciamento de Decisões e Inserção de Respostas.....	34
Figura 17: Tela de Gerenciamento de Rodadas.....	35
Figura 18: Tela de Inserção de Rodadas.....	35
Figura 19: Tela Inicial de um Estudante.....	36
Figura 20: Tela de Criação de uma Partida.....	36
Figura 21: Tela Inicial de uma Partida Sendo Jogada.....	37
Figura 22: Tela de uma Partida Durante o Jogo.....	38
Figura 23: Tela Final de uma Partida.....	39
Figura 24: Diagrama de Classes da Melhoria de Critérios de Decisões e Variáveis Globais.....	43

LISTA DE TABELAS

Tabela 1: Funcionalidades dos Papéis de Usuários.....	20
---	----

SUMÁRIO

1 INTRODUÇÃO	10
2 MOTIVAÇÃO	11
2.1 O CASE GAME	11
2.2 O QUE É UM JOGO	11
2.3 JOGOS EDUCATIVOS	12
2.4 MÉTODO DE CASO	13
3 A TECNOLOGIA	14
3.1 RUBY ON RAILS	14
3.1.1 Gems	15
3.2 SQLITE E POSTGRESQL	15
3.3 HEROKU	16
3.4 GIT E GITHUB	16
3.5 WAFFLE	17
4 PROPOSTA DE AMBIENTE PARA O CASEGAME	18
4.1 MODELO DE DADOS	18
4.2 PAPÉIS DE USUÁRIOS DO SISTEMA	19
4.2.1 Administrador	19
4.2.2 Criador	20
4.2.3 Professor	20
4.2.4 Estudante	20
4.3 REQUISITOS DO SISTEMA	21
4.3.1 Requisitos funcionais	21
4.3.2 Requisitos não-funcionais	27
5 A FERRAMENTA CASE GAME	28
6 CONSIDERAÇÕES FINAIS	40
6.1 DIFICULDADES ENCONTRADAS	40
6.1.1 Linguagem de Programação Nova	40
6.1.2 Design Patterns	40
6.1.3 Conceito do passo-a-passo do jogo.	41
6.1.4 Conceito de respostas escritas arbitrariamente pelo usuário.	41
6.2 POSSÍVEIS TRABALHOS FUTUROS	41
6.2.1 Variáveis Globais e critérios de resposta.	42
6.2.2 Importação de Planilhas para regras do jogo	44

6.3 CONCLUSÃO	44
REFERÊNCIAS	46

1 INTRODUÇÃO

Este trabalho descreve como e o porquê foi desenvolvido um sistema que contempla o jogo de caso, metodologia de ensino criada e implantada em Harvard e que ganha espaço no ramo do ensino.

A área da educação tem passado por grandes transformações, diferentes metodologias de aprendizado têm sido testadas e adotadas por profissionais da esfera, a fim de aumentar o aproveitamento dos alunos.

Há algum tempo, o uso de jogos para ensino tem ganhado destaque entre educadores. Essa metodologia ainda tem muito a crescer, mas já é possível apontar alguns casos de sucesso.

O jogo foi desenvolvido em uma tecnologia totalmente nova para os desenvolvedores do projeto, o Ruby on Rails. Apesar do desafio de aprender uma nova linguagem e nova tecnologia, o framework ajudou bastante no desenvolvimento, em função das suas convenções. Muitas ferramentas foram usadas para auxiliar durante o desenvolvimento, como: GitHub, Waffle, Heroku e outras.

Para o entendimento do sistema, foi necessário fazer uma detalhada análise do sistema com Diagrama de Classes, Requisitos Funcionais e Não Funcionais e Descrição dos Atores do Sistema.

O CaseGame consiste em um jogo de perguntas e respostas, onde os próximos passos seguidos dependem da resposta dada no passo anterior. As perguntas são baseadas em situações reais comumente vivida em uma corporação, e os jogadores simulam ser um líder com decisões a tomar.

2 MOTIVAÇÃO

Nesse capítulo iremos dizer qual é a motivação do projeto. Da onde surgiu a ideia tanto do projeto, quanto dos conceitos utilizados nele.

2.1 O CASE GAME

CaseGame é um jogo educativo que tem como objetivo ajudar e simplificar o método de aprendizado e avaliação em uma sala de aula e torna mais amigável a interação entre professores e alunos nesse processo. Utiliza dos conceitos de Método de Caso e Jogos Educativos em seu funcionamento. Com o sistema, o aluno participa do processo de aprendizado de forma mais divertida e o professor consegue reunir facilmente resultados da participação da sua turma com ajuda da tecnologia. Não há nível de graduação para ser utilizado, podendo ser aplicado em turmas de ensino básico e fundamental, até mesmo em turma de graduação e pós-graduação.

Explicaremos os conceitos abordados na construção da aplicação a seguir.

2.2 O QUE É UM JOGO

Definir exatamente o que é um jogo é algo extremamente difícil. Muitos estudiosos têm suas próprias definições de jogo, e algumas delas se contradizem, mostrando que o que é um jogo para algumas pessoas, pode não ser para outras.

Existem diversos tipos de jogos, entre eles podemos citar: jogos de tabuleiro, jogos eletrônicos, jogos de dados, jogos de cartas, jogos de mesa.

A palavra jogo vem do termo em latim “*jocus*” que significa brincadeira ou divertimento. Um jogo é toda atividade recreativa e estruturada que envolve pelo menos um jogador e que tenha regras. Além da parte do divertimento, atualmente é muito comum encontrarmos jogos com fins educacionais, trazendo aprendizado para os alunos usuários de tais jogos. A nossa ferramenta está ligada a um jogo com esse propósito educacional.

Em seu livro “*Homo Ludens*”, Huizinga define jogo como: "uma atividade voluntária exercida dentro de certos e determinados limites de tempo e espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo,

acompanhado de um sentimento de tensão e alegria e de uma consciência de ser diferente de vida cotidiana" (HUIZINGA, 1938).

2.3 JOGOS EDUCATIVOS

Jogos educativos são uma ótima ferramenta de aprendizagem. Eles ajudam na motivação do aluno, contribuem como estratégias alternativas para o desenvolvimento de conteúdo, na aquisição de certas habilidades, pois muitas pessoas possuem mais facilidade para aprender e memorizar através desse método de ensino. Pelo fato de ser uma maneira de ensino lúdica, ela não é cansativa e ajuda a despertar interesse dos alunos.

Segundo Vygotsky, o lúdico influencia enormemente o desenvolvimento da criança. “É através do jogo que a criança aprende a agir, sua curiosidade é estimulada, adquire iniciativa e autoconfiança, proporciona o desenvolvimento da linguagem, do pensamento e da concentração.” (Vygotsky, 1989)

Para Grando(2001) , inserir jogos no meio educacional implica em vantagens tais como:

- fixação de conceitos já aprendidos de uma forma motivadora para o aluno
- introdução e desenvolvimento de conceitos de difícil compreensão
- desenvolvimento de estratégias de resolução de problemas
- aprender a tomar decisões e saber avaliá-las
- propicia o relacionamento de diferentes disciplinas
- o jogo favorece a socialização entre alunos e a conscientização do trabalho em equipe
- as atividades com jogos podem ser utilizadas para reforçar ou recuperar habilidades de que os alunos necessitem. Útil no trabalho com alunos de diferentes níveis
- as atividades com jogos permitem ao professor identificar, diagnosticar alguns erros de aprendizagem, as atitudes e as dificuldades dos alunos

Mas para Grando também implica em desvantagens como:

- quando os jogos são mal utilizados, existe o perigo de dar ao jogo um caráter puramente aleatório, tornando-se um “apêndice” em sala de aula. Os alunos jogam e se sentem motivados apenas pelo jogo, sem saber porque jogam

- a perda de “ludicidade” do jogo pela interferência constante do professor, destruindo a essência do jogo
- o tempo gasto com as atividades de jogo em sala de aula é maior e, se o professor não estiver preparado, pode existir um sacrifício de outros conteúdos pela falta de tempo
- a coerção do professor, exigindo que o aluno jogue, mesmo que ele não queira, destruindo a voluntariedade pertencente a natureza do jogo;

Muito dessas vantagens e desvantagens citadas dependem das regras e objetivos do jogo, e também de como ele é ministrado pelo professor em questão.

A nossa ferramenta tem pouca interferência do professor, ele não é necessariamente para ser utilizado em sala de aula, podendo ser jogado pelo estudante em sua própria casa, com isso não gasta o tempo em sala de aula.

Uma das grandes vantagens da ferramenta é de poder criar um jogo específico para cada tipo de aluno, com isso o professor pode manter um acompanhamento de como os seus alunos estão em diversos quesitos e com isso ajudar no reforço dos estudos.

2.4 MÉTODO DE CASO

Idealizado na Universidade de Harvard, o método de caso é uma metodologia de ensino que avalia a capacidade de tomada de decisão dos participantes. Os alunos são colocados no papel de gestor diante de situações reais, sendo obrigados a aliar planejamento e espontaneidade.

Este estilo de aula faz um contraponto com aulas expositivas permitindo que o aluno pense, argumente e convença seus colegas, colocando em prática assuntos que em uma apresentação podem parecer abstratos.

Nesta aula o professor deve mediar os alunos, apontar problemas das decisões tomadas e avaliar as decisões tomadas.

3 A TECNOLOGIA

Nesse capítulo falaremos de todas as tecnologias que usamos no decorrer do projeto. Algumas características deles e o motivo da seleção.

3.1 RUBY ON RAILS

A linguagem de programação usada neste projeto foi o Ruby, uma linguagem interpretada e uma das únicas linguagens populares que nasceu fora dos EUA e Europa. Ela foi criada em 1993 por Yukihiro Matsumoto, no Japão e é bastante conhecida por ter uma performance boa. O Ruby diferente de outras linguagens Orientada a Objetos possui uma sintaxe bem simples e muitos métodos que ajudam o desenvolvedor.

A principal motivação para essa decisão foi o uso do framework Rails, uma ferramenta baseada no modelo MVC que facilita a geração de código e configuração do projeto e foi responsável pelo crescimento da popularidade do Ruby.

Estas facilidades são consequência dos conceitos da ferramenta “Convention Over Configuration”. Este lema prega a diminuição de decisões do desenvolvedor, padronizando a estrutura da aplicação. Por exemplo, o nome de uma tabela da base de dados será, por padrão, o plural do nome dado ao modelo, a estrutura de pastas também segue um esquema, por fim, o relacionamento entre o modelo das aplicações também é previamente definido pelo Rails.

Outro conceito importante usado em muitas linguagens de programação é o DRY acrônimo para “Don’t Repeat Yourself”, em português “Não se Repita”. “Partials” seguem esta norma, elas consistem em blocos de Html que são usados em mais de um lugar.

Segundo a empresa Redmonk, Ruby foi a oitava linguagem mais usada em 2017. A comunidade de desenvolvedores Rails tem grande influência sobre o crescimento da linguagem, ela é conhecida por membros que estão sempre dispostos a ajudar iniciantes, eles também contribuem desenvolvendo um grande número de bibliotecas, em Rails chamadas de Gems. Outro fator para a popularização foi a adoção da linguagem por várias grandes empresas como: Twitter, Airbnb, GitHub, Groupon, entre outras.

