

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

AprEnDO: aplicativo didático para suporte no aprendizado de EDO's - um relato de experiência

Autor: Leonardo Arthur Degolim Oliveira
Orientadora: Prof^a Dr^a Tatiane da Silva Evangelista
Data de apresentação: 05/07/2019

Brasília, DF
2019



Leonardo Arthur Degolim Oliveira

AprEnDO: aplicativo didático para suporte no aprendizado de EDO's - um relato de experiência

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

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientadora Prof^a Dr^a Tatiane da Silva Evangelista

Brasília, DF

2019

Leonardo Arthur Degolim Oliveira

AprEnDO: aplicativo didático para suporte no aprendizado de EDO's - um relato de experiência/ Leonardo Arthur Degolim Oliveira. – Brasília, DF, 2019-
70 p. : il. (algumas color.) ; 30 cm.

Orientadora Prof^a Dr^a Tatiane da Silva Evangelista

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2019.

1. Aplicativo. 2. EDO 1^a ordem. I. Prof^a Dr^a Tatiane da Silva Evangelista.
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. AprEnDO: aplicativo
didático para suporte no aprendizado de EDO's - um relato de experiência

CDU 02:141:005.6

Leonardo Arthur Degolim Oliveira

AprEnDO: aplicativo didático para suporte no aprendizado de EDO's - um relato de experiência

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

Prof^a Dr^a Tatiane da Silva Evangelista
Orientadora

**Prof^o Dr. Ronni Geraldo Gomes de
Amorim**
Convidado 1

Prof^o Dr. Fábio Macêdo Mendes
Convidado 2

Brasília, DF
2019

Resumo

O objetivo deste trabalho foi desenvolver um jogo de celular o AprEnDO, para analisar o apoio à aprendizagem de equações diferenciais ordinárias (EDO) de 1^a ordem. O jogo contém perguntas a respeito de classificação e resolução dessas equações. A metodologia de trabalho foi um relato de experiência de uma aplicação de jogo em uma classe de Cálculo 2 (C2) da Faculdade do Gama da UnB em que a professora orientadora ministrou o ensino. Em 1/2019 a turma começou a utilizar o jogo e foram aproximadamente 2 meses de coletas em um servidor de dados para relatar a experiência do AprEnDO em sala de aula e verificou-se a baixa participação dos alunos. Como método de validação para a organização do conhecimento na estrutura cognitiva houve o uso de mapas conceituais.

Palavras-chave: Equação Diferencial. Aplicativo para celular. Jogo. Software.

Lista de ilustrações

Figura 1 – Processo de montagem do componente React	20
Figura 2 – Processo de atualização do componente React	21
Figura 3 – Prototipação das telas iniciais do jogo	25
Figura 4 – Prototipação das telas dos módulos	26
Figura 5 – Nova tela de resolução	27
Figura 6 – Prototipação das telas dos módulos	28
Figura 7 – Processo geral do jogo aprEnDO	31
Figura 8 – Processo de inicialização do jogo aprEnDO	32
Figura 9 – Processo de jogo de classificação	33
Figura 10 – Processo de ganhar uma fase	33
Figura 11 – Processo de jogo de resolução	34
Figura 12 – Envio de estatísticas	34
Figura 13 – Zerar estatísticas após envio	35
Figura 14 – Diagrama de Pacote do AprEnDO	35
Figura 15 – Diagrama de Classe do AprEnDO	37
Figura 16 – Tela inicial	47
Figura 17 – Modo de classificação parte 1	49
Figura 18 – Modo de classificação parte 2	50
Figura 19 – Modo de classificação parte 3	51
Figura 20 – Exemplo da fase de classificação de homogeneidade	52
Figura 21 – Exemplo da fase de classificação de tipo	53
Figura 22 – Feedback fornecido à uma tentativa inválida	54
Figura 23 – Tela de finalização do modo de classificação	55
Figura 24 – Modo de resolução parte 1	56
Figura 25 – Modo de resolução parte 2	57
Figura 26 – Jogo da memória com 1 carta selecionada	58
Figura 27 – Jogo da memória com alguns pares encontrados	59
Figura 28 – Vitória de jogo da memória	60
Figura 29 – Tela de envio de estatísticas	61
Figura 30 – Alerta de matrícula inválida	61
Figura 31 – Alerta de falta de internet	62
Figura 32 – Loader visível enquanto envia estatísticas	63
Figura 33 – Alerta de envio com sucesso	63
Figura 34 – Estatísticas colhidas de classificação	65
Figura 35 – Estatísticas colhidas de resolução	65

Listas de abreviaturas e siglas

APP	Aplicativo
API	<i>Application Programming Interface</i>
C2	Cálculo 2
Cap.	Capítulo
ED	Equação Diferencial
EDO	Equação Diferencial Ordinária
FGA	Faculdade do Gama
JSON	<i>JavaScript Object Notation</i>
MC	Mapa Conceitual
OCDE	Organização para a Cooperação e Desenvolvimento Econômico
PISA	Programa Internacional de Avaliação de Estudantes
PVI	Problemas de valor inicial
RBIE	Revista Brasileira de Informática na Educação
RENOTE	Revista de Novas Tecnologias na Educação
UnB	Universidade de Brasília
VI	Valor Inicial

Sumário

1	INTRODUÇÃO	9
2	REFERENCIAL TEÓRICO	12
2.1	Sucesso de jogos no ensino	12
2.2	Teoria Ausubel e Mapa Conceitual	15
2.3	Engenharia de software	17
2.3.1	Levantamento de requisitos de software	18
2.3.2	Digital Game Based Learning LGBL	18
2.3.3	Características dos Jogos Educacionais	19
2.3.4	Plataformas mobile	19
2.3.4.1	React Native	19
3	METODOLOGIA	22
4	ENGENHARIA DE SOFTWARE	23
4.1	Requisitos de Software	23
4.2	Prototipação	24
4.3	Banco de equações	28
4.3.1	Como se pensou	28
4.3.2	Como está hoje	29
4.4	AprEnDO	30
4.5	Servidor aprEnDO	44
4.6	Ambiente de desenvolvimento	45
4.7	Acesso ao código	46
5	MANUAL DO APRENDO	47
5.1	Pacote 1 - Jogo	47
5.1.1	Módulo 1: Classificação	47
5.1.2	Módulo 2: Resolução	55
5.2	Pacote 2 - Servidor	60
6	APRENDO: O RELATO DE EXPERIÊNCIA E A ANÁLISE	64
6.1	Relato de experiência	64
6.2	Análise	64
7	CONCLUSÃO	68

1 Introdução

Sabe-se que os jogos propiciam diversão ao seres humanos e vê-se que a gamificação está sendo explorada em várias áreas diversas como marketing, saúde e educação. Segundo (NETO; BLANCO; SILVA, 2017), as tecnologias digitais em sala de aula podem ser instrumentos que auxiliam no processo de ensino e de aprendizagem. Acredita-se que os jogos e os aplicativos gamificados podem ser usados na educação para ajudar mais pessoas a aprender e se interessarem pelo assuntos abordados. Será que usar jogos digitais para o ensino de Matemática pode ajudar os alunos a se interessarem e aprenderem mais? Se sim, como utilizar?

No Brasil os jovens tem dificuldades com Matemática. Segundo o relatório do (INEP, 2015) o Brasil tem uma qualidade de ensino de Matemática inferior a de muitos países. A Organização para a Cooperação e Desenvolvimento Econômico (OCDE) utiliza uma escala de classificação que vai de 1 a 6 para as habilidades de Matemática e de acordo com o conjunto da população brasileira que participou da pesquisa no Programa Internacional de Avaliação de Estudantes (PISA), inferiu-se que 70.3% dos estudantes brasileiros estão abaixo do nível 2, o qual foi estabelecido como o mínimo para exercer a cidadania como cidadão pleno (INEP, 2015). O estudo é realizado a cada três anos. Foi realizado no ano passado, porém até o momento o resultado ainda não foi publicado <<http://portal.inep.gov.br/web/guest/acoes-internacionais/pisa/resultados>>.

Além das informações relatadas acima outras duas também serviram para justificar e concretizar a proposta deste trabalho.

Um deles é o estudo de (SOUZA, 2016) o qual diz que Cálculo 2 (C2) é uma disciplina das que mais causa a evasão dos alunos do curso de Matemática noturno na UnB. Após saber da existência do estudo no site de monografias da UnB houve a comparação da ementa de C2 no curso de Matemática noturno do Darcy Ribeiro com a ementa de C2 do tronco comum no curso de Engenharias da UnB no Gama e foi constatado que há a equivalência dos conteúdos ministrados das disciplinas. A disciplina para os dois cursos introduz aos alunos o conteúdo de equações diferenciais, assunto este que por acaso é pouco encontrado na literatura de gamificação, o mais encontrado é voltado para área de limite, derivada e integral como mostrado o aplicativo em (SILVA *et al.*, 2016) e também pode ser observado na própria FGA. Verificou-se que há poucos jogos de Matemática para o conteúdo de equação diferencial. O único jogo de equações diferenciais encontrado é um de vídeo-game que aplica as equações diferenciais na movimentação dos personagens (princípio da dinâmica de Newton)(GIACINTI *et al.*, 2013). A maioria dos jogos relacionando números e equações é para o ensino fundamental.

Existem também estudos mostrando como a tecnologia da suporte para jogos na hora do ensino e ajuda na fixação do conhecimento.

Tendo relatado as dificuldades no ensino de Matemática e a possível ajuda de gamificação e jogos digitais, gerou-se a questão: Como dar suporte no ensino de EDO 1^a ordem apresentados em sala de aula de forma lúdica? A partir deste questionamento, levantou-se o objetivo: desenvolver um jogo para celular Android que dê suporte ao ensino de equações diferenciais ordinárias (EDO) de 1^a ordem. O jogo visa treinar os alunos a reconhecer, classificar e resolver equações diferenciais presentes no dia-a-dia e no ambiente da engenharia.

Decidiu-se fazer um jogo para celular com o intuito de inserir no ambiente dos alunos uma ferramenta a mais (o jogo) para ajudar os estudantes a aprenderem mais e se possível se divertindo, ou que pelo menos seja interessante. Deseja-se que seja um meio de treinamento e fixação do conhecimento para aprenderem.

A primeira parte da metodologia seguida foi a pesquisa bibliográfica para levantar o referencial teórico a respeito da contribuição efetiva de jogos e seu sucesso no ensino. Já a segunda parte da metodologia foi um relato de experiência do desenvolvimento do software e a aplicação em alunos de C2 na Universidade do Gama, utilizando estatísticas levantadas do jogo chamado aprEnDO para avaliar o impacto do aplicativo educacional para celular como ferramenta de suporte didático no ensino principalmente em classificação de EDO de 1º nível.

Utilizou-se o processo de prototipação de baixa fidelidade para desenhar as telas do jogo pois ela serve como auxílio para uma das etapas do processo de desenvolvimento de software que é a elicitação dos requisitos. Houve a consolidação dos requisitos funcionais que foram definidos e a modelagem de um diagrama de classe para explicar a aplicação AprEnDO que é um projeto do *react native*. São indicados os domínios utilizados para baixar as dependências que compõem o software e toda a parte da hospedagem dos códigos tanto da aplicação, como para gerar o arquivo de alimentação e os que definiram o servidor que guarda os dados das estatísticas enviadas pelos alunos. O projeto pode ser reutilizado por quem desejar e todo interesse em ajudar na contribuição é bem vindo. Após ter o jogo pronto foi aplicado em uma turma de C2 no período do primeiro semestre de 2019 para que pudessem ser gerados dados e estatísticas para concluir se o jogo trouxe alguma eficiência no aprendizado ou não.

O jogo é separado em dois pacotes. O primeiro deles com dois módulos para jogar com diferentes fases e dificuldades e o outro pacote lida com o envio de estatísticas dos dados dos jogadores para um servidor com um banco de dados. Foi desejado planejar o pacote 2 para ser "fácil" de enviar os dados colhidos para análise. É chamado de fácil por ser considerado rápido do jogador utilizar. Exige-se apenas a matrícula do aluno (que

será validada) e a confirmação de um clique para acordar compartilhar suas estatísticas de variações entre o tempo atual e o último envio.

Houve uma falta de planejamento no desenvolvimento de software devido aos atrasos de elicitação dos requisitos o que impossibilitou a completude de alguns e como o jogo havia data certa para início de aplicação impossibilitou o término do desenvolvimento de alguns requisitos.

O Capítulo 2 abordará o referencial teórico, apoiando a gamificação e jogos como uma prática que deixa as tarefas e atividades mais divertidas, descrevendo relatos a respeito do uso de mapas conceituais para evidenciar a presença de conhecimento na estrutura cognitiva e revisando conceitos de Engenharia de Software para desenvolvimento de jogos. O Capítulo 3 explica a metodologia do trabalho seguida e o que se espera encontrar como conceito na avaliação dos mapas mentais. O Capítulo 4 fala a respeito da Engenharia de Software aplicada no projeto, cita requisitos do jogo, artefatos de auxílio, ambiente de desenvolvimento, o banco de dados, como é o processo de empacotamento e o servidor que recebe dados de jogo. O Capítulo 5 é o manual do AprEnDO que explica as fases do jogo, como espera-se que ele seja jogado e fotos do aplicativo. O Capítulo 6 mostra a experiência da aplicação e a análise dos dados colhidos e dos mapas conceituais. O capítulo 7 apresenta a conclusão do trabalho e o possíveis atividades futuras e por fim as referências do trabalho.