Arquivos importantes pra se destacar são: o “GemFile”, onde reúnem-se todas as Gems e o Rails é responsável por adicioná-las na aplicação; o “config/database.yml” onde é adicionado o driver do banco de dados; no “config/routes.rb” localiza-se todas as páginas e métodos acessados pela aplicação. Na pasta “app” ficam os códigos da aplicação.

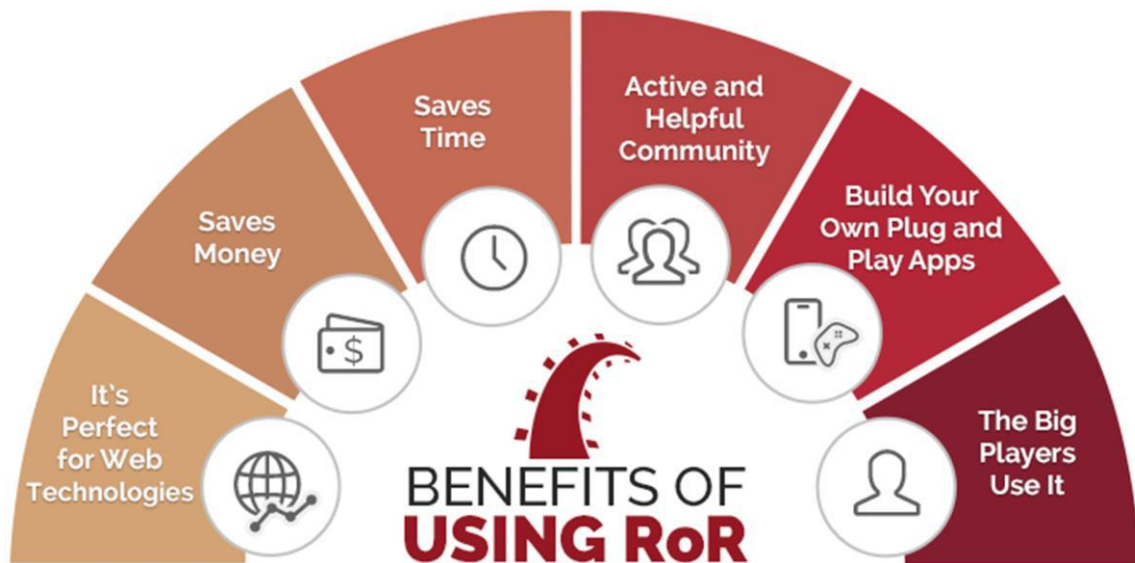


Figura 1: Benefícios do Ruby on Rails

3.1.1 Gems

Bibliotecas são códigos feitos por outras pessoas para problemas comuns, em Ruby as bibliotecas são chamadas de Gems. O próprio Rails é considerado uma Gem, assim como o Banco de Dados.

Estas foram as gems escolhidas e suas finalidades: Para realizar o gerenciamento de papéis de usuários a gem ‘Pundit’, a ‘Bcrypt’ para encriptação de senha, o ‘Bootstrap’ foi escolhido para ajudar no layout da aplicação, o ‘Paperclip’, para upload de imagens. Para ajudar no desenvolvimento, ‘byebug’ e ‘web-console’, ajudam no debug do projeto.

3.2 SQLITE E POSTGRESQL

Dois gerenciadores de banco de dados foram usados durante a execução do projeto. O SQLite é o padrão do Ruby on Rails, na criação do projeto, seguindo esta convenção do framework esta ferramenta foi mantida. Após a necessidade da montagem de um ambiente de produção e a incompatibilidade com o ambiente usado no Heroku, o sistema usado para gerenciar a base foi mudado para o PostGres, que aceita um grande número de acessos, ao contrário do anterior.

3.3 HEROKU

Para que o cliente conseguisse acompanhar o desenvolvimento do projeto, um ambiente de produção foi montado com a plataforma Heroku. Ela faz todo o serviço de empacotamento, construção e inicia a aplicação em um ambiente público.

O Heroku tem extensa documentação para o Ruby on Rails, facilitando assim o deploy da aplicação.

3.4 GIT E GITHUB

Conhecendo a importância de versionar um projeto, o Git foi eleito para tal função, por ser a ferramenta mais usada do mercado e pela grande experiência dos integrantes do time. O GitHub é um plataforma de hospedagem de código integrado com o Git.

A ferramenta se tornou muito popular entre desenvolvedores open source em razão da funcionalidade “pull request”. Nela, os usuários podem enviar seu código para que o administrador do projeto possa analisar, comentar e então aceitar.

Com o Git foi possível acompanhar a evolução da aplicação, as contribuições por parte do time, além de ter garantido quatro computadores com o projeto, diminuindo a chance de perda do projeto por eventos inesperados.

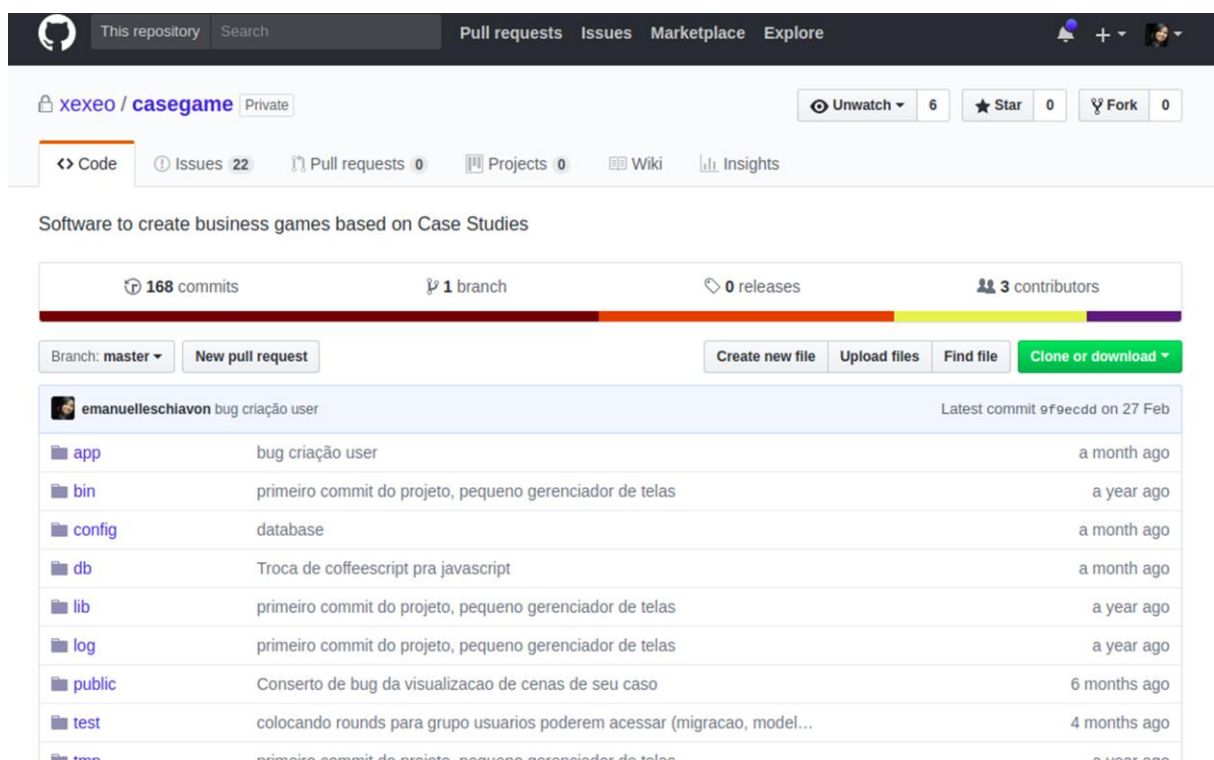


Figura 2: Exemplo de repositório no Github

3.5 WAFFLE

Na tentativa de aumentar a produtividade da equipe, um quadro Kanban foi adotado para o gerenciamento do projeto. O sistema escolhido para aplicar esta metodologia foi o Waffle.

Assim como outros da categoria, O Waffle tem um quadro configurável onde é possível aplicar métodos ágeis para garantir uma boa comunicação de um time. Ele foi escolhido pois tem integração para reportar erros diretamente do GitHub.

O quadro foi composto pelas seguintes raias: Backlog, Planejado, Pronto, Em Progresso, A Ser Testado, Testando e Pronto.

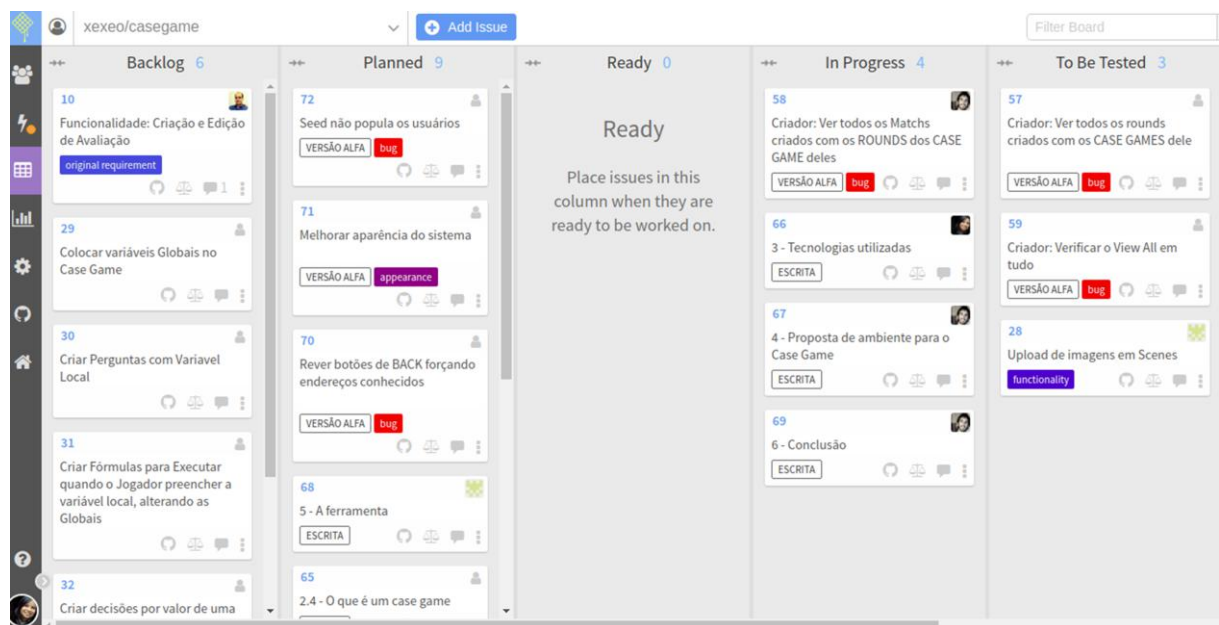


Figura 3: Quadro Kanban da ferramenta Waffle

4 PROPOSTA DE AMBIENTE PARA O CASEGAME

Nesse capítulo tentaremos explicar formalmente e tecnicamente como foi a proposta do ambiente do projeto. As classes e diagramas, suas funcionalidades para depois mostrar o funcionamento dele.

4.1 MODELO DE DADOS

A partir do diagrama a seguir (figura 4), podemos entender um pouco mais sobre o funcionamento do projeto.

As tabelas “Users”, “Assignments” e “Roles” fazem parte do contexto de gerenciamento de usuários. Um usuário cadastrado no sistema pode ter n papéis. Cada papel dá um determinado acesso ao usuário. Falaremos mais das permissões de cada papel de usuário mais adiante.

As tabelas “Case_Games”, “Scenes”, “Decisions” e “Answer” já estão no contexto de gerenciamento dos jogos do sistema. Cada instância de um Case_Game terá várias cenas (Scenes) que serão as etapas onde o jogo se encontra ao ser jogado. Cada uma dessas cenas, têm uma “Decision”, que são as perguntas das etapas, que contém várias resposta (“Answers”). A resposta escolhida em uma decisão leva a uma próxima cena. Assim que a engenharia do jogo funciona. Se uma cena de um jogo não tiver nenhuma decisão, significa que é uma cena final.

As tabelas do diagrama “Rounds”, “Matches”, “Teams” e também “Users” fazem parte do gerenciamento de Sessões, que chamamos no diagrama de “Rounds”. Uma instância de sessão (round) é justamente a abertura de um jogo (Case_Game) por um professor, para uma turma. Turmas são representadas pela tabela “Teams” que é um conjunto dos alunos que estão permitidos a jogar/executar cada sessão de um jogo. Para executar o jogo é necessário criar uma, ou várias, instâncias de um Partida (“Match”) que pertence a sessão.

Por último, a tabela Score que representa as etapas do processo da partida sendo jogada pelo aluno. Armazena todas as informações e histórico de uma partida.

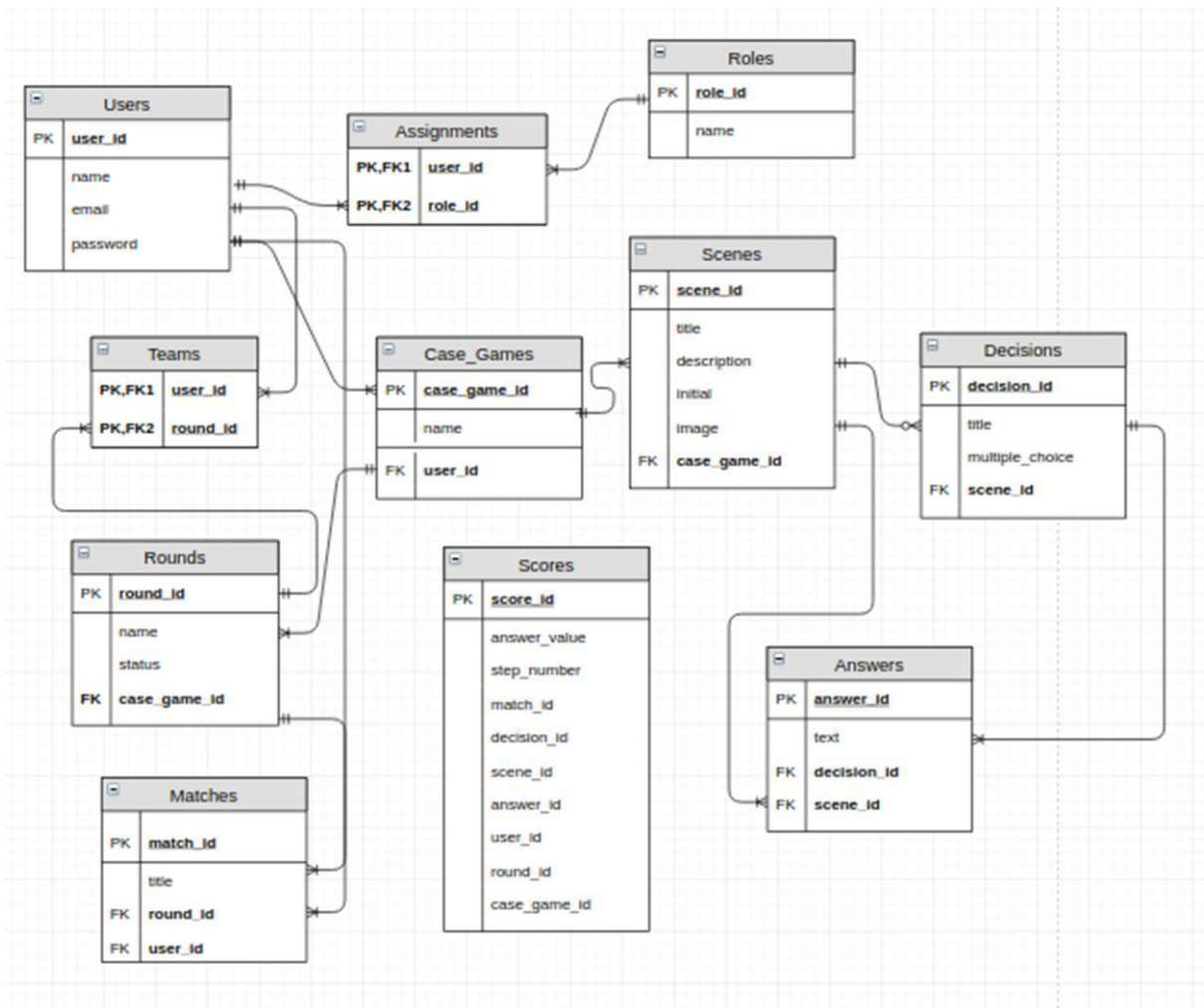


Figura 4: Diagrama de Classes

4.2 PAPÉIS DE USUÁRIOS DO SISTEMA

Nesse capítulo iremos mostrar os papéis que um usuário cadastrado no sistema pode assumir, passando uma descrição e a tabela com relação as suas funcionalidades. Temos também uma explicação mais sucinta e ilustrativa na tabela 1 mais abaixo.

4.2.1 Administrador

O usuário com papel de Administrador tem como principal função gerenciar os usuários do sistema. Tem o poder de criar novos usuários, editar e remover os existentes. Pode também atribuir qualquer papel para os usuários já existentes.

4.2.2 Criador

O usuário com papel de Criador tem como principal função criar os jogos (CaseGames) e as instancias possuidas por um jogo, como as telas, as decisões e as respostas para as decisões. Esse usuário só gerencia as instancias dos jogos, que posteriormente serão utilizados pelos usuários de papéis Professor e Estudantes.

4.2.3 Professor

O usuário com papel de Professor tem como principal função abrir novas sessões de jogos para sua turma de alunos. Pode também, além de gerenciar essas sessões, ver as respostas das partidas criadas por seus alunos.

4.2.4 Estudante

O usuário com papel de Estudante tem como principal função criar partidas dos jogos permitidos por seus professores.

Tabela 1: Funcionalidades dos Papéis de Usuários

Funcionalidade \ Papéis de Usuários	Administrador	Criador	Professor	Estudante
Case Games		Cria, Visualiza todos. Edita e exclui os dele.	Visualiza todos	
Cenas		Visualiza, cria, edita e exclui as Cenas do Casegame dele.		Visualiza apenas durante o jogo.
Sessões		Visualiza as Sessões do Casegame dele.	Cria, visualiza todas. Edita e exclui as dele.	Visualiza as Sessões que tem acesso.
Partidas		Visualiza as Partidas do Casegame dele.	Visualiza as Partidas das Sessões dele.	Cria, Visualiza e joga as Partidas dele.
Usuários	Cria, visualiza, edita e exclui todos.			

4.3 REQUISITOS DO SISTEMA

Nesse subcapítulo iremos listar todos os requisitos funcionais e não funcionais do sistema.

4.3.1 Requisitos funcionais

1. Sistema deve mostrar área de login para acesso do usuário.

Descrição: Sistema deve mostrar área de login com campo e-mail e senha.

Justificativa: Todo usuário necessita estar cadastrado e logado para ter acesso ao sistema.

Fonte: Qualquer usuário.

Critério de aceitação: Se o usuário estiver cadastrado e preencher os dois campos corretamente, ele consegue acesso ao sistema, se não, o sistema mostra as mensagens de erro.

Satisfação do usuário: 1/5

Insatisfação do usuário: 5/5

2. Sistema deve mostrar área para cadastro de novo usuário.

Descrição: Sistema deve mostrar área de de novo usuário com os campos de nome, email, senha e confirmação de senha. Além disso, a possibilidade de voltar para a tela anterior.

Justificativa: Todo usuário necessita estar cadastrado e logado para ter acesso ao sistema.

Fonte: Qualquer usuário.

Critério de aceitação: Se o usuário preencher os dados dos quatro campos corretamente, o usuário é cadastrado com papel de Estudante, e posteriormente acessa a área principal do sistema. Se não, o sistema mostra mensagem de erro.

Satisfação do usuário: 1/5

Insatisfação do usuário: 5/5

3. Usuário logado com papel de Administrador deve poder gerenciar usuários do sistema.

Descrição: Ao se logar no sistema com o Papel de Administrador, o usuário visualizará uma tela com a lista de usuários e deve adicionar novos usuários no sistema,

indicando o nome, o email, a senha, a confirmação de senha e o papel desse novo usuário a ser adicionado. Além disso, o usuário logado também pode editar os dados de outros usuários já cadastrados, assim como removê-los do sistema.

Justificativa: Um usuário administrador é o único que tem o poder de adicionar outros usuários, indicando inclusive o papel deles.

Fonte: Administrador

Critério de aceitação: Para adicionar ou editar um novo usuário, se o usuário logado com papel de administrador preencher os campos corretamente, um novo usuário deve ser inserido no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Para remover um usuário, o sistema perguntará se o usuário tem a certeza da remoção. Confirmando, recebe uma mensagem de sucesso. Cancelando, volta para a tela anterior.

Satisfação do usuário: 2/5

Insatisfação do usuário: 3/5

4. Usuário logado com papel de Criador deve poder gerenciar Casegames.

Descrição: Ao logar no sistema com papel de Criador, o usuário visualizará uma tela com a lista de instâncias de Casegames e deve poder adicionar novos casegames no sistema, indicando o nome. Além disso, o usuário também pode editar os dados dos casegames assim como removê-los do sistema.

Justificativa: O usuário com papel de criador é o responsável por criar e abastecer dados do jogos que serão abertos por Professores e jogados por alunos.

Fonte: Criador

Critério de aceitação: Para adicionar ou editar um novo casegame, se o usuário logado preencher os campos corretamente, um novo casegame deve ser inserido no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Para remover um casegame, o sistema perguntará se o usuário tem a certeza da remoção. Confirmando, recebe uma mensagem de sucesso. Cancelando, volta para a tela anterior.

Satisfação do usuário: 4/5

Insatisfação do usuário: 5/5

5. Usuário logado com papel de Criador deve poder gerenciar cenas dos Casegames.

Descrição: Ao entrar em um determinado casegame, o usuário logado visualizará uma área da tela com a lista de cenas do casegame selecionado e deve poder adicionar novas cenas no sistema, indicando o nome, a descrição, uma possível imagem e se a cena é inicial. Além disso, o usuário também pode editar os dados das cenas criadas, assim como removê-las do sistema.

Justificativa: As cenas são as telas que constroem um casegame. O usuário com papel de criador é o responsável por criar e abastecer os dados dos jogos que serão abertos por professores e jogados por aluno.