2 Referencial Teórico

O referencial teórico presente discorrerá a respeito de três temas para defender a ideia deste trabalho. O primeiro tema reforça o uso de jogos para aplicação na educação e comenta a respeito da falta de jogo de Matemática no ensino superior. O segundo tema justifica o uso do mapa conceitual como uma ferramenta de validação de conhecimento adquirido e por fim o terceiro tema trará conceitos da engenharia de software para um bom processo de desenvolvimento de software em jogos e/ou aplicações gamificadas.

2.1 Sucesso de jogos no ensino

Alguns autores diferenciam jogos e gamificação, porém neste estudo ambos serão tratados de maneira igual querendo se referir à atividades lúdicas. Esta seção visa apoiar jogos como uma estratégia boa para aprendizado dos alunos/jogadores e também como uma boa ferramenta para ser utilizada como aprendizagem. Espera-se que com o uso de jogos os estudantes gostem e se atrevam mais no contexto da Matemática buscando por conta própria o conhecimento e que se tornem mais independentes.

No século XXI com todo o avanço tecnológico o celular se tornou um aparelho indispensável para o ser humano, então nada melhor do que unir o útil ao agradável e utilizá-lo como uma ferramenta de auxílio e de motivação para os estudantes (SANTOS, 2017). O uso de computadores ou tecnologias da informação como o celular é útil para melhorar o engajamento nas tarefas, principalmente com exercícios e aplicações para a prática das matérias ensinadas em sala de aula (DUPAUL; STONER, 2007).

Como a finalidade na sala de aula é estimular ideias dos alunos, ensinar apenas com aulas expositivas tradicionais pode dificultar o aprendizado. Por isso os jogos chegaram às salas de aula para ajudar (SOUZA; FRANÇA, 2016, p. 4). De acordo com o dicionário Michaelis, uma das definições de jogo é "Qualquer atividade recreativa que tem por finalidade entreter, divertir ou distrair" (MICHAELIS, 2019). Alguns sinônimos são: brincadeira, entretenimento, diversão, entre outros. O jogo tem além do seu papel lúdico, a característica da construção de conhecimento, ou seja, fazer com que o jogador aprenda através das experiências adquiridas por ele (COELHO, 2010) (SILVA *et al.*, 2016) (NUNES; PARREIRA, 2015) (FUKUSAWA; CARNIELLO, 2015). De acordo com (COELHO, 2010), o caráter lúdico proporciona ao aluno uma participação efetiva no processo de ensino aprendizagem se tornando um momento ímpar de crescimento pessoal e coletivo.

Os jogos e a gamificação se mostraram como uma estratégia efetiva tanto no ensino

fundamental quanto no ensino superior. Na literatura, encontram-se vários trabalhos que demonstram profissionais de educação utilizando os jogos como ferramenta de auxílio ao aprendizado (SOUZA; FRANÇA, 2016, p. 3). Abaixo seguem alguns exemplos.

- Um jogo chamado "O bicho papão da Matemática virou um gatinho" que ensina Matemática para alunos do 1º e 2º ano fundamental. Este ajuda os alunos a fazer divisões. Segundo a notícia disponível no portal [mec](#), 300 alunos se beneficiaram deste projeto. Um dos símbolos que ficou marcado era de uma aluna que tinha reprovado, tirava notas baixas, não interagia muito com os outros alunos e acabou se envolvendo, aprendendo o jogo de tal maneira que passou a tirar 10, ir resolver no quadro e se sentir capaz. É relatado também que não melhorou só em Matemática, como em outras matérias. Segundo o professor, além da menina citada, muitos outros alunos que não sabiam divisão aprenderam também.
- O estudo (SOUZA; FRANÇA, 2016) realizou uma pesquisa nos Anais da base WEI para selecionar artigos que contém jogos que auxiliam no aprendizado das disciplinas de Engenharia de Software. Foram selecionados preferencialmente os estudos que validassem os resultados com alunos (SOUZA; FRANÇA, 2016). Como resultado foi obtido um levantamento de 20 jogos focados em disciplinas do curso, dentre elas gestão de projetos, algoritmos, estrutura de dados, teste de software entre outras.
- O estudo (SILVA *et al.*, 2016) é uma proposta de aplicativo gamificado para ensino de Cálculo 1, onde é proposto um jogo para o ensino de Matemática com o conteúdo voltado para os temas de conjunto, limite, derivada e integral.
- O estudo de (KATMADA; MAVRIDIS; TSIATSOS, 2014) cita um jogo educacional de matemática com um agente que ensina aritmética que é capaz de aprender. O programa utiliza de inteligência artificial.

Alguns dos motivos pelo qual os profissionais da educação optam pela utilização de jogos são:

- Despertar o interesse dos jovens e trazer diversos benefícios à educação (SILVA *et al.*, 2016).
- O jogo é uma prática que ajuda na concretização do conhecimento, além de tornar o ambiente mais prazeroso (COELHO, 2010).
- Motivação e aprendizado por descoberta (FUKUSAWA; CARNIELLO, 2015 apud SAVI; ULBRICHT, 2008, p. 2).

- Desenvolvimento de habilidades cognitivas, experiência de novas identidades, socialização, coordenação motora e comportamento expert ([SAVI; ULRICHT, 2008](#), p. 3 e 4).
- Poder proporcionar a vivência em experiências de aprendizagem concretas ([MONSALVE; WERNECK; CESAR, 2010](#) apud [SOUZA; FRANÇA, 2016](#), p. 4).

De acordo com ([NUNES; PARREIRA, 2015](#)), sabe-se que os jogos educacionais têm sido intensamente utilizados por profissionais da área de educação como auxílio para a construção do conhecimento e para motivar os alunos, porém os resultados encontrados mostraram que apesar de existirem muitos jogos, poucos são de Matemática. Dos jogos de Matemática encontrados, foi constatado que uma minoria deles são para o ensino superior, e dos jogos de Matemática para o ensino superior apenas um deles falou de equações diferenciais, que é o estudo ([GIACINTI et al., 2013](#)) o qual aplica os conceitos na movimentação dos personagens (princípio da dinâmica de Newton), porém o jogo não é focado no ensino de equação diferencial ordinária.

O estudo de ([NETO; BLANCO; SILVA, 2017](#)) é uma revisão sistemática de literatura realizada nas bases de dados Scielo Library, BIREME Biblioteca, Science Direct, ACM Library e IEEE Xplore Digital Library e nos periódicos Revista Brasileira de Informática na Educação e a Revista de Novas Tecnologias na Educação. O estudo procurou artigos que constatavam a existência de ferramentas relacionadas com gamificação e dificuldades de aprendizagem de matemática. De 2008 trabalhos, nenhum eram relacionando gamificação e dificuldades de matemática. Por fim o estudo concluiu que:

identifica-se a necessidade de pesquisas sobre esta temática, já que as dificuldades de aprendizagem na Matemática são frequentes em sala de aula, e a gamificação tem-se mostrado uma ferramenta promissora nos ambientes de ensino e aprendizagem em todos os níveis de ensino.

No entanto para ([DICHEVA et al., 2015](#)) que é citado no estudo ([NETO; BLANCO; SILVA, 2017](#)), a falta de pesquisa nesta área é justificada por ser uma temática nova. Acredita-se que com o tempo o número de interessados e pesquisas na área aumentará.

Com todas as constatações acima, espera-se ter convencido que muitos autores já estão utilizando jogos ou aplicações como uma ferramenta de auxílio para o ensino e que em alguns casos elas são um sucesso, fazendo com que de fato os jogadores aprendam o conteúdo que o jogo deseja transmitir. Constatando que há poucos jogos de Matemática para equações diferenciais este trabalho trata-se do desenvolvimento de um jogo para o ensino de matemática à alunos do ensino superior.

2.2 Teoria Ausubel e Mapa Conceitual

Na teoria de Ausubel de aprendizagem significativa uma variável bastante crucial é a estrutura cognitiva do aprendiz (SUBJACENTE, 2011, p. 26), pois é nela onde o novo conhecimento irá se fixar. O novo conhecimento deve se relacionar de uma maneira não arbitrária e substantiva na estrutura cognitiva (SUBJACENTE, 2011, p. 26).

Acredita-se que para tornar mais eficiente a absorção do conhecimento dos alunos, para analisar o ensino/aprendizado deve-se olhar na esfera do alunos e não dos professores. Como assim? Ao invés do professor ensinar para o aluno, visão essa que torna o aluno como reativo, devemos olhar do ponto em que o aluno aprende com o professor. Desta segunda maneira é uma visão pró-ativa, onde o aluno não tem apenas o professor como única fonte de absorção de conhecimento, mas como uma das fontes. Nesta segunda visão o estudante deve ir atrás de como adquirir o conhecimento e seguir conselhos de pessoas mais experientes (como os professores).

O mapa conceitual (MC) é uma maneira de se alcançar a aprendizagem significativa, mas primeiro de tudo, o que é um MC? De acordo com (NOVAK, 2000 apud SOUZA; BORUCHOVITCH, 2009), "mapa conceitual é uma estrutura hierárquica, iniciados por conceitos mais abrangentes, os quais progressivamente vão sendo relacionados com conceitos mais específicos e esclarecendo suas relações de subordinação".

Um MC é formado por conceitos e palavras de enlace. Um conceito pode ser uma palavra ou uma expressão chave identificado geralmente em um retângulo ou alguma outra forma e palavras de enlace podem ser uma palavra ou uma expressão que conecta conceitos de modo que dois conceitos conectados por uma palavra ou expressão de enlace é chamado de proposição. Podem ser formadas proposições verdadeiras ou falsas, por isso deve-se analisar um mapa conceitual para checar se as proposições formadas não contém erros.

O MC pode ser alterado tanto em profundidade quanto em extensão. Profundidade se refere a especificação de algum conceito geral granularizando-o em novas proposições, ou seja, novos conceitos e palavra de enlace. Extensão se refere à adição de novos conceitos, porém não a granularização de um conceito anterior.

Um estudo lido foi o "Mapa conceitual: seu potencial como instrumento avaliativo", ele ajudou a entender vantagens e desvantagens de um mapa conceitual. Foi realizado com trinta e duas alunas de pedagogia do terceiro semestre que utilizaram o mapa conceitual para sintetizar informações de muitos textos que foram lidos. Todos os mapas eram apresentados para a turma e havia o debate dos mapas em relação a estar abordando a maioria dos conceitos chaves dos textos. As alunas eram identificadas como Ax (sendo x o número de um a trinta e dois, que é a quantidade de participantes). Foram realizados entrevistas, questionários e também solicitado aos grupos que registrassem três vantagens e desvantagens.

gens para avaliar a experiência. Vantagens do MC segundo (SOUZA; BORUCHOVITCH, 2009):

- identificar as dificuldades de aprendizagem

A2 diz que "mapas conceituais tornam os conhecimentos mais claros no que se sabe ou não, porque evidencia o que foi aprendido, mostrando também dúvidas, dificuldades e erros" (SOUZA; BORUCHOVITCH, 2009).

- Favorecer a reelaboração de conceitos a sua consequente sedimentação.
- Proporcionar feedback quase imediato.

O MC deixa claro a reorganização cognitiva, pois os conceitos conforme são aprofundados e entendidos melhor vão se estendendo e as proposições formadas são alteradas. (SOUZA; BORUCHOVITCH, 2009)

- Integração e ampliação dos conhecimentos.

"O trabalho com mapas conceituais nos levou a aprender a identificar os elementos essenciais e inter-relacioná-los" A8 (SOUZA; BORUCHOVITCH, 2009).

Mapas conceituais são ferramentas de avaliações de conhecimento, a partir dele é possível ver onde as pessoas estão errando nos conceitos e nas proposições, também é possível avaliar o quão extenso é a rede de conhecimentos da pessoa em determinado tema. (SOUZA; BORUCHOVITCH, 2009).

Exige capacidade e certo treino e conhecimento para conseguir organizar as informações existentes em seu cérebro de modo a expô-los em conceitos e conectores. Neste estudo (CORREIA; VALLE; SILVA, 2009) cita a dificuldade de corrigir mapas conceituais comparado a questões de múltipla escolha, por isso diz ser necessário os alunos estarem incluído pois os alunos aprendem mais estando incluídos na correção podendo comparar com os outros ao mesmo tempo com o seu.

Em (SOUZA; BORUCHOVITCH, 2009) é dito que o mapa conceitual além de ser uma ferramenta avaliativa, também se configurou como estratégia de aprendizagem, vantagem enunciada por 31% das duplas participantes.

Neste estudo (SOUZA; BORUCHOVITCH, 2009) que as alunas utilizaram o MC e depois avaliaram o uso, 38% das alunas declararam que o mapa conceitual possibilita efetivar sucessivas síncreses, análises e sínteses, porém precisa ser discutido em conjunto para sempre aumentar a compreensão.