Fonte: Criador

Critério de aceitação: Para adicionar ou editar uma nova cena, se o usuário logado preencher os campos corretamente, uma nova cena deve ser inserida no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Para remover uma cena, o sistema perguntará se o usuário tem a certeza da remoção. Confirmando, recebe uma mensagem de sucesso. Cancelando, volta para a tela anterior.

Satisfação do usuário: 4/5

Insatisfação do usuário: 5/5

6. Usuário logado com papel de Criador deve poder gerenciar decisões das cenas.

Descrição: Ao entrar em uma determinada cena, o usuário logado visualizará uma área da tela com a lista de decisões da cena selecionada e deve poder adicionar novas decisões no sistema, indicando o título e o tipo das respostas (múltipla escolha ou resposta curta). Além disso, o usuário também pode editar os dados das decisões, assim como removê-las do sistema.

Justificativa: As decisões contêm respostas que fazem a ligação entre cenas diferentes de um jogo. O usuário com papel de criador é o responsável por criar e abastecer os dados dos jogos que serão abertos por professores e jogados por aluno.

Fonte: Criador

Critério de aceitação: Para adicionar ou editar uma nova decisão, se o usuário logado preencher os campos corretamente, uma nova decisão deve ser inserida no sistema e uma

mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Para remover uma decisão, o sistema perguntará se o usuário tem a certeza da remoção. Confirmando, recebe uma mensagem de sucesso. Cancelando, volta para a tela anterior.

Satisfação do usuário: 4/5

Insatisfação do usuário: 5/5

7. Usuário logado com papel de Criador deve poder gerenciar possíveis respostas das decisões.

Descrição: Ao entrar em uma determinada decisão, o usuário logado visualizará uma área com a lista de respostas da decisão selecionada e deve poder adicionar novas respostas no sistema, indicando o título e a cena que será mostrada caso a resposta seja selecionada ou escrita. Além disso, o usuário também pode editar os dados das respostas, assim como removê-las do sistema.

Justificativa: As respostas fazem a ligação entre cenas diferentes de um jogo. O usuário com papel de criador é o responsável por criar e abastecer os dados dos jogos que serão abertos por professores e jogados por aluno.

Fonte: Criador

Critério de aceitação: Para adicionar ou editar uma nova resposta, se o usuário logado preencher os campos corretamente, uma nova resposta deve ser inserida no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Para remover uma resposta, o sistema perguntará se o usuário tem a certeza da remoção. Confirmando, recebe uma mensagem de sucesso. Cancelando, volta para a tela anterior.

Satisfação do usuário:4/5

Insatisfação do usuário: 5/5

8. Usuário logado com papel de Professor deve poder gerenciar sessões de jogos (Rounds).

Descrição: Ao se logar no sistema com o Papel de Professor, o usuário visualizará uma tela com a lista de sessões de jogos e deve adicionar novas sessões no sistema, indicando

o nome da sessão, o casegame a qual a sessão pertence e os usuários com papel de aluno que poderão criar jogos dessa sessão. Além disso, o usuário logado também pode editar os dados de outras sessões já cadastradas, assim como bloqueá-las do sistema, caso estejam desbloqueadas e vice-versa.

Justificativa: Os professores são responsáveis por habilitar os casegames criados para sua turmas. Daí a necessidade das sessões de jogos.

Fonte: Professor

Critério de aceitação: Para adicionar ou editar uma nova sessão de jogo, se o usuário logado preencher os campos corretamente, uma nova sessão deve ser inserida no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada. É necessário que no campo de seleção de alunos, exista a possibilidade de selecionar múltiplos alunos, além de buscá-los pelo nome.

Satisfação do usuário: 4/5

Insatisfação do usuário: 3/5

9. Usuário logado com papel de Professor deve poder visualizar instâncias de jogos.

Descrição: Ao entrar na tela de visualização de um sessão de um jogo, além das informações pertinentes à sessão, que podem ser editadas, o usuário logado também poderá visualizar informações dos jogos criados por Alunos como o nome do jogos, o nome do aluno que criou, a data de criação e de atualização e todas as respostas do Aluno dentro desse jogo.

Justificativa: O Professor precisa visualizar as respostas dos alunos de sua turma em cada jogo para poder fazer a avaliação.

Fonte: Professor

Critério de aceitação: Clicar em uma sessão criada e visualizar as informações de todos os jogos da sessão: Nome do jogo, nome do aluno, data de criação, data da última resposta e lista de respostas.

Satisfação do usuário: 5/5

Insatisfação do usuário: 4/5

10. Usuário logado com papel de Estudante deve poder criar instâncias de jogos (Matches).

Descrição: Ao se logar no sistema com o Papel de Estudante, o usuário visualizará uma tela com a lista de jogos e deve adicionar novos jogos no sistema, indicando o nome do jogo e a sessão a qual o jogo pertence.

Justificativa: O Estudante é o responsável pela etapa final do sistema, que é a de executar a resposta de cada jogo criado e posteriormente ser avaliado por um professor.

Fonte: Estudante

Critério de aceitação: Para adicionar um novo jogo, se o usuário logado preencher os campos corretamente, um novo jogo deve ser inserido no sistema e uma mensagem de sucesso deve ser exibida na tela. Se não, uma mensagem de erro deve ser mostrada.

Satisfação do usuário: 4/5

Insatisfação do usuário: 4/5

11. Usuário logado com papel de Estudante deve poder jogar as instâncias de jogos.

Descrição: Ao se logar no sistema com o Papel de Estudante, o usuário visualizará uma tela com a lista de jogos (matches) criados por ele. Para cada um desses jogos, o usuário poderá jogar o jogo. O jogo escolhido para ser jogado apresentará uma nova tela no sistema que representará a lógica do casegame, com suas telas, decisões e respostas a serem respondidas pelo aluno que está jogando.

Justificativa: O Estudante é o responsável pela etapa final do sistema, que é a de executar a resposta de cada jogo criado e posteriormente ser avaliado por um professor.

Fonte: Estudante

Critério de aceitação: Ao entrar na tela de jogar o jogo, o sistema exibirá a partida (match) que está sendo jogado e suas informações. Além disso, a tela e decisão em que a partida se encontra e a resposta para essa decisão, que pode ser representada por um campo de texto ou respostas em múltipla escolha. Ao escolher ou digitar a resposta, o sistema exibirá uma próxima cena, com sua decisão de acordo com a última resposta. Se a cena não tiver nenhuma decisão, considera-se que o jogo foi completado. Além disso, o Estudante pode visualizar nessa mesma tela, todas as respostas já inseridas por ele nessa partida.

Satisfação do usuário: 5/5

Insatisfação do usuário: 5/5

4.3.2 Requisitos não-funcionais

1. Layout deve ser intuitivo e rodar em diferentes plataformas

Descrição: Todas as telas do sistemas devem ser representadas nas diferentes plataformas: PCs, Notebooks, Tabletes e Smartphones.

Justificativa: Necessita-se que o usuário, principalmente com o papel de aluno possa utilizar o sistema em qualquer momento. Portanto, há a possibilidade do usuário utilizar-se do sistema em diferentes plataformas.

Fonte: Qualquer usuário

Critério de aceitação: Todas as informações de todas as telas devem aparecer em todas as plataformas, Nenhuma informação deve ser perdida.

Satisfação do usuário: 2/5

Insatisfação do usuário: 3/5

5 A FERRAMENTA CASE GAME

A tela inicial, caso não tenha um usuário já logado no sistema, faz uma breve definição da ferramenta e nos mostra dois botões para ir para o login .

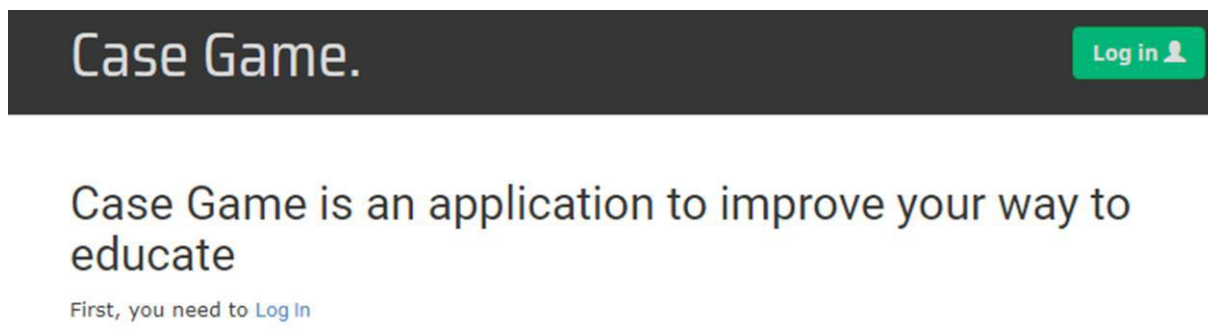


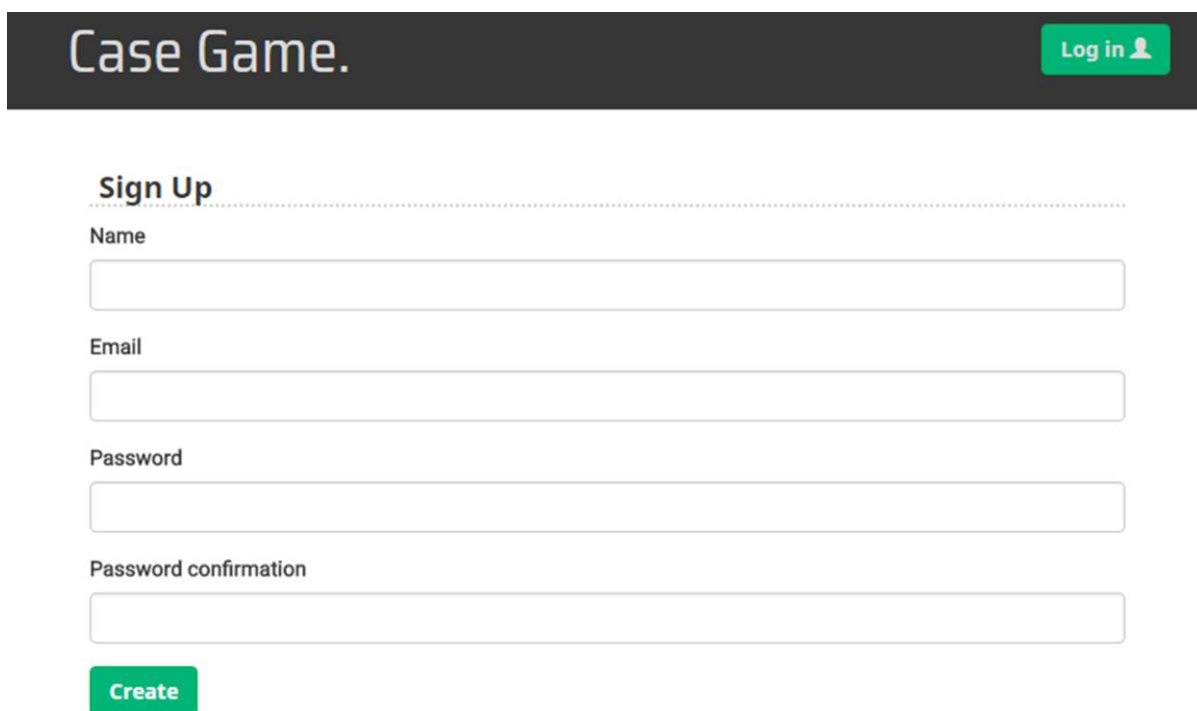
Figura 5: Saudação Inicial

Na tela de login podemos abrir a sessão para utilizar a ferramenta caso já tenha um usuário criado apenas escrevendo o e-mail e a senha em questão, porém caso seja novo um botão de cadastro está disponível.



Figura 6: Tela de Login

Para criar um novo usuário é necessário fornecer: Nome, E-mail e senha, além de confirmar a senha por motivos de segurança. Todos os campos citados devem ser preenchidos. O e-mail deve conter o caracter “@” para identificar que é um e-mail, a senha deve ter no mínimo 6 caracteres e a confirmação da senha deve ser exatamente igual a mesma para poder criar o usuário.



Case Game. [Log in](#)

Sign Up

Name

Email

Password

Password confirmation

[Create](#)

Figura 7: Tela de Registro de Usuário

Em nossa ferramenta temos quatro papéis que podem ser atribuídos aos usuários. Cada usuário recebe pelo menos um e no máximo todos os 4. São eles: *administrador*, *criador*, *professor* e *estudante*.



Caso o usuário seja criado pelo botão de “*Sign Up*”, ele automaticamente recebe o papel de estudante. Para os demais papéis é necessário que o novo usuário seja criado ou editado por um usuário que já é administrador.



Agora uma demonstração do que cada papel de usuário pode fazer.

Administrador
















Ele é o responsável por editar, criar e excluir usuários do sistema.

A tela inicial do administrador mostra no máximo cinco dos usuários já criados exibindo o nome, e-mail e os papéis que cada um recebeu. O total de usuários está representado no número ao lado de Users. Para visualizar a lista completa basta clicar no botão “*View All*”. Das ações que ele pode exercer estão a visualização do usuário (olho), a edição de usuário (lápis), a exclusão de usuário (X) e a criação de novo usuário (botão de criar).

Case Game.  

Welcome, Administrador  [Log out](#) 

Users



Name	Email	Roles	Actions
Administrador	administrador@email.com	Administrator Administrator	  
Professor	professor@email.com	Professor Professor	  
Criador	criador@email.com	Creator Creator	  
Estudante	estudante@email.com	Student Student	  
Deus	deus@email.com	Administrator Administrator Professor Professor Creator Creator Student Student	  



[+ Create new User](#)
[View All](#)

Figura 8: Tela de Gerenciamento de Usuário

Um administrador criando um usuário escolhe quantos papéis ele receberá, podendo marcar de um até quatro deles.

Todos os demais campos recebem o mesmo tratamento e verificação do que a criação através do botão “*Sign Up*”.

Case Game.  

Welcome, Administrador  [Log out](#) 

New User

Name

Email

Role

☐ Administrator
 ☐ Professor
 ☐ Creator
 ☐ Student

Password

Password confirmation

[Back](#)
[Save](#)

Figura 9: Tela de Cadastro de Novo Usuário

Criador

O criador cria os Case Game, as cenas de cada Case Game e as decisões das cenas. Ele pode visualizar os Case Games já criados, editar e excluir os cases os quais ele é o criador.

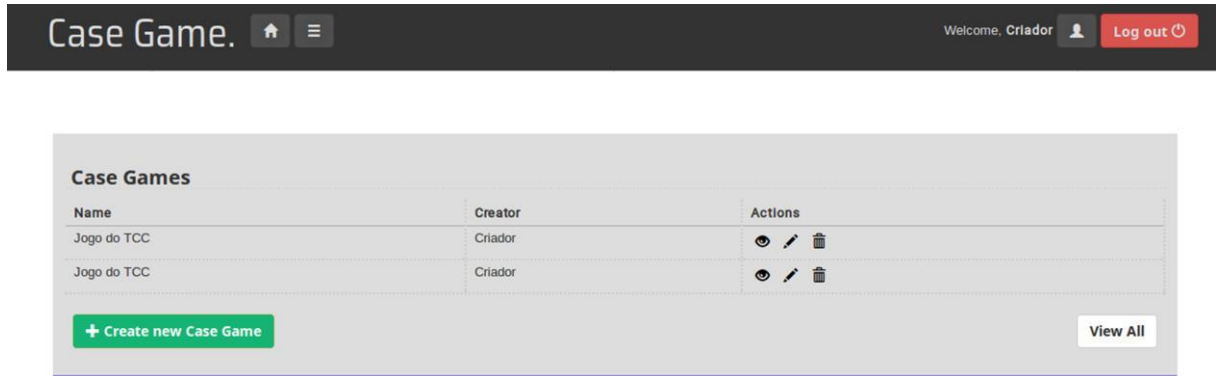


Figura 10: Tela de Gerenciamento de Jogos

Escolhendo o nome do novo case game a ser criado.

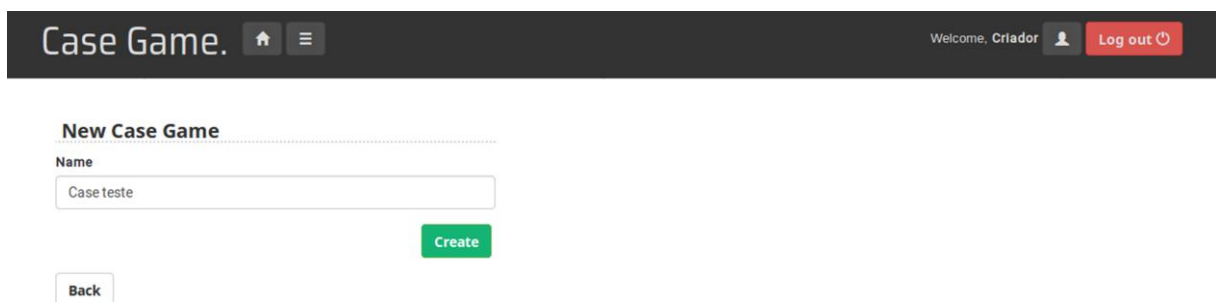


Figura 11: Tela de Inserção de Um Jogo

Após criar o Case Game, o criador pode começar a adicionar as cenas

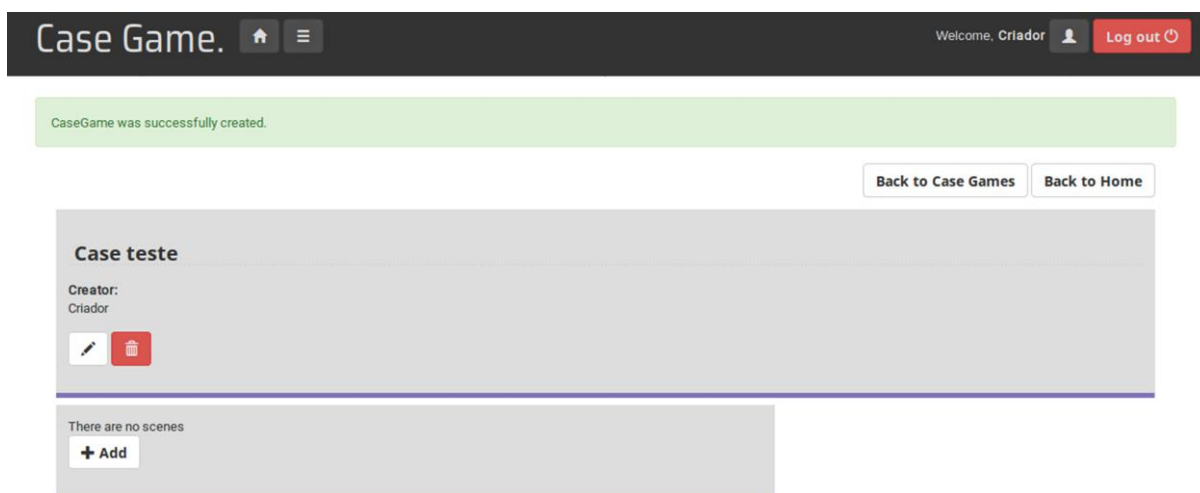


Figura 12: Tela de um Jogo e Gerenciamento de suas Cenas

Para criar a cena é necessário preencher o nome dela e sua descrição. Além disso deve-se dizer se ela será a cena inicial do case game, se sim é necessário marcar o *checkbox initial*. Cada Case Game só pode ter uma tela inicial, que é a primeira cena do fluxo de cenas. Portanto, caso já exista uma cena inicial nesse mesmo case game, uma mensagem de erro será exibida para notificar o criador.

Por último ainda é possível adicionar uma imagem (jpeg ou png) para a cena em questão, ela será exibida para os jogadores quando estiverem nessa cena.

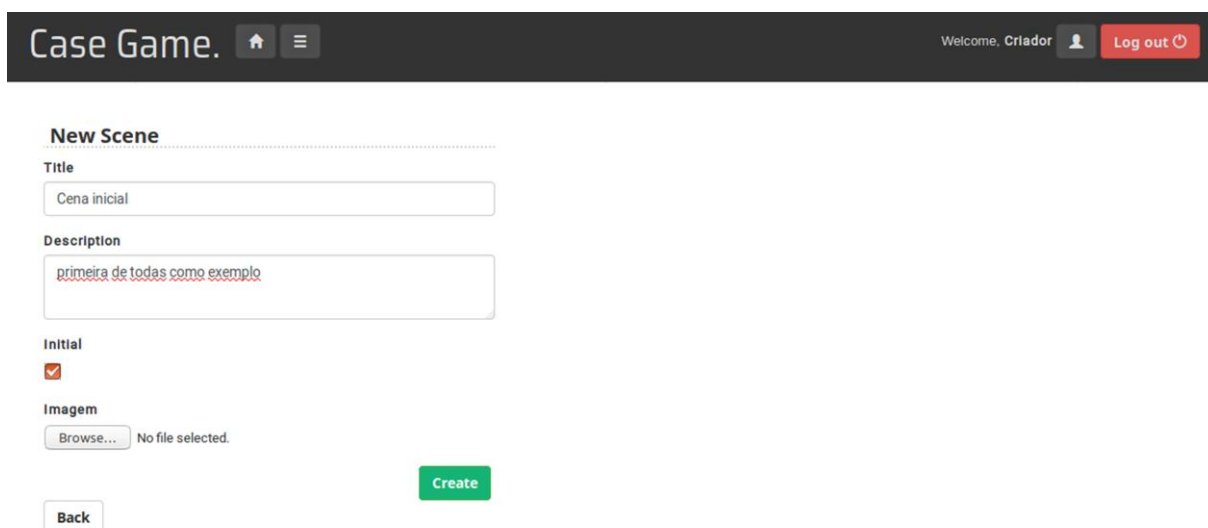


Figura 13: Tela de Inserção de Cenas

Uma mensagem de confirmação da cena criada aparece. Agora dentro da cena podemos editar ou excluir a mesma e adicionar as decisões.