Deseja-se utilizar o MC como ferramenta que avalia o verdadeiro aprendizado, para isso o mapa conceitual precisa ser capaz de evidenciar que existe o conhecimento e a informação presente, e que também as proposições sejam verdadeiras.

Exemplo de trabalho que utilizou mapa conceitual (COSTA; SILVA; RIBEIRO, 2012) pois está de acordo com (NOVAK, 2000) e acredita também que mapas conceituais contribuem para o ensino-aprendizagem.

Mapa conceitual não basta aumentar o tamanho da rede e de ligações, é necessário ter uma análise de conteúdo, se os conceitos estão relacionadas ao tema e se as proposições são verdadeiras ou podem ter sido mal compreendidas (SOUZA; BORUCHOVITCH, 2009).

Com os relatos acima do estudo (SOUZA; BORUCHOVITCH, 2009) citou a respeito das alterações cognitivas que podem acontecer ao se desenhar um MC com proposições verdadeiras. À conceitos e conhecimentos que o cérebro já havia assimilado estes podem se reorganizar tornando a redes de conhecimento do cérebro mais robusta.

2.3 Engenharia de software

Hoje em dia softwares estão presente em todos os lugares, tudo que a gente 'toca'. Softwares não se desgastam (físicamente) como hardware, mas estão sujeitos a modificações durante o seu ciclo de vida (FILHO, 2015). As modificações as vezes podem causar efeitos acidentais e/ou não esperados. Um software precisa de modificações conforme o tempo passa e para isso acontecer com mais tranquilidade exige a necessidade de uma documentação. Quanto melhor a documentação, mais fácil para entender. Para manter um bom projeto de software é necessário ter ou criar uma cultura de engenharia de software para adotar as melhores práticas. As quais que compreendem os pilares de custo, tempo de desenvolvimento e qualidade de software (FILHO, 2015).

No glossário de terminologia de Engenharia de Software da IEEE Std 610.12-1990, define-se Engenharia de Software como a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de um software; isto é aplicação de engenharia de software. Engenharia de software também pode ser o estudo das abordagens (IEEE, 1990).

Como softwares precisam de manutenção corretiva e/ou evolutiva, também está sujeito a inserção de defeitos decorrentes do desenvolvimento. Estes defeitos podem ser vistos e consertados antes da entrega (FILHO, 2015) ou ser descoberto pelo usuário que está utilizando e não espera se deparar com o erro. Por isso além de uma documentação, software também precisa de testes, estes que quando bem feitos asseguram a qualidade e confiabilidade do produto de software.

Engenharia de software está presente e tem descrições de melhores práticas em todas as fases desde a concepção, elaboração, construção e transição de um projeto. Seja em um projeto com metologia tradicional ou ágil, um projeto passa por essas fases. A engenharia de software "tem como objetivo apoiar o desenvolvimento profissional de software, cobrindo todos os aspectos da produção de um software." (MONSALVE; WERNECK; CESAR, 2010 apud SOUZA; FRANÇA, 2016)

[(SOUZA; FRANÇA, 2016)] (BENITTI; MOLLÉRI, 2008) concordam que a engenharia de software é uma área muito jovem e sofre contínuas mudanças nos seus fundamentos tecnológicos concretizadas nos métodos e ferramentas de suporte, portanto necessita de métodos de ensino lúdicos e dinâmicos que possam contribuir na aprendizagem do estudante.

Engenharia de Software percorre o levantamento de requisitos de um jogo, o planejamento e desenvolvimento das funcionalidades, testes para garantir que está tudo funcionando como o ocorrido e o empacotamento e a entrega para a finalização. Após cumprir este ciclo, para um software continuar 'vivo' é necessário que com o tempo ele receba manutenção e melhorias e também se desejar, evoluções. Além de agregar valor para o cliente ou desenvolver o acordado, também é necessário gerenciar sua infra-estrutura, definir onde hospedar as aplicações, os custos decorrentes de utilizar serviços ou hardwares de terceiros, realizar as configurações para a padronização e fazer o controle de mudanças e gestão da qualidade.

2.3.1 Levantamento de requisitos de software

É uma das fases mais importantes. A maioria dos projetos falha pelo levantamento de requisitos incorretos. O grau de insatisfação do cliente também está relacionado à levantamento de requisitos incorretos. É importante a consolidação dos requisitos para a equipe inteira ter uma visão alinhada. A consolidação dos requisitos serve como guia para descrever as funcionalidades a serem desenvolvidas e para verificação de que o software foi construído corretamente de acordo com o esperado. Permite uma organização e maior controle do que já foi feito e está pendente para ser feito.

2.3.2 Digital Game Based Learning LGBL

Aplica-se em situações onde o software a ser desenvolvido é simples, os requisitos são bem conhecidos tecnologia acessível e recursos para o desenvolvimento estão disponíveis.

A metodologia Digital Game Based Learning (DGBL) é um tipo das que prezam por utilizar jogos digitais em prol do aprendizado, para isso tem que ser medido a usabilidade do software, e os resultados do jogo tem que ser efetivos. A metodologia DGBL

que continua evoluindo seu método instrucional tem a necessidade de evidências empíricas para validar seu valor educacional e mostrar como o valor poderia ser aplicado mais efetivamente (KATMADA; MAVRIDIS; TSIATSOS, 2014).

Os jogos tem que ser testados em como estão sendo usado nas pessoas, pois ele podem não estar sendo benéficos. O autor cita que não há provas de que os jogos digitais pode sem benéficos para todas os tipos de atividades. Com o avanço da tecnologia, os jogos complexos requerem recursos de hardware mais potente e mais tempo para jogar, o que talvez pode indicar que não deve ser usado em sala de aula, pois não substitui o repasse do conhecimento (KATMADA; MAVRIDIS; TSIATSOS, 2014).

2.3.3 Características dos Jogos Educacionais

Primeiramente o jogo deve motivar o jogador. Características de jogos que atraem os jogadores, desafios, (KATMADA; MAVRIDIS; TSIATSOS, 2014) diz alguns fatores motivacionais que jogos atrativos devem ter: objetivos claros ao jogador, resultados incertos, fornecer feedbacks e o aumento gradual de dificuldade.

Outras recomendações para os jogos educacionais é que devem estar aderidos a grandes programas educacionais e deve incorporar grandes elementos para ajudar os estudantes a construir novas estruturas de conhecimento ou completar as existentes (KATMADA; MAVRIDIS; TSIATSOS, 2014).

2.3.4 Plataformas mobile

Existem diversas maneiras de se construir APP para mobiles. Algumas apenas para celulares Android, outras para sistemas iOS e outras para ambas plataformas. Existem *frameworks* de desenvolvimento onde o código gerado já é nativo da própria plataforma alvo (Android ou iOS) e outras onde o código é transformado em partes para a plataforma nativa. Existe um projeto chamado *kivy*, onde é escrito código na linguagem Python e o *kivy* com suas bibliotecas converte o código para gerar aplicações para Android e iOS.

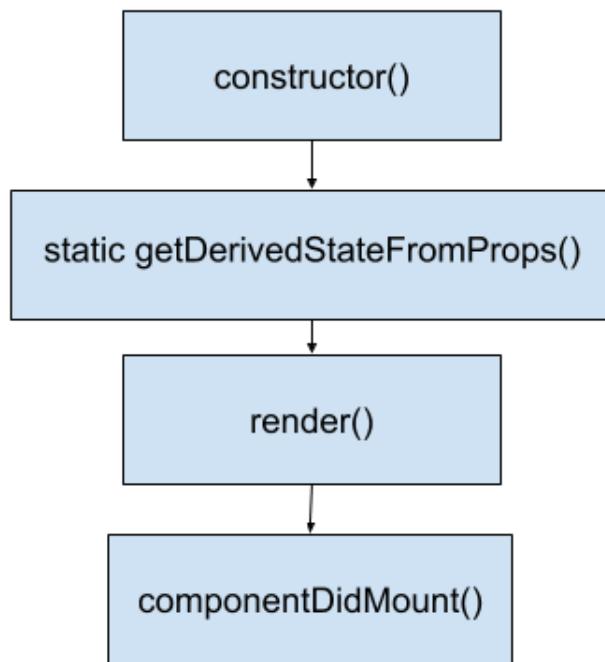
Outros *frameworks* analisados que também geram código para ambas plataformas de celular são o *Godot* e *React Native*. Por maior familiaridade do autor com o *HTML*, *CSS* e *JavaScript* optou-se por utilizar o framework *React Native*.

2.3.4.1 React Native

Aqui será explicado sucintamente o funcionamento do *React Native*. O Framework trabalha com componentes e possibilita a composição e agregação de componentes. Para isso é necessário que as classes criadas extendam (herança) a classe *React.Component*. Ao realizar a herança os componentes passam a ser obrigados a implementar o método *render()* e também herdaram dois tipos de dados: *props* e *states*. O método *render()* faz parte

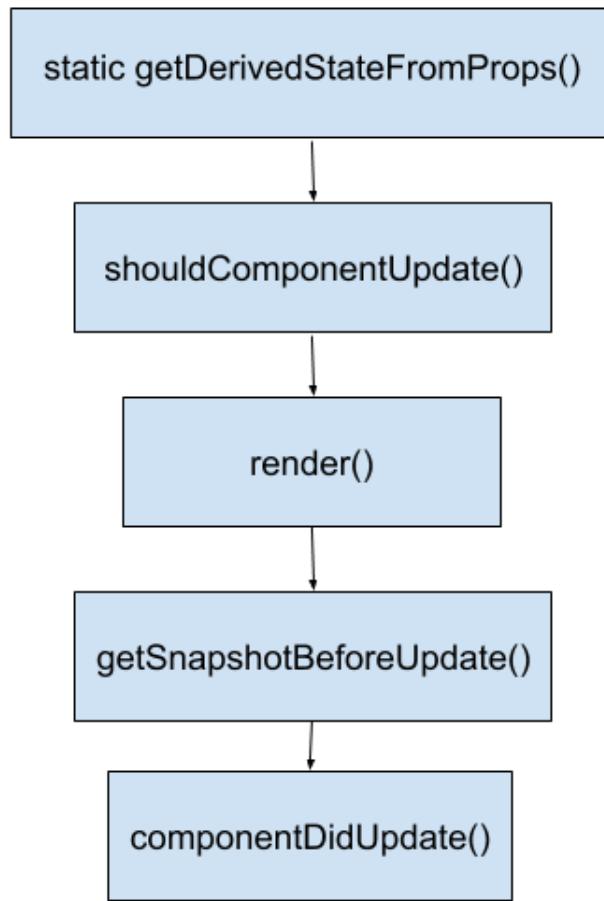
do ciclo de vida da montagem e atualização de um componente é o que irá renderizar o objeto na tela mostrando a sua estilização. As props são características estáticas passadas pela classe pai que um objeto recebe ao ser criado e o state são propriedades que podem ser alteradas pelo próprio componente. Componentes extendidos da classe *React.Native* passam a ter o ciclo de vida para montagem e atualização conforme descrito nas Figuras 1 e 2.

Figura 1 – Processo de montagem do componente React



Fonte: do próprio autor

Figura 2 – Processo de atualização do componente React



Fonte: do próprio autor

3 Metodologia

A metodologia utilizada foi a de relato de experiência e é de caráter quantitativo e qualitativo. Diz-se qualitativa por se preocupar com as percepções das pessoas em relação ao uso do jogo AprEnDO. Diz-se quantitativa por analisar os dados enviados dos alunos para o servidor de coleta.

Foi levantado um referencial teórico a respeito da contribuição e o sucesso de jogos no ensino, conceitos da Engenharia de Software que podem ser levados em conta de modo a auxiliar o desenvolvimento de software e a utilização de mapas conceituais como ferramenta para verificar a aderência de conhecimento nos alunos.

O relato de experiência é sobre a aplicação do jogo aprEnDO em uma turma de Engenharias da UnB-FGA na disciplina de C2. Por questões éticas e em respeito aos discentes voluntários que participaram, os nomes dos alunos serão preservados. Os levantamentos ocorreram no primeiro semestre de 2019. O conteúdo da matéria ministrada aos alunos durou cerca de UM mês. Após o período de ensino da matéria, a turma utilizou a aplicação em torno de dois meses para o reforço do conteúdo e neste tempo foram enviando suas estatísticas para um servidor central. Os dados foram consolidados em tabelas para análise e também foram analisados os mapas conceituais produzidos pelos alunos. Foram relatados *bugs*, erros e falhas encontrados pelos alunos durante a fase de jogo.

O que espero encontrar no mapa conceitual confeccionado pelos alunos para verificar a absorção do conhecimento de EDOs são proposições verdadeiras da classificação quanto a

- Tipo: Ordinária ou Parcial.
- Ordem: 1, 2, 3 ou superior.
- Linearidade: Linear ou não linear.
- Homogeneidade: Homogênea ou não homogênea.
- Separável: Separável ou não separável.
- Exata: Exata ou não exata.

Pois os mapas conceituais são a maneira usada para evidenciar que houve aprendizado significativo, de acordo com o ([SUBJACENTE, 2011](#)).

4 Engenharia de Software

4.1 Requisitos de Software

Os requisitos de software foram levantados e utilizados na hora do desenvolvimento para alcançar os objetivos elencados para o jogo.

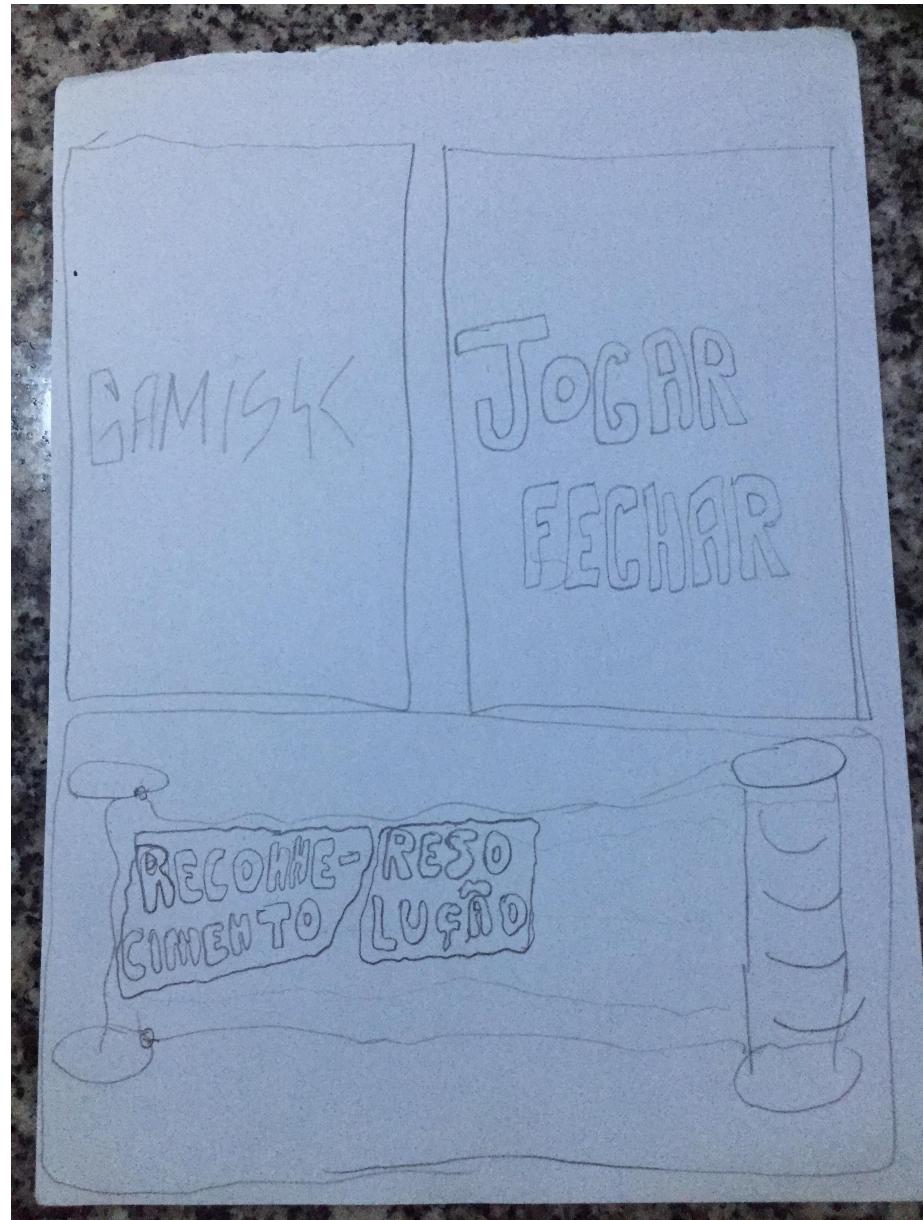
- Ter um módulo para classificação de EDO
 - Ter fase de classificação de ordem
 - Ter fase de classificação de tipo
 - Ter fase de classificação de linearidade
 - Ter fase de classificação de homogeneidade
 - Ter fase de classificação de separável
 - Ter fase de classificação de exata
 - Escolher 20 questões aleatórias do banco de equações
- Ter um módulo para resolução de EDO de 1^a ordem
 - Ter fase de resolução de EDO 1^a ordem homogênea
 - Ter fase de resolução de EDO 1^a ordem não homogênea
 - Ter fase de resolução de EDO 1^a ordem exata
 - Ter fase de resolução de EDO 1^a ordem não exata
 - Escolher 10 pares de equações aleatórias do banco de equações
- Estar disponível para download no google play
- Sinalizar ao usuário fases concluídas
- Permitir envio de bugs e erros
 - Notificar se o erro é por falta de internet
 - Notificar se o erro é por problema do servidor
- Permitir envio de sugestões e feedback
 - Notificar se o erro é por falta de internet
 - Notificar se o erro é por problema do servidor

- Ter um módulo de envio de estatísticas
 - Identificar o aluno pela matrícula
 - Enviar estatísticas de classificação
 - * armazenar fases concluídas
 - * armazenar quantidade de vezes que cada fase foi concluída
 - * armazenar 20 questões presentes na tentativa
 - * armazenar tempo total gasto em cada conclusão
 - Enviar estatísticas de resolução
 - * armazenar fases concluídas
 - * armazenar quantidade de vezes que cada fase foi concluída
 - * armazenar 10 questões presentes na tentativa
 - * armazenar tempo total gasto em cada conclusão
 - Sinalizar ao usuário quando a estatística for enviada
 - Sinalizar ao usuário quando a estatística não for enviada
 - * Notificar se o erro é por falta de internet
 - * Notificar se o erro é por problema do servidor

4.2 Prototipação

A prototipação não seria utilizada, até deparar-se com o problema de não saber como seria a imagem final das telas, então optou-se por utilizar a prototipagem de baixa fidelidade, onde o importante seria ter uma noção geral de como seriam as telas. Desenhou-se então no papel um exemplo de como imaginava-se que seria a tela final do jogo. O primeiro desenho foi o seguinte

Figura 3 – Prototipação das telas iniciais do jogo



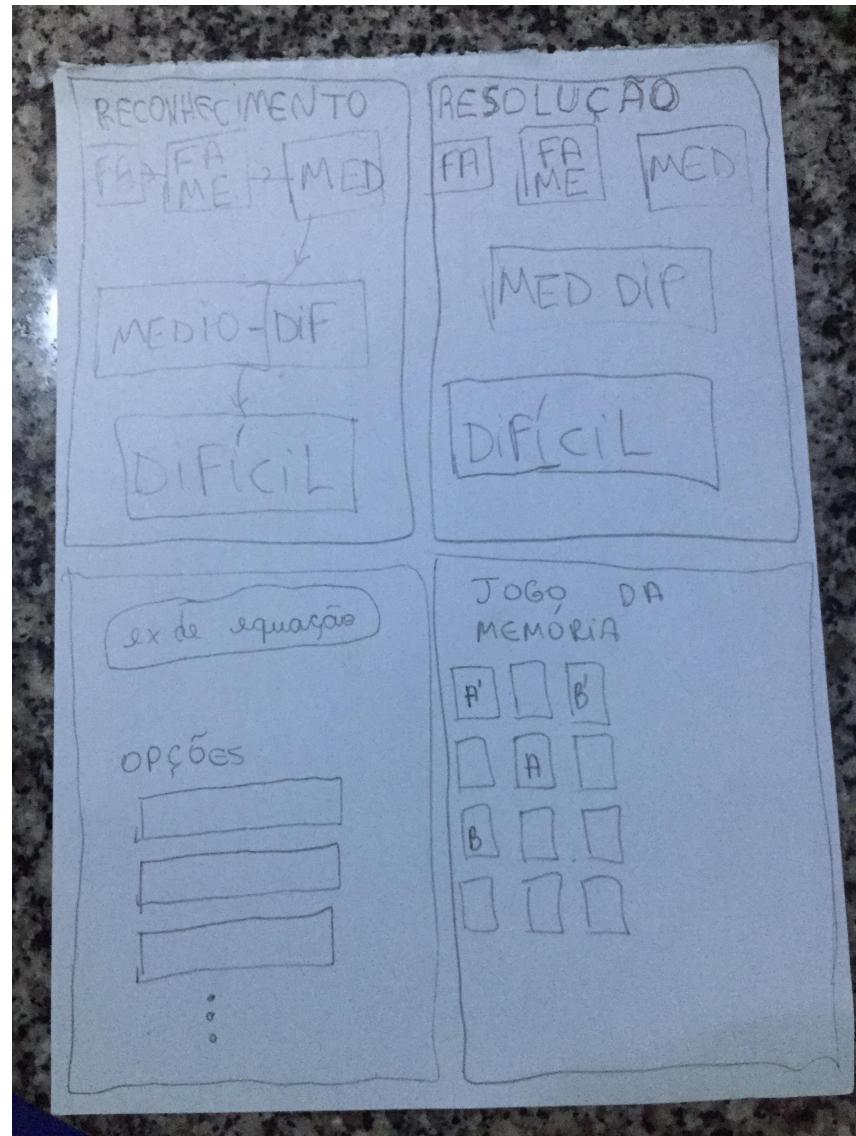
Fonte: do próprio autor

Na Figura 3 no primeiro retângulo em cima e à esquerda é possível ver escrito "Gamisk", que seria o primeiro nome do jogo e a primeira tela a ser apresentada ao clicar no ícone do APP. Ao lado do retângulo Gamisk à direita é possível ver escrito "Jogar" e "Fechar" que seria a segunda tela do jogo. Porém percebeu-se que não haveria necessidade das duas telas e então decidiu-se pular direto para a que apresenta "RECONHECIMENTO" E "RESOLUÇÃO". A necessidade da tela gamisk seria em caso do jogo demorar para carregar recursos, o que não acontece. A necessidade da segunda tela era para ter o botão fechar, para o jogador poder sair do jogo, porém os celulares android apresentam o botão "Home" para sair da aplicação, então viu-se desnecessário criar a funcionalidade de sair do jogo. Então ao abrir o APP o usuário já será redirecionado para escolher o módulo de

jogo que deseja jogar.

A Figura 4 retrata como seria o jogo enquanto pensava-se que existiria os níveis de dificuldades fácil, médio e difícil. Depois houve o mapeamento dos níveis de dificuldade para o indicado na Figura 6. A mudança foi então que ao invés das fases terem nome 'fácil', 'médio' e 'difícil', elas passaram a ter o nome de 'ordem', 'homogeneidade', 'tipo' e 'linearidade' e foram ordenadas de acordo com o que acreditava-se ser mais fácil para classificar até o mais difícil.

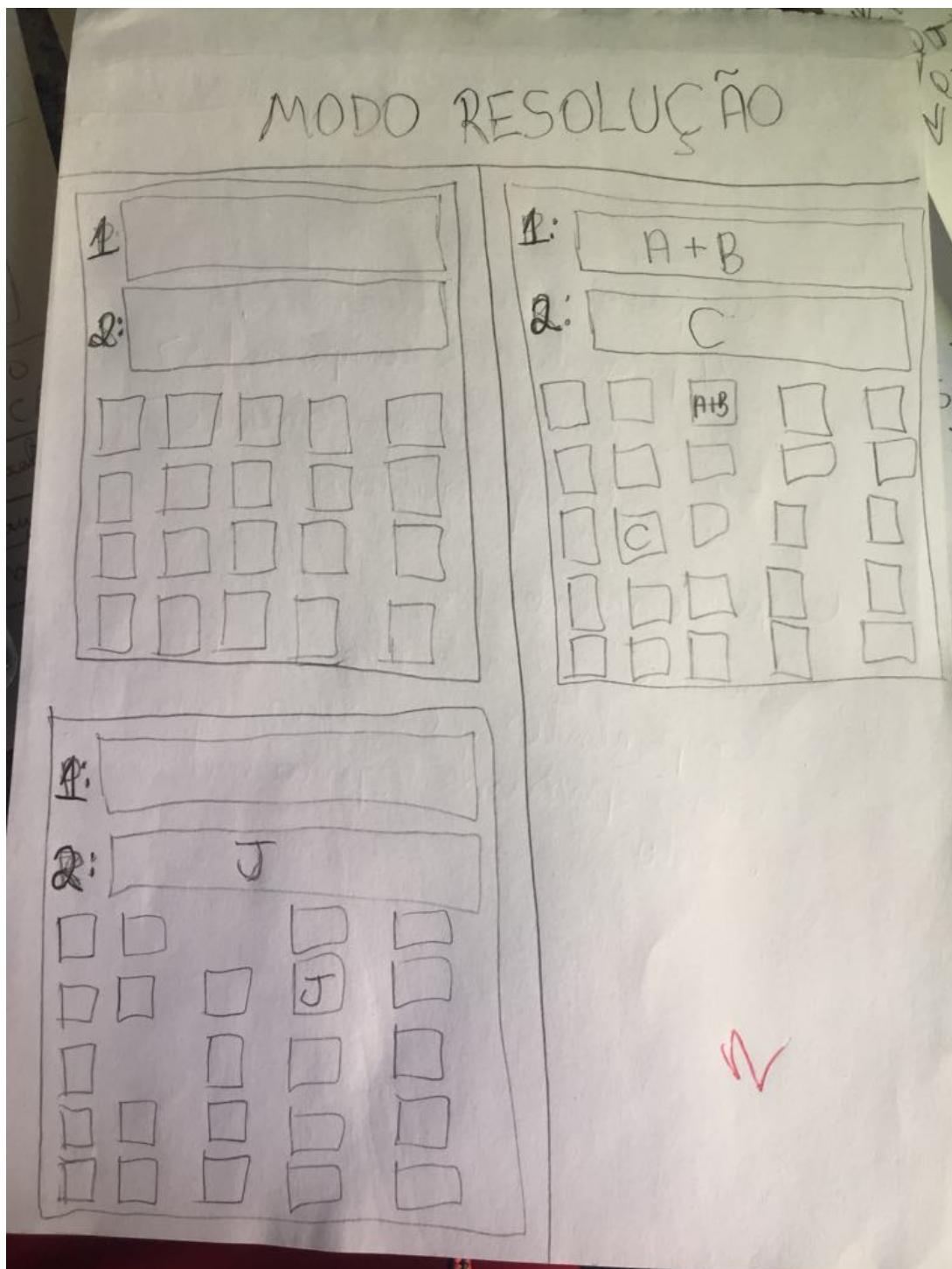
Figura 4 – Prototipação das telas dos módulos