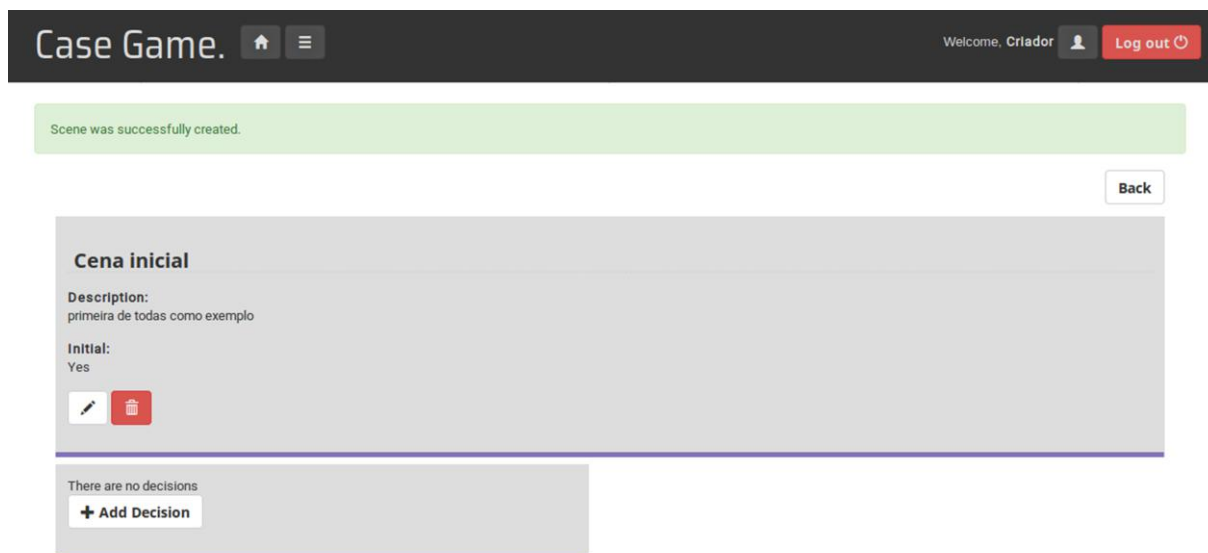


Figura 14: Tela de uma Cena e Gerenciamento de suas Decisões

Escolhemos o título da nossa decisão, e qual será o tipo dela: Múltipla Escolha para dar as opções de resposta, ou Resposta Invisível na qual não são dadas as opções e o jogador de papel Estudante deverá escrever uma das respostas corretas.

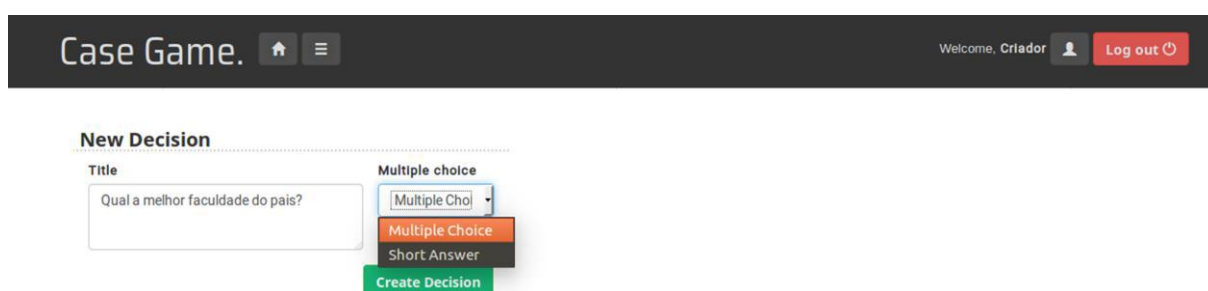


Figura 15: Tela de Inserção de uma Decisão

Na tela da decisão criada podemos visualizar o título e o tipo dela, caso o criador queira editar. Além disso podemos visualizar todas as respostas já criadas para essa decisão, podendo editar ou excluir as já criadas. Por último podemos criar outras opções de resposta. A resposta é constituída por um texto e pela cena seguinte a resposta escolhida. No caso da figura abaixo a resposta UFRJ leva o jogador para a cena denominada *Cena 2*.

Case Game.
Welcome, Criador
Log out

Answer was created successfully

Back

Qual a melhor faculdade do pais?

Title

Qual a melhor faculdade do pais?

Multiple choice

Multiple Choice

Update Decision

Answers

Answer	Next Scene
UERJ	Cena 2

Add Answer

Text

UERJ

Next scene

Cena 2

Create Answer

Figura 16: Tela de Gerenciamento de Decisões e Inserção de Respostas

Professor

O professor pode criar Rounds de Case Games já criados por um criador. Também pode editar os Rounds por ele já criados. Nenhum Round criado é excluído, apenas desativado, podendo ser reativado.

Na tela inicial ele visualiza os Rounds já criados, onde demonstra o nome do Round, a qual Case Game ele pertence, e o status dele que pode ser ativo (aberto) ou inativo. Aberto permite que um usuário com papel estudante crie uma partida desse Round para jogar, caso esteja fechado, os estudantes não conseguem criar a partida daquele Round.

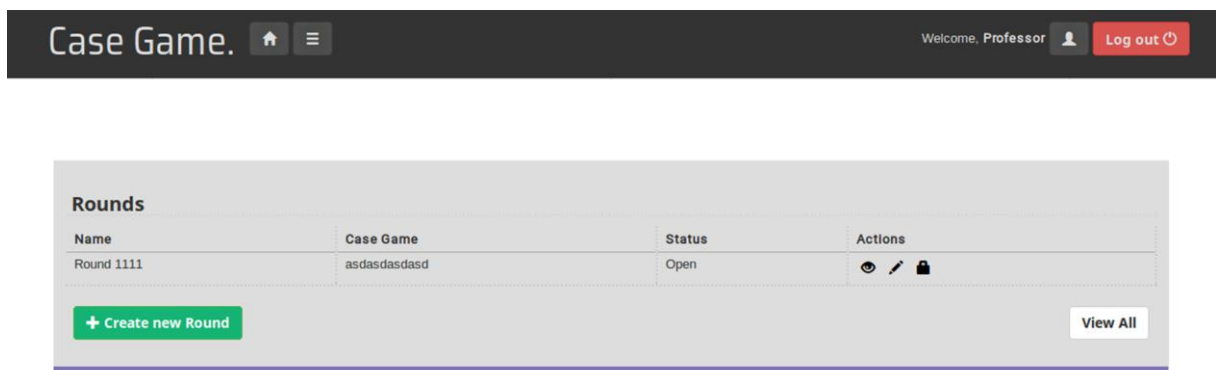


Figura 17: Tela de Gerenciamento de Rodadas

Na criação de um novo Round o usuário professor deve dar um nome para o Round, escolher de qual dos Case Games existentes ele vai pertencer e por último selecionar quais usuários com papel de estudante podem criar partidas desse Round, para isso basta escolher da lista da esquerda os que desejar, caso nenhum seja escolhido, todos os usuários poderão criar a partida do Round em questão.

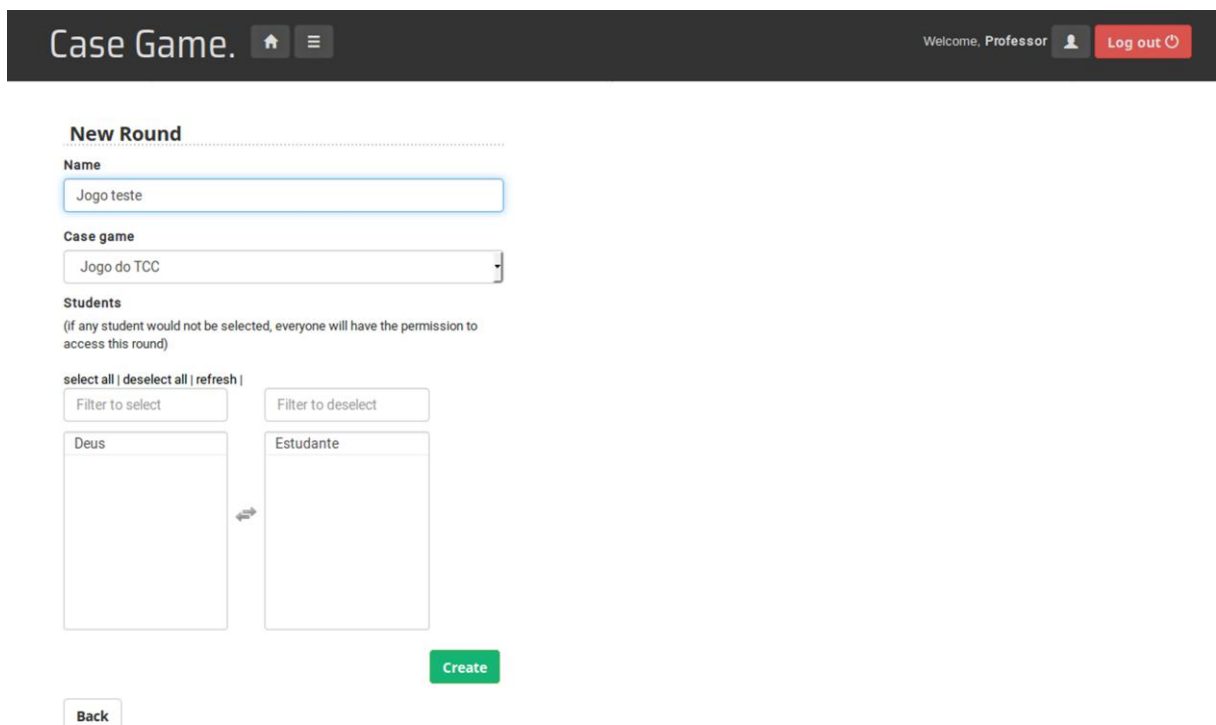


Figura 18: Tela de Inserção de Rodadas

Estudante

Ao entrar no sistema, o estudante pode gerenciar seus matches e visualizar os Rounds que pode participar.

O estudante pode criar e jogar uma ou várias partidas (Match) de um Round criado por um professor.

Na tela inicial do estudante ele visualiza os Rounds abertos por professores que selecionaram o próprio estudante para poder criar partidas. Além disso podemos ver as partidas já criados pelo estudante, permitindo continuar a partida de onde ele parou, e criar novas partidas.

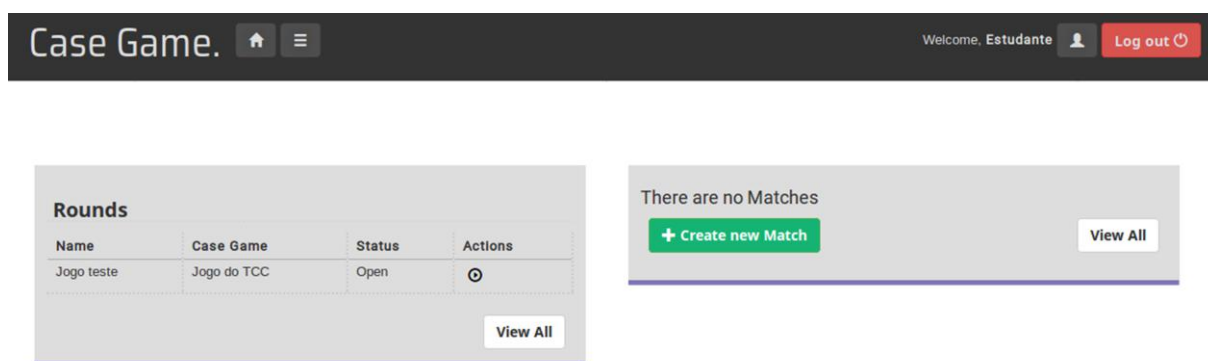


Figura 19: Tela Inicial de um Estudante

Para criar a partida é necessário dar um título a ela e escolher de qual dos Rounds abertos ela vai pertencer.