Fonte: do próprio autor

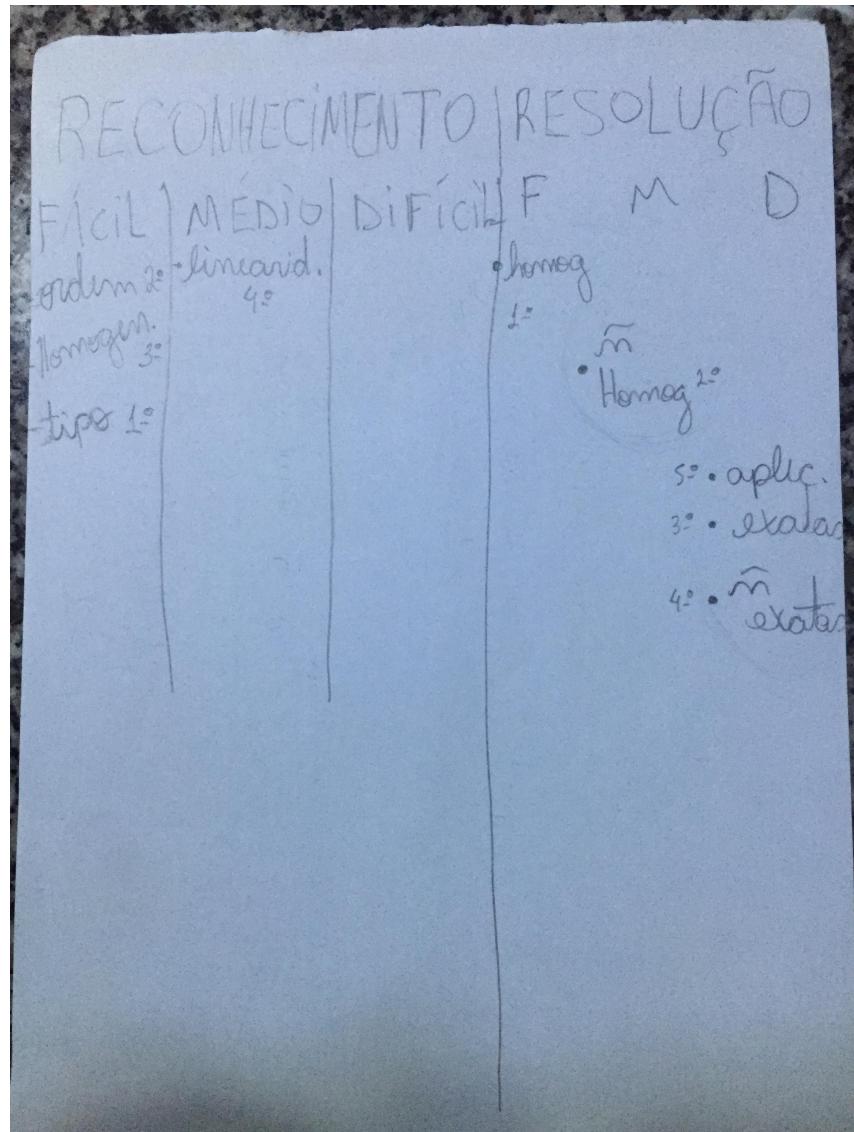
A imagem do jogo da memória acima onde tem o par A , A' e B , B' teve de ser redimensionada pelo tamanho das imagens não ficar legível apenas no desenho das cartas. Para isso a criação de duas telas para mostrar as duas equações *clickadas* com maior clareza. É possível ver a nova tela na Figura 5

Figura 5 – Nova tela de resolução



Fonte: do próprio autor

Figura 6 – Prototipação das telas dos módulos



Fonte: do próprio autor

4.3 Banco de equações

Antes de chegar ao modelo do projeto como está hoje, foi pensado em um modelo anterior. Após algumas reflexões optou-se seguir por outro caminho considerado mais simples.

4.3.1 Como se pensou

No início pensou-se em utilizar um banco de dados relacional para armazenar os metadados e as imagens renderizadas das equações. No Modelo Entidade Relacionamento (MER) modelado inicialmente estavam presentes seis entidades, cada uma com seus atributos.

- EQUAÇÃO_DIFERENCIAL (linearidade, separável, homogênea, exata e o `id`).
- ORDEM (primeira, segunda, terceira, ordem_superior).
- DIFICULDADE (fácil, facilmedio, medio, mediodifícil e difícil).
- TIPO (ordinária e parcial).
- PERGUNTA (img_src, largura e comprimento).
- RESPOSTA (img_src, largura e comprimento).

Houve a reflexão de criar uma entidade pai IMAGEM e fazer com que PERGUNTA e RESPOSTA herdem suas propriedades. Este modelo apesar de incompleto, seria viável para a solução, mas optou-se por não seguir utilizando um banco de dados e *queries*.

4.3.2 Como está hoje

O projeto banco de equações está disponível em <<https://github.com/LeonardoRk/TCC-2>> dentro da pasta chamada **banco**. Este projeto visa preparar e validar informações e imagens de perguntas e respostas das equações diferenciais que serão utilizadas no jogo.

O AprEnDO foi populado com as imagens e os metadados fornecidos pelo Wolfran Alpha através de um script de população. O Wolfran Alpha é um serviço online que tem como uma de suas muitas funcionalidades resolver equações diferenciais e apresentar informações quanto à classificação. Com o intuito de realizar requisições à Application Programming Interface API foi criada uma conta gratuita para liberar acesso a uma chave gratuita (chave usada: 3GGQAT-98EG4KV6VL) que permite 2000 requisições por mês. Para se conectar com o Wolfran Alpha foi realizado o *download* de uma biblioteca em javascript que pode ser encontrada em: <<https://products.wolframalpha.com/api/libraries/javascript/>>

Foi desenvolvido um script alimentado por um arquivo de *seeds* que realiza as requisições para a API do Wolfran Alpha para ler os metadados de cada EDO's e guardar/baixar os dados necessários. Os metadados são no formato *JSON*. Eles fornecem a url para a imagem .gif das equações perguntas e respostas (quando existente), estas são utilizadas no jogo, então é necessário mapeá-las em um arquivo index.js para adicionar na pasta resource do jogo no *react native* para fazer com que a internet não seja um requisito para jogar.

Na pasta banco existe um arquivo chamado **seeds.txt** que é o arquivo com as equações diferenciais para alimentar o banco de dados da aplicação aprEnDO. O script **equações.js** lê o arquivo de seeds equação por equação, faz a requisição para o Wolfran

Alpha utilizando os códigos da pasta wolfran_api e requisita todos os pod disponíveis (pod são os arrays de informação disponibilizados). Após ter os pods são filtradas as informações desejadas e salvas em arquivos de informações localizadas na pasta **info**. Os pods apresentam as *urls* das imagens de equações de perguntas e respostas, quando existe resposta. Equações sem solução só podem ser utilizadas no primeiro módulo do jogo, o de classificação e não são incluídas no módulo de resolução. As imagens de perguntas baixadas são salvas na pasta **pergunta** e as imagens de respostas salvas na pasta **resposta**.

Com todas as informações desejadas de cada equação é possível utilizar o arquivo **estatisticas.js** que lê todos os arquivos de informação para contabilizar as informações da quantidade de equações diferenciais, quais tem resposta e quais não, quais são homogêneas, exatas, separáveis, linear, não linear, ordem1, ordem2, ordem3, ordem de 4 para cima são consideradas ordem superior, além de fazer a contagem total,também são indicados o número da equação. Essas informações são escritas num arquivo de controle para que possa ser lido pelo aplicativo aprEnDO e fazer a seleção das equações correta para renderizar, a depender do nível que a pessoa está jogando. O nome do arquivo de controle é **DADOS_GERAIS.json**.

4.4 AprEnDO

A linguagem de programação utilizada é o nodejs com o framework react native para gerar aplicação em código nativo android. Para baixar os pacotes e fazer o controle dos mesmos está sendo utilizado o nvm e o yarn.

O ambiente de desenvolvimento usa um emulador para simular o jogo.

A pasta do jogo aprendo que encontra-se dentro da pasta desenvolvimento tem o projeto react-native com node.js e android, o package.json com as dependências, o index.js que é o ponto de entrada, os códigos fonte e os recursos. Também tem um arquivo makefile que por padrão abre o emulador de celular android, com permissão de superusuário inicia o servidor do react para desenvolvimento, cria no emulador a mais recente build do aplicativo e inicia o terminal para logs do celular emulador.

Os arquivos da pasta e uma breve explicação: O arquivo index.js faz a importação dos módulos que são também as telas do jogo localizadas na pasta screen do código fonte em src. Também importa o módulo de navegação entra telas que precisa ser configurado para estar ciente das screens que antecedem a atual, sem a referência ao voltar a tela o jogo fecha. Uma configuração default que foi setada era para desaparecer a barra superior de navegação das telas, porém percebeu-se que em alguns celulares a barra aparecia e um pouco desconfigurada, já em outros modelos comportava como o esperado e estava invisível.

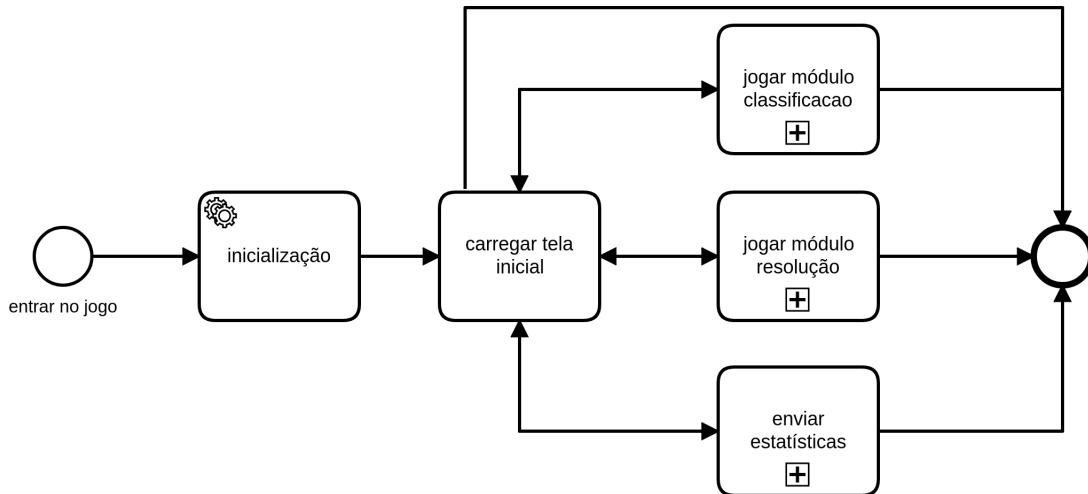
A pasta android tem o código gerado pelo react-native o qual é um projeto padrão com os arquivos necessário para android, código fonte em .java, o arquivo manifest, gradle e arquivos de configuração. Algumas dependências do jogo em react native foram adicionadas manualmente no código nativo do android.

A pasta node_modules apresenta os vários módulos do node instalados em *background*.

A pasta resource tem as imagens de todas as equações de perguntas usada no módulo de classificação, as respostas usadas no módulo de resolução e seus arquivos index.js que mapeia o número das equações com a importação das mesmas dentro do jogo, a importação só se da porquê todas as imagens já possuem o termo “require(‘numero da equação’)” que é pré-processada antes da escolha da equação aleatória a ser importada dentro da fase. Outro arquivo importante que tem nessa pasta é o **DADOS_GERAIS.json** que possui os metadados das equações, quanto à todas classificações, tipo, ordem, homogeneidade e etc.

Foram desenhados processos para entender o fluxo do jogo. O primeiro processo é uma visão geral.

Figura 7 – Processo geral do jogo aprEnDO

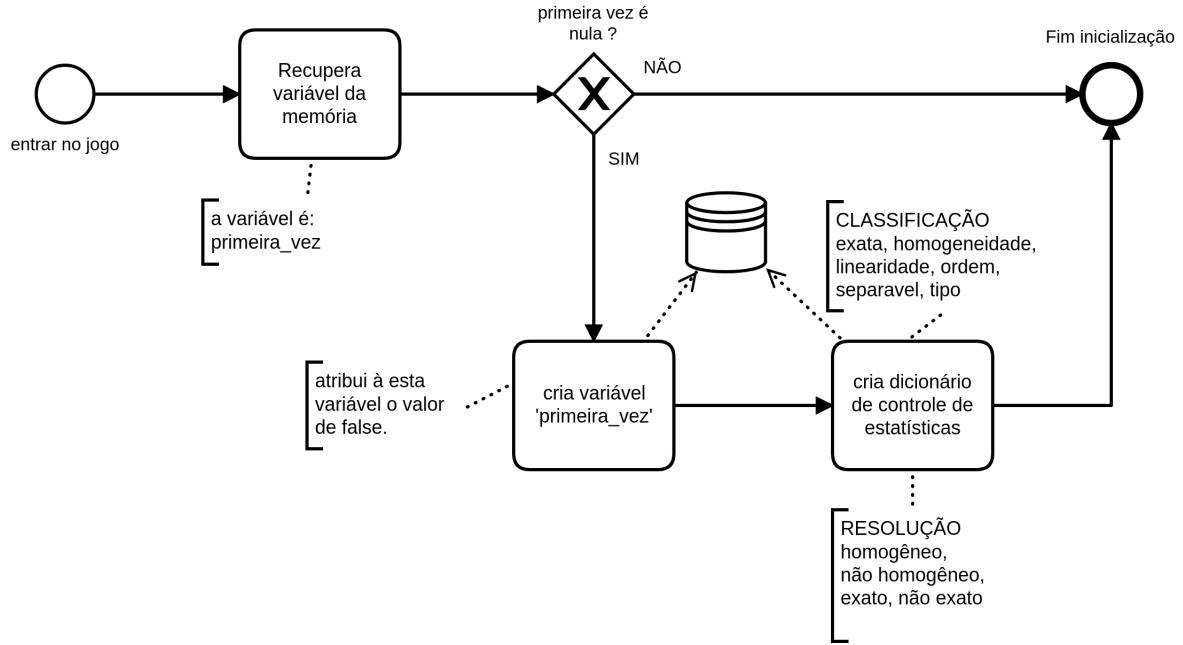


Fonte: do próprio autor

Na Figura 7 mostra-se que ao entrar no APP a primeira atividade é um script do sistema, assim que este termina é carregada a tela inicial que permite ao usuário jogar o módulo de classificação, jogar o módulo de resolução, enviar estatísticas ou simplesmente sair do jogo sem fazer nada.

A primeira macro-atividade é a inicialização, indicada na Figura 8.

Figura 8 – Processo de inicialização do jogo aprEnDO

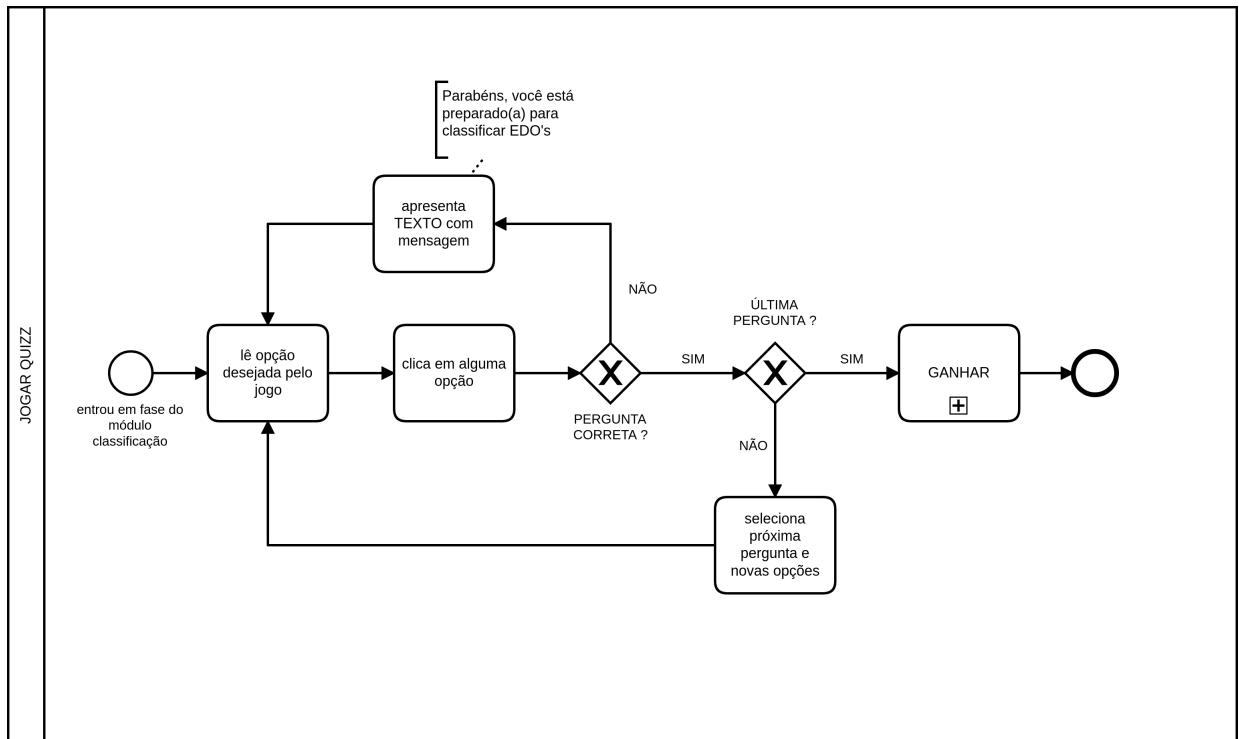


Fonte: do próprio autor

O processo 8 recupera a variável 'primeira_vez' da memória, caso a variável exista e tenha o valor de 'falso' significa que o usuário já jogou outras vezes, então o processo termina. Caso seja realmente a primeira vez do usuário é necessário criar a variável 'primeira_vez' para sinalizar que não é mais a primeira vez. Também é criado a estrutura de estatísticas para classificação e resolução e persistida no banco de dados do celular.

A segunda macroatividade é o processo de jogar o módulo de classificação, nele o usuário precisa escolher alguma fase de classificação. A Figura 9 mostra quais são as possíveis ações a serem tomadas pelo jogador.

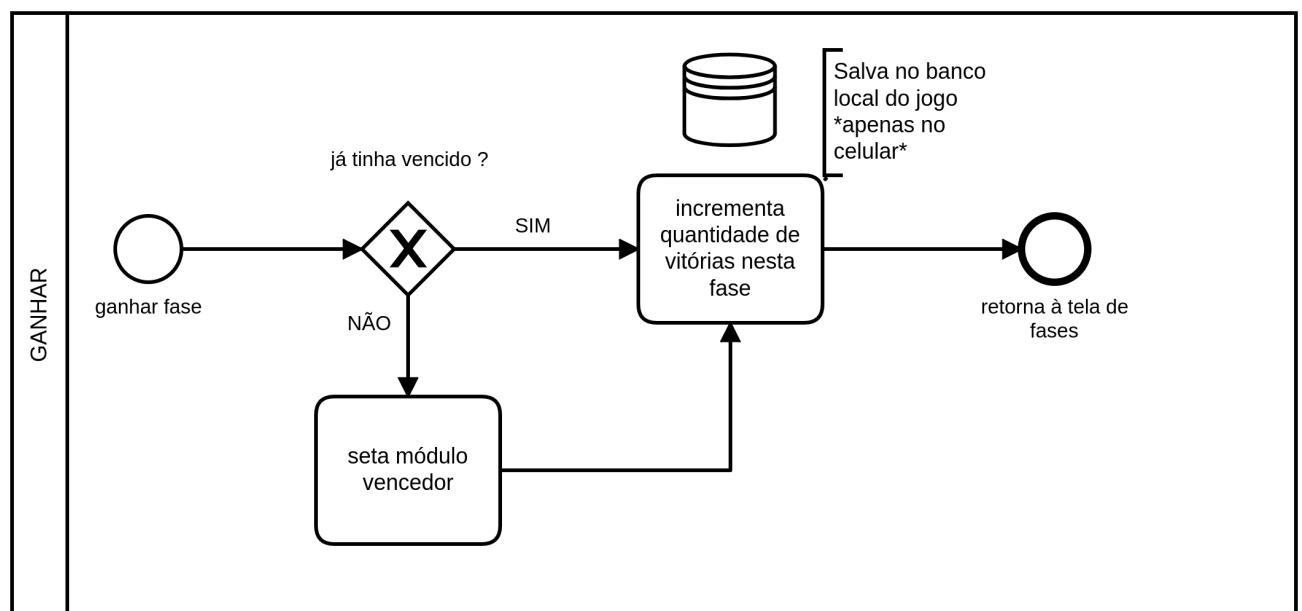
Figura 9 – Processo de jogo de classificação



Fonte: do próprio autor

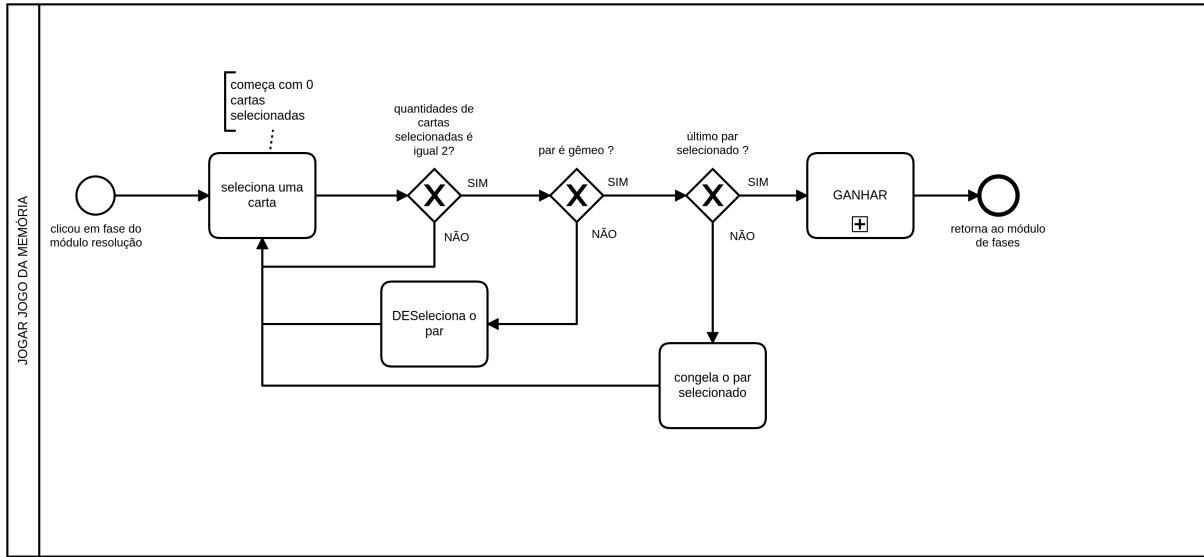
Ambos os módulos 9 e 11 podem ativar o processo da Figura 10 que consiste em armazenar as estatísticas de mais uma vitória nos dados do celular para que possam ser enviados ao servidor de coleta.

Figura 10 – Processo de ganhar uma fase



Fonte: do próprio autor

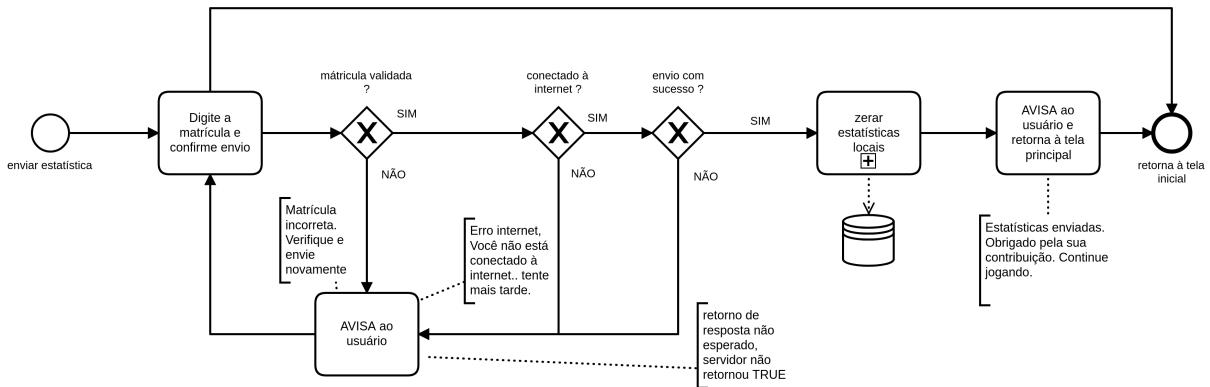
Figura 11 – Processo de jogo de resolução



Fonte: do próprio autor

A macro-atividade "GANHAR" presente nas Figuras 9 e 11 envolve o envio de estatísticas de jogo para um servidor de análise e coleta. O processo é detalhado na Figura 29.