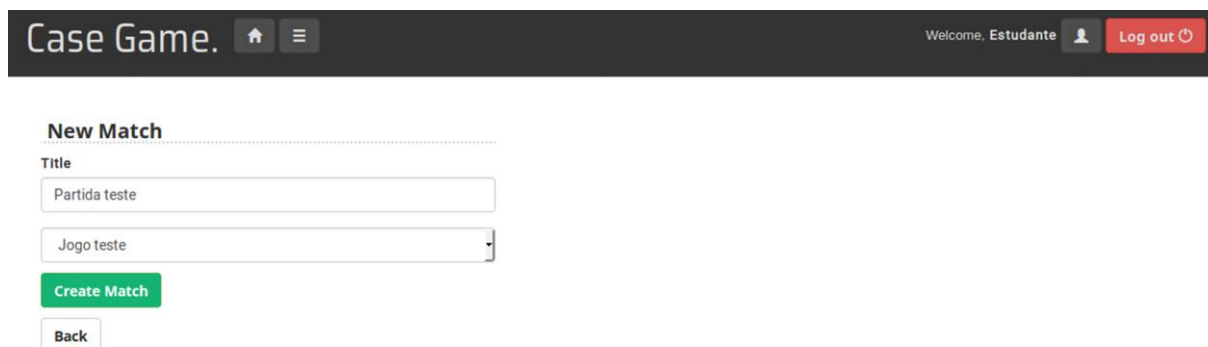


Figura 20: Tela de Criação de uma Partida

Ao iniciar a partida temos a cena inicial do case game com as opções de resposta, o aluno responde e clica no botão “Next” para avançar para a próxima cena do jogo.

Cada passo dado no fluxo da partida é uma instancia da entidade Score. Essa instância armazena os valores sobre qual jogo e aluno pertence além do que em qual etapa se encontra e qual a resposta dada na última decisão. Portanto, é possível visualizar todas as respostas da atual partida (Match).

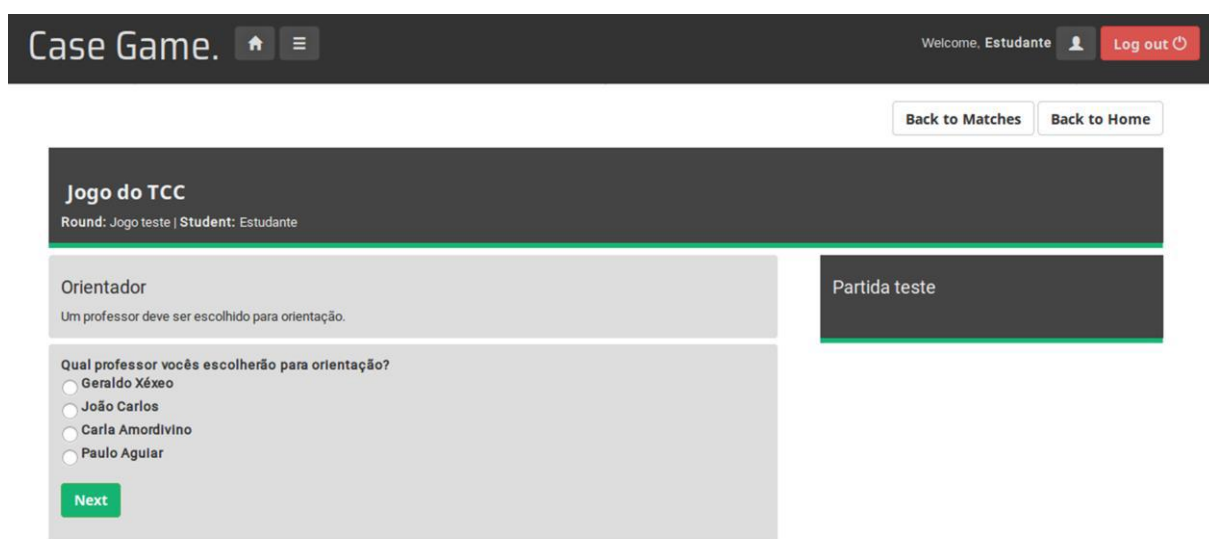






Figura 21: Tela Inicial de uma Partida Sendo Jogada

Ao responder a pergunta da cena, uma notificação é mostrada no topo da tela avisando que o placar foi criado. A tela do lado direito guarda a ordem das respostas dadas pelo jogador.

Case Game.



Welcome, Estudante  [Log out](#) 

Score was successfully created.

[Back to Matches](#) [Back to Home](#)

Jogo do TCC

Round: Jogo teste | Student: Estudante

Tema do TCC

Um tema deve ser escolhido para o trabalho de conclusão do curso.

Quanto componentes estarão no projeto?

☐ 3

☐ 1

☐ 2

☐ Mudar o tema

[Next](#)

Partida teste

Qual professor vocês escolherão para orientação?

A: Geraldo Xéxeo

Qual área vocês escolherão para o TCC?

A: Jogos

Figura 22: Tela de uma Partida Durante o Jogo

Ao chegar na última cena a mensagem de FIM é demonstrada e disponibiliza um botão para se criar uma nova partida. Lembrando que a última cena é uma cena sem decisão.

Caso o aluno saia do sistema ou volte para a tela inicial, o programa salva a última cena que ele respondeu e quando ele quiser retomar essa partida ele voltará para a cena salva.

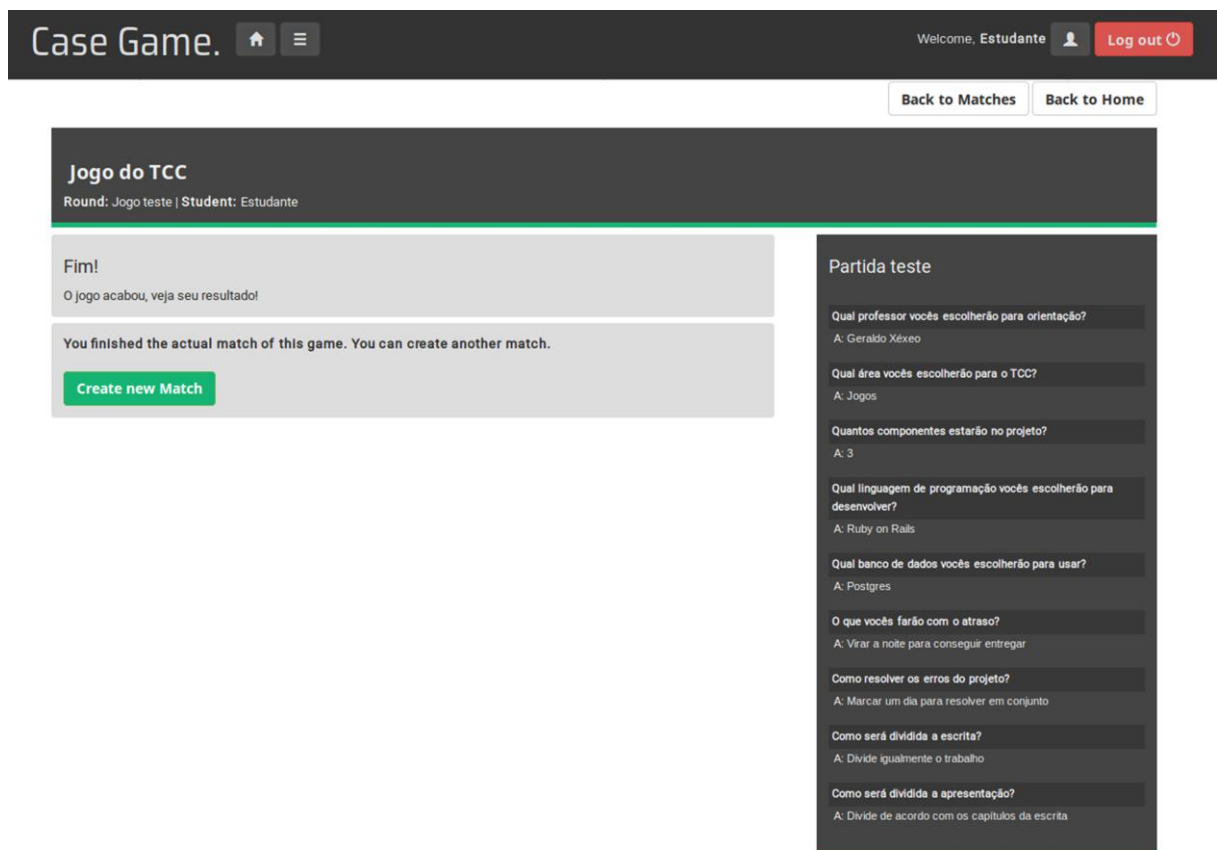


Figura 23: Tela Final de uma Partida

6 CONSIDERAÇÕES FINAIS

Nesse capítulo, avaliamos todas as dificuldades encontradas no projeto e o como resolvemos tais situações. Além de apontarmos possíveis melhorias para novas futuras versões.

6.1 DIFICULDADES ENCONTRADAS

Durante o desenvolvimento do projeto, esbarramos em algumas dificuldade, a maioria delas de âmbito técnico.

6.1.1 Linguagem de Programação Nova

Já no princípio do projeto, optamos por usar uma linguagem que nenhum dos membros do grupo já teria tido contato. Por essa questão escolhemos o Ruby e a ferramenta Ruby on Rails para o desenvolvimento. Como era o primeiro contato, tentamos aprender a ferramenta codificando o projeto diretamente e nos baseando na própria documentação.

Porém dessa forma, percebemos que para hospedar o projeto em um servidor de produção livre, precisaríamos mudar o banco de dados para o Postgres, que até então estávamos usando o SQLite que é o padrão do framework. Essa migração foi a primeira dificuldade encontrada no projeto.

6.1.2 Design Patterns

Um dos poderes da linguagem Ruby é a fácil utilização de design patterns. Don't Repeat Yourself, Tell Don't Ask, etc. Como o conhecimento do grupo nesses conceitos era muito raso, escrevemos nosso produto sem utilizá-los. Isso não foi um grande agravante no desenvolvimento da solução. Mas sim, algo que poderaa ter nos ajudado a escrever códigos mais limpos e legíveis e também a oportunidade que desperdiçamos para treinar esses conceitos.

6.1.3 Conceito do passo-a-passo do jogo.

Uma das primeiras grandes dificuldades foi a questão de fazer o jogo funcionar de fato. Apesar de já termos desenvolvido o conceito das decisões, que eram basicamente perguntas, em suas respostas apontavam para uma próxima cena, não conseguíamos pensar em como o usuário Estudante iria criar um jogo e armazenar o status desse jogo para que a qualquer momento pudesse voltar e continuar jogando. A solução foi bem simples, criamos uma nova tabela Score que armazenaria o passo atual do Estudante no jogo, e portanto, todos os valores necessário para tal, como o Match que estava sendo jogado, qual Round, qual Usuario adicionava, Case Game, Cena, Decisão e Resposta. Ou seja, a cada resposta, uma nova instância de Score era criada com todos esses valores. Além desses também o valor “step_number” que é o número do passo do aluno. A cada resposta adicionada, acrescia-se de uma unidade ao step_number, assim, podendo-se saber em que etapa do jogo, o aluno se encontrava. Apesar da solução simples, precisamos nós, os membros do grupo, nos reunir e também utilizar de técnicas de programação ágil para concluir essa etapa.