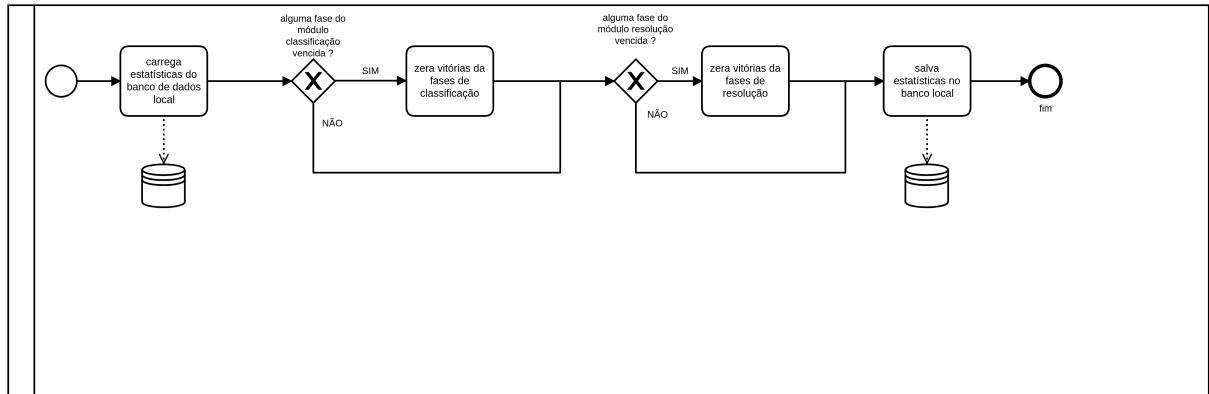
Figura 12 – Envio de estatísticas



Fonte: do próprio autor

É necessário que o jogador digite sua matrícula corretamente, pois esta será validada com a quantidade de 8 algarismos. Caso a matrícula seja validada e haja a conexão com a internet ocorrerá o envio dos dados, qualquer erro pelo caminho haverá o aviso específico ao jogador. Após a persistência dos dados no servidor, os dados locais são zerados para que não sejam redundantes no próximo envio. O processo de zerar estatísticas é apresentado na Figura 13.

Figura 13 – Zerar estatísticas após envio



Fonte: do próprio autor

Zerar estatísticas percorre as fases de classificação e resolução verificando qual fase já foi completada ao menos uma vez e trocando o valor atual para 0. Após o loop nos módulos os dados são persistidos na memória do celular.

Outro artefato de documentação desenvolvido foi o diagrama de classes. Este foi modelado de modo a facilitar o desenvolvimento e o entendimento da pasta src, pois deu uma guia no que precisava ser feito e como as classes do jogo e os componentes se relacionam.

Figura 14 – Diagrama de Pacote do AprEnDO

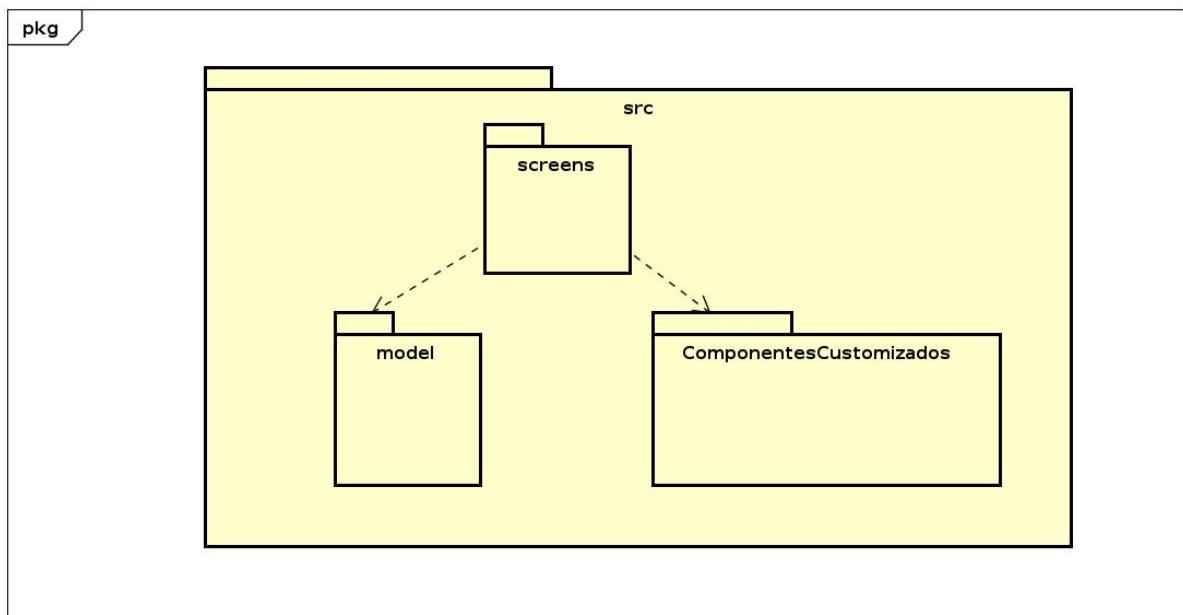
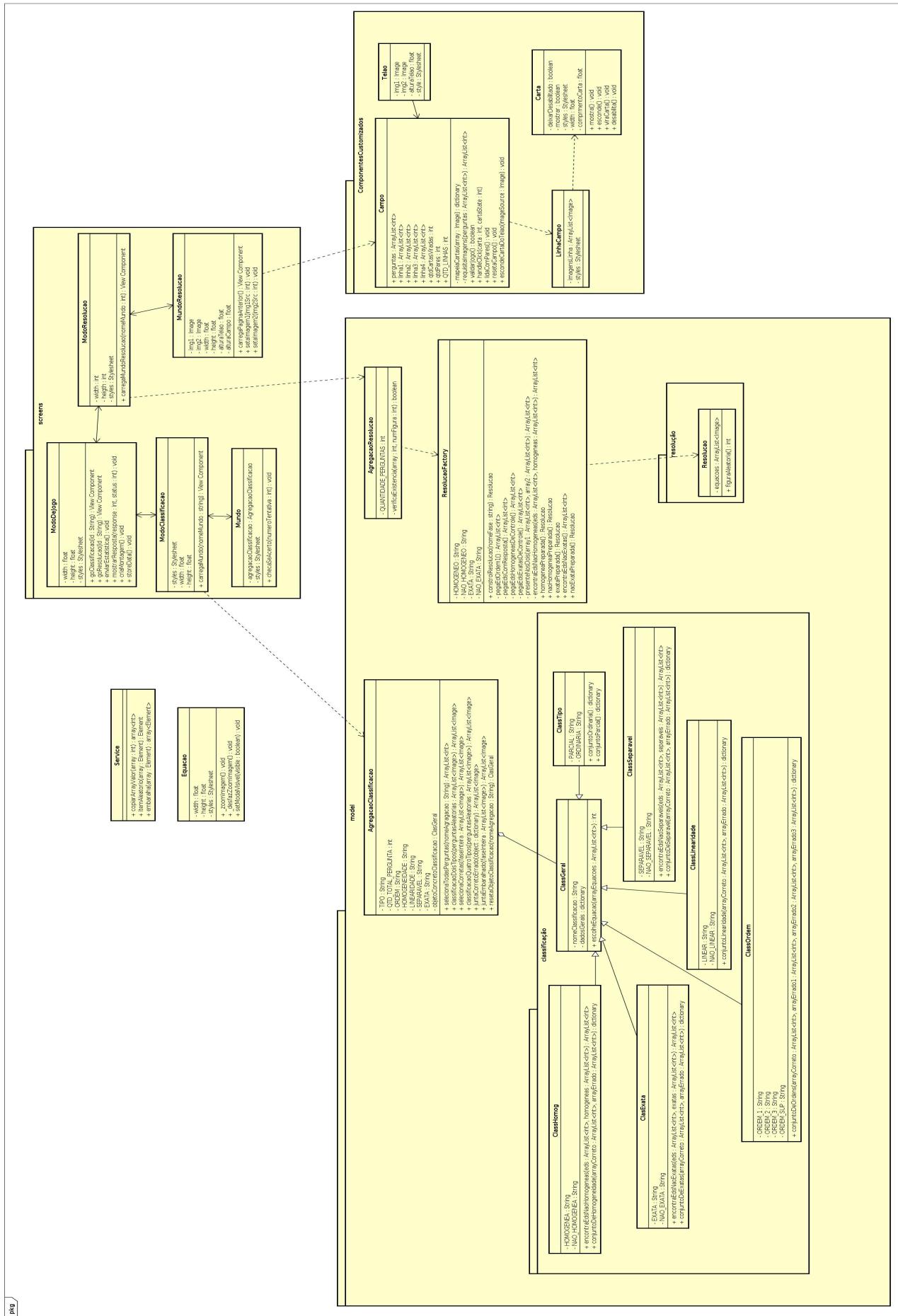


Figura 15 – Diagrama de Classe do AprEnDO



A imagem pode ser vista maior em <<https://github.com/LeonardoRk/TCC-2/wiki/Diagramas#diagrama-de-classe-do-aprendo>>

A pasta src apresenta o código fonte escrito em React (node.js). A modelagem da aplicação é possível visualizar no diagrama de classe. Tem-se a pasta dos componentes customizados, a pasta model (que tem o modelo de perguntas e respostas e classes construtoras da agregação de perguntas e respostas para o jogo), a pasta screens que contém as telas de jogo, sendo que algumas screens contém componentes que estão na pasta de componentes customizados. Tem o arquivo Service.js que realiza algumas tarefas auxiliares necessárias em partes misturadas do jogo, como por exemplo embaralhar um array (conjunto de dados homogêneos), realizar uma cópia de valores de um array e escolher um item aleatório.

A seguir uma explicação das funções desenvolvidas que estão na pasta SRC e suas subpastas.

PASTA: SRC

- ARQUIVO:Service

- copiarArrayValor: recebe um vetor como parâmetro e copia apenas os seus valores sem manter a referência. Para isso criar um vetor vazio, percorre o recebido e escreve seus valores no novo vetor que será retornado.
- itemAleatorio: recebe um vetor como parâmetro. Em caso de não pelo menos um elemento, nulo é retornado. Caso contrário com o apoio da biblioteca Math de matemática, um número flutuante entre 0 e 1 é escolhido e multiplicado pelo comprimento do vetor. O número é arredondado para baixo e então o item nesta posição do vetor é retornado.
- embaralha: recebe um vetor e aleatoriamente troca as posições dos valores

- ARQUIVO:Equacao

- _zoomImagem: função ativada quando houver um longo clique em cima de algum exemplo de equação. Esta habilita o aparecimento de uma modal para mostrar equação ampliada.
- _desfazZoomImagem: função ativada quando o dedo é tirado da tela após um longo clique em cima de algum exemplo de equação. Esta desabilita o aparecimento de uma modal que mostra equação ampliada.
- setModalVisivel: função chamada por _zoomImagem e _desfazZoomImagem para habilitar ou desabilitar o aparacimento da modal com equação ampliada.

PASTA:SRC/screens

- ARQUIVO:ModoDeJogo

- goClassificacao: a função goClassificacao utiliza a dependência Navigation e serve para redirecionar para a tela de classificação das equações diferenciais. Recebe como parâmetro um ID que é um identificador único do tipo string referente ao nome da screen que é para redirecionar.
- goResolucao: a função goResolucao utiliza a dependência Navigation importada no projeto que serve para redirecionar para a tela de resolução das equações diferenciais. Recebe como parâmetro um ID que é um identificador único do tipo string referente ao nome da screen que é para redirecionar.
- enviarEstatísticas: a função enviarEstatisticas funciona de modo assíncrono e utiliza da dependência netInfo para checar seu dispositivo está conectado à internet. Caso o dispositivo não esteja, um alerta avisando ao usuário que não está conectado à Internet é acionado e retorna a página inicial da aplicação, caso haja conexão com a internet é checado se a matrícula digitada pelo usuário não é nula e ao remover os caracteres de barra de espaço verifica-se que não está vazio e que contém 8 dígitos. Em caso de matrícula inserida corretamente é acionado um loader enquanto as estatísticas locais do celular são capturados e formatadas para então realizar uma requisição ao servidor do Heroku na rota '/estatísticas'. O conteúdo JSON é formatado para string e então enviado. Após o processamento no servidor uma resposta é retornada para o aplicativo e a função enviarEstatisticas chama a função mostraResposta que tratará de mostrar a resposta adequada a depender do status de retorno do Servidor. Em caso de alguma exceção ocorrer durante todo o processamento um alerta é mostrado ao usuário indicando que houve erro com o servidor e pedindo para retornar mais tarde. O loader é desabilitado e novamente retorna-se a tela inicial da aplicação.
- mostraResposta: a função mostraResposta é acionada pela função enviarEstatisticas e também assíncrona. Recebe como parâmetro a resposta do Servidor e o código de status. Caso a resposta tenha sucesso NÃO OKAY é mostrado um alerta para o usuário informando retorno não esperado e retorna-se à tela inicial. Em caso de resposta sucesso OKAY é carregado da memória local as estatísticas atuais e um loop percorre todas as fases do modo de classificação e de resolução zerando a quantidade de vezes que o jogador terminou de jogar, para que no próximo envio seja enviado apenas as variações entre o último envio e o atual. Após as estatísticas estarem zeradas, sstas são salvas na memória do celular e um alerta de estatísticas enviadas com sucesso é apresentado ao usuário, só então retorna-se a tela inicial.

- criaMontagem: a função é acionada apenas na primeira vez o que o jogador acessa o jogo para então criar o template das estatísticas de classificação e resolução e todas as suas fases e assim é salvo na memória do celular. O tipo de dado das estatísticas é um Json.
- storeData: a função storeData é acionada toda vez que um jogador entra no jogo. É tentado recuperar da memória uma variável chamada ??, caso esta variável seja nula significa que é a primeira vez do usuário no jogo, esta variável é então criada e salva na memória com valor falso para nas outras vezes não cair neste caso, chama-se a função ?? para gerar o template das estatísticas. Caso a variável ?? já exista e retorne o valor de falso, significa que não é a primeira vez do usuário no jogo e nada acontece.

- ARQUIVO:ModoClassificacao

- carregaMundo: função assíncrona que tem como parâmetro o nome de mundo que será criado para ser repassado ao objeto agregação classificação. Este objeto serve para a função na hora de chamar a nova tela de mundo passando as propriedades: vetor de perguntas aleatórias criadas, vetor do número das imagens corretas, vetor das fases embaralhadas, o total de questões e o nome do mundo.

- ARQUIVO:ModoResolucao

- carregaMundoResolucao: função assíncrona que recebe como parâmetro o nome do mundo para criar objeto agregaçãoResolução e passar para próxima screen chamada ?? junto com a propriedade ?? que é um vetor provido pelo objeto agregaçãoResolução

- ARQUIVO:Mundo

- checaSeAcerto: função assíncrona é ativada assim que alguma opção de classificação é clickada. Recebe como parâmetro o número da tentativa da rodada, que varia de 1 a 4. Caso o número da tentativa clicada seja igual ao da pergunta correta o número da perguntaAtual é incrementado e o número da quantidade de perguntasCorretas também é incrementado apenas em caso de não ser a última pergunta. Em caso de ser a última pergunta e tentativa estiver correta as estatísticas são atualizadas com o incremento de uma vitória na fase jogada do módulo de classificação. Ao vencer a fase um alerta de parabenização é mostrado e abre-se uma janela que ao ser confirmada retorna à tela de classificação. No caso de tentativa incorreta é mostrado um alerta de tente novamente e retorna-se à questão pendente.

- ARQUIVO:MundoResolucao

- carregaPaginaAnterior: utiliza a dependência Navigation e apenas fecha a página atual que a página anterior já está na pilha e passa então a ser a atual.

PASTA:SRC/model

- ARQUIVO:AgregacaoClassificacao

- selecionaTodasPerguntas: este método recebe o parâmetro do nome da fase de classificação e retorna um array com a escolha aleatória de vinte perguntas da fase que será jogada.
 - classificacaoDoisTipos: esta função serve para todas as fases de classificação que tem apenas duas perguntas (tipo, homogeneidade, linearidade, separável e exata). Não serve apenas para "ordem". A função recebe o array de vinte perguntas aleatórias e para cada pergunta monta um dicionário com uma equação correta e três equações erradas. Esse conjunto é adicionado a um array. É retornado o array de dicionário com vinte conjuntos.
 - selecionaCorretas: Recebe como parâmetro o array de dicionários com os vinte conjuntos. Monta-se um array com o número das vinte equações corretas e este array é retornado.
 - classificacaoQuatroTipos: esta função serve para todas as fases de classificação que tem apenas quatro perguntas, que no caso é apenas para "ordem". A função recebe o array de vinte perguntas aleatórias e para cada pergunta monta um dicionário com uma equação correta e três equações erradas. Esse conjunto é adicionado a um array. É retornado o array de dicionário com vinte conjuntos.
 - juntaCorretoErrado: recebe um dicionário como parâmetro que contém as chaves "correto" e "errado". O correto tem apenas um número de equação e o errado tem três números de equação. Todos os número de equação do dicionário são adicionados a um único array e este é retornado.
 - juntaEmbaralhado: recebe como parâmetro o array de dicionários e embaralha as quatro questões de cada fase, de modo que a equação correta não fique sempre na mesma posição. É retornado o array com todas as equações da fase embaralhado.
 - resetaObjetoClassificacao: recebe como parâmetro um nome da fase e retorna um objeto especializado do norme da fase. Pode ser tipo, ordem, homogeneo, linearidade, separável ou exata.

- ARQUIVO:AgregacaoResolucao

- verificaExistencia: recebe como parâmetro um array com número de figuras e o número de uma figura. Retorna verdadeiro caso o número da equação esteja contido no array e retorna falso caso o número da equação não esteja no array.
- ARQUIVO:ResolucaoFactory
 - constroiResolucao: este método recebe como parâmetro o nome da fase para retornar o objeto com as equações possíveis do arquivo de controle
 - pegaEdOrdem1:
 - pegaEdsComResposta:
 - pegaEdsHomogeneasDeControle:
 - pegaEdsExatasDeControle:
 - presenteNosDois:
 - encontraEdsNaoHomogeneas:
 - homogeneaPreparada:
 - naoHomogeneaPreparada:
 - exataPreparada:
 - encontraEdsNaoExatas:
 - naoExataPreparada:

PASTA:SRC/model/classificação

PASTA:SRC/model/resolução

PASTA:SRC/ComponentesCustomizados

- ARQUIVO:Campo
 - mapeiaCartas: recebe o array retornado da requisitaImagens o qual inicia-se com uma pergunta e sucessivamente sua solução. Um loop é feito no array para criar um dicionário com chaves as perguntas e valor as respostas. Por isso o loop é iterado apenas nas perguntas, que são as posições pares. Ao final da iteração e população do dicionário, o mesmo é retornado.
 - requisitaImagens: recebe um array com o número das figuras aleatórias sorteadas para esta partida e faz o import através da função require procurando no arquivo index das perguntas e das respostas. O retorno é um array com vinte figuras, sendo dez perguntas e dez soluções das perguntas. OBS: o array de retorno é ordenado com uma pergunta e a sua solução sucessivamente.

– validarJogo: é chamada sempre que duas cartas estão viradas para cima. A função serve para decidir se duas cartas formam um par. A verificação ocorre comparando com o array de cartas mapeado chave e valor. O procedimento checa se a primeira carta virada é chave ou valor. Se for chave, então compara a segunda carta com o valor, se for valor, procura-se então a chave para comparar com a segunda carta para assim confirmar se a dupla forma um par gêmeo ou não. Em caso de ser um par, pela regra do jogo é decrementado uma unidade dos pares restantes, se for o último par já acontece a atualização das estatísticas de qual fase do módulo resolução foi vencido e a parabenização ao usuário com retorno para a tela de escolha de fase. Se não for o último par, a função apenas retorna informando se as cartas atuais formam um par ou não.

– handleClick: é acionada quando ocorre o click em qualquer carta enviando a informação se a carta é para ser virada ou escondida (true ou false). Se for um click para mostrar a carta, então a variável quantidadeDeCartasViradas é incrementado em 1. OBS: Se esta for a única carta virada então na posição reservada para a imagem 1 será renderizada a equação da carta OBS1: Se for a segunda carta virada, a carta será desenhada no espaço reservado para a imagem 2 e será aplicado o procedimento do jogo da memória para validar se as 2 cartas foram um par.

Se for um click para não mostrar a carta é acionada a função para esconder a carta do telão (espaço reservado para renderização das equações), e se o valor da variável não cair em nenhum dos casos previstos um novo erro é lançado.

– lidaComPares: é chamada pelo handleClick deste mesmo arquivo. Recebe como parâmetro um booleano que representa se as duas cartas viradas para cima formam um par ou não. Em caso de serem um par, as cartas tem a função de click desabilitadas, e são travadas viradas pra cima com a imagem da equação sendo mostrada. Em caso de não serem um par as duas são viradas para o lado inverso escondendo o seu conteúdo.

– resetaCampo: esta função limpa as 2 equações renderizadas no telão, altera o estado de quantidadeDeCartasViradas para 0 e limpa o vetor de 2 posições que armazena as cartas viradas no momento.

– escondeCartaDoTelao: recebe como parâmetro a imgSrc da imagem a ser escondida. Para isso também é necessário decrementar a quantidade de cartas viradas em uma unidade e então checar qual das duas cartas do telão que tem a imgSrc igual ao que foi passado para esta função, para então poder setá-lo como nulo.

- ARQUIVO:Telao

- ARQUIVO:LinhaCampo

encaminha a função handle para cada carta do campo assim como o conteúdo da carta também, ou seja, a sua imagem, seja uma carta de pergunta, ou de resposta.

- ARQUIVO:Carta

- mostra: alterna o estado da variável mostrar e troca o valor da animação de 0 graus para 180 graus, realizando o efeito de virar a carta.
- esconde: alterna o estado da variável mostrar e troca o valor da animação de 180 graus para 0 graus, realizando o efeito de virar a carta e mostrar o fundo.
- viraCarta: checa o estado da variável mostrar para decidir se chama a função mostra ou esconde e chama a função handle que lida com as regras do jogo da memória.
- desabilita: troca o estado da variável deixarDesabilitado, a qual é chamada apenas quanto um par de cartas gêmeas é encontrado. Desse modo impede que a carta seja rotacionada desse momento para frente, deixando-a sempre virada com a parte do conteúdo para cima, porém desabilitada para cliques.

4.5 Servidor aprEnDO

O servidor aprendo hospedado no heroku serve para receber estatísticas dos aplicativos enviados pelos alunos. Os alunos para enviar a estatística precisam informar o número da matrícula. As informações são enviadas em formato JSON e armazenadas no banco de dados integrado com o servidor.

O jogo aprEnDO além do aplicativo de celular tem um servidor para contabilizar dados do jogo. O servidor está hospedado no heroku utilizando uma contra free que esteve disponível durante a fase de teste dos alunos, apesar de gratuito comparado com a necessidade que era necessário deu para suprir todas as expectativas. Junto com a aplicação foi adicionado um plugin de banco de dados mongodb. Este banco armazenou em um documento os metadados do jogo. O mongodb é um add-on no mLab que permite integração com o heroku, ele permite uma conta temporária com uma quantidade limitada de capacidade de dados, mas para o necessário que são apenas arquivo json. O único problema é que o mLab não recomenda utilizar a conta free para a produção por não produzir replicação, porém o banco estava sendo acompanhado muitas vezes por dia para coletar os dados recebidos dos alunos. O servidor foi escrito em node.js. O script index.js do servidor cria uma rota /estatísticas que é a utilizada pelos aplicativos para enviar a requisição POST.

O servidor está hospedado no github do heroku, pela interface da linha de comando (CLI) pode ser clonado o projeto com o comando 'heroku git:clone -a servidor-aprendo'. Para desenvolvimento foi utilizado uma única branch, que foi o necessário. Mais branches seriam criadas, caso fosse necessário. Tendo em vista que o servidor deveria estar pronto até o início da data de aplicação do jogo com os alunos, até esta data o servidor estava sendo desenvolvido e testado, o único problema crucial que poderia ocorrer seria não realizar os commits dos códigos adicionados e removidos. Assim que o servidor estivesse pronto, a expectativa era não precisar alterar mais código nele, apenas se o servidor quebrasse por algum caso inesperado, aí sim o servidor teria de entrar no ar novamente e novas branches de correção seriam criadas para não impedir o funcionamento parcial do servidor. Porém como os logs do servidor estavam sendo acompanhados muitas vezes por dia durante o período de aplicação e o servidor não caiu nenhuma vez, não foi necessário criar branches adicionais para a correção de erros.

Ao receber os dados json enviados do aplicativo do celular para o servidor, o mesmo concatena todos os dados que chegam em uma variável apenas. Ao terminar a leitura dos dados de chegada eles são formatados é escrito uma resposta de sucesso verdadeiro, a função 'lê histórico' é chamada e a resposta é encerrada.

A função 'lê histórico' recupera o documento existente no banco de dados heroku_41w86511 hospedado no mLab, da coleção chamada histórico. A função analisa checa se a matrícula recebida pela requisição já havia alguma vez enviado estatísticas ou se é uma matrícula nova. Em caso de matrícula nova, uma nova linha é adicionada com chave primária a matrícula do aluno e salva no banco. Em