6.1.4 Conceito de respostas escritas arbitrariamente pelo usuário.

Uma última dificuldade encontrada durante o desenvolvimento do projeto foi a possibilidade do usuário com papel de estudante não precisar escolher uma resposta pré definida para a decisão, ou seja, a resposta múltipla-escolha, podendo escrever qualquer valor em um campo de resposta. A solução mais rápida e simples foi ter essa possibilidade para o usuário e o sistema apenas verificando se existia uma resposta pré definida que fosse igual ao que o estudante escreveu como resposta no campo de texto. Se não existir resposta pré-definida igual à escrita, o sistema informa que não existe essa resposta e o jogo permanece na mesma tela.

6.2 POSSÍVEIS TRABALHOS FUTUROS

Explicaremos agora, alguns possíveis módulos e evoluções que o projeto possa receber.

6.2.1 Variáveis Globais e critérios de resposta.

Como dito anteriormente, um dos possíveis incrementos do sistema seria a possibilidade do Estudante escrever qualquer valor em um campo texto como resposta. A ideia é que funcionasse envolvendo 3 novas entidades, Variáveis Globais, Critérios e Avaliadores:

As variáveis globais seriam definidas ao criar uma nova instância de Case Game. Um case game poderia ter diversas variáveis globais. Essa variável seria uma variável inicial do Case Game que seria modificada durante o jogo, no percorrer das respostas inseridas pelo Aluno.

O avaliador seria criado e instanciado na criação de decisões. Uma decisão poderia ter vários avaliadores. O avaliador indicaria o quanto que a resposta preenchida pelo usuário influenciaria no valor iniciado no case game pela Variável Global.

Os critérios seriam as restrições que indicariam em qual etapa o jogo estaria. Por exemplo, se uma variável global atingisse um valor menor/maior do que “x”, o jogo passaria para a cena “j” do jogo.

Ou seja, um Case Game teria uma Variável Global inicializada no jogo que poderia ser modificada pelas respostas do Estudante do jogo através de Avaliadores, que indicariam o quanto e como o valor preenchido modificaria essa variável global, e, dependendo dos critérios, mudaria o rumo do jogo, mostrando novas cenas.

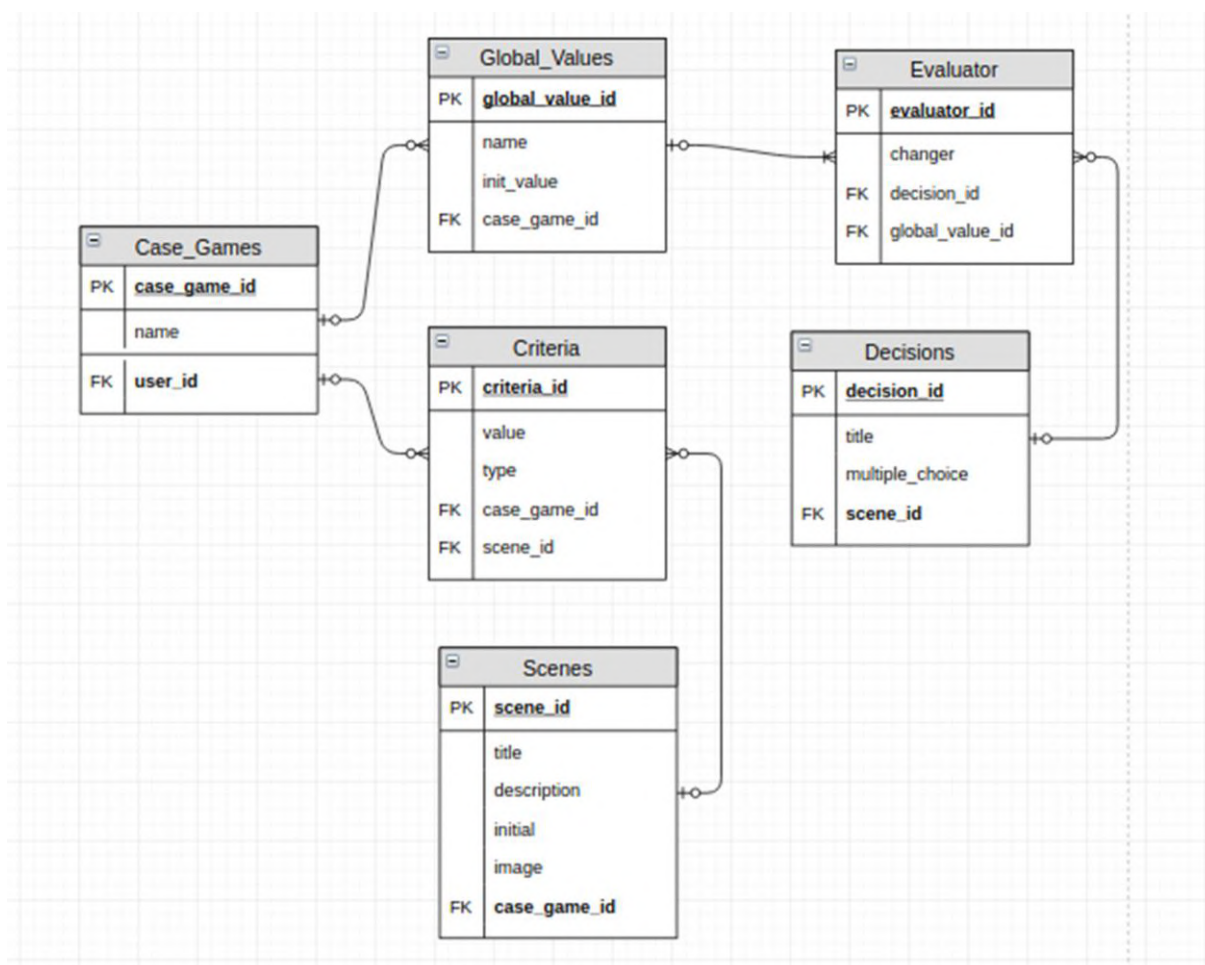


Figura 24: Diagrama de Classes da Melhoria de Critérios de Decisões e Variáveis Globais

Exemplo prático: Um case game de investimento forneceria a informação que o Estudante começa com um capital inicial de 20 mil reais. Essa seria nossa variável global: “Investimento”:”20000”.

Na primeira decisão, o jogo perguntaria quanto o aluno estaria apto a investir em determinado capital de risco. Essa decisão teria um avaliador dizendo que o valor respondido pelo aluno subtrairia da variável global “investimento”. Ou seja, o aluno respondendo 18.000, a variável global estaria com 2.000 (20.000-18.000).

Ao criar o Case Game, o criador do jogo poderia criar o seguinte Criterio: “Investimento” menor que “5000” tem como próxima cena “Cena final: Falta de recursos”.

6.2.2 Importação de Planilhas para regras do jogo

Uma ideia que pensamos no início do projeto, mas não foi possível de desenvolvê-la era a integração das regras do jogo com planilhas do Excel. A ideia é que o sistema pudesse importar uma planilha do excel com restrições parecidas com um problema de Programação Linear, em que o Excel utiliza do Solver, por exemplo, para resolvê-las. Como não chegamos a tempo de pensar no desenvolvimento dessa feature, não conseguimos detalhá-la com muita profundidade.

6.3 CONCLUSÃO

Esse projeto foi criado e desenvolvido para dar início a um processo capaz de melhorar a avaliação de cursos com ambiente tecnológico. Um pequeno passo na área de Tecnologia da Informação para melhorar e/ou auxiliar metodologias de avaliação em cursos de diversas graus de escolaridades.

Foi perceptível no decorrer do projeto, que a falta de conhecimento inicial das tecnologias utilizadas nos dificultou na criação de um código atualizado nas boas práticas de programação. Porém, fez-se valer o aprendizado de novas tecnologias e saímos da zona de conforto, todos os membros da equipe paralelamente, o que provocou grande busca de aprendizado e comunicação.

As ferramentas de gerenciamento de tarefas também nos auxiliaram e nos tornou possível o desenvolvimento de um sistema de forma não presencial da equipe. Reuniões presenciais esporádicas foram cruciais no gargalo de grande problemas do desenvolvimento de requisitos.

Como descrito nesse documento muito ainda há e pode ser feito. O projeto abre espaço para outros desenvolvedores criarem e pensarem novas funcionalidade em um sistema de avaliação de curso.

Práticas, metodologias e ferramentas dentro do mundo de desenvolvimento também podem ser utilizadas no projeto. Aplicação de Teste Unitários; utilização de técnica TDD (*Test Driven Development*) para diminuir o tempo de codificação corrigindo bugs; criação de uma API do back-end, transformando também a aplicação em RESTful; e utilização de ferramentas como *Vagrant* ou *Docker* para padronização de configurações de ambiente e integração contínua.

Resumidamente, o projeto ao final nos fez praticar a parte de desenvolvimento de um sistema Web, estudar sobre novas ferramentas, a busca por novos conhecimentos teóricos de programação e a utilização de boas técnicas de gerenciamento de projetos e de equipe. E como resultado final um sistema que é uma inicialização de uma maneira de avaliação de forma automatizada, de fácil uso e divertida.

REFERÊNCIAS

8 MOTIVOS PARA ESTUDAR RUBY ON RAILS. Disponível em: <<https://udgwebdev.com/8-motivos-para-estudar-ruby-on-rails/>>. Acesso em: 05 maio 2018.

CHRISTENSEN, Chris. **Teaching by The Case Method**: Case Method in Practice. Disponível em: <<http://www.hbs.edu/teaching/case-method/Pages/default.aspx>>. Acesso em: 05 maio 2018.

DESENVOLVIMENTO ÁGIL PARA WEB COM RUBY ON RAILS. Disponível em: <<https://www.caelum.com.br/apostila-ruby-on-rails/>>. Acesso em: 05 maio 2018.

DIAS, André Felipe. **Qual a melhor ferramenta de controle de versão**: subversion, git ou mercurial. Disponível em: <<https://blog.pronus.io/posts/qual-a-melhor-ferramenta-de-controle-de-versao-subversion-git-ou-mercurial/>>. Acesso em: 05 maio 2018.

GRANDO, R. C. **O jogo na educação: aspectos didático-metodológicos do jogo na educação matemática**. Unicamp, 2001

HUIZINGA, J. **Homo ludens**. São Paulo: Perspectiva, 2005.

O'GRADY, Stephen. **The RedMonk Programming Language Ranking**. Disponível em: <<http://redmonk.com/sograde/2017/06/08/language-rankings-6-17/>>. Acesso em: 05 maio 2018.

VELOSO, Rosângela Ramos; SÁ, Antônio Villar Marques. **Reflexões sobre o jogo**: conceitos, definições e possibilidades. Disponível em: <<http://www.efdeportes.com/efd132/reflexoes-sobre-o-jogo.htm>>. Acesso em: 05 maio 2018

VYGOTSKY, L. S. O papel do brinquedo no desenvolvimento. In: VYGOTSKY, L. S. **A formação social da mente**. São Paulo: Ed. Martins Fontes, 1989. p. 106-118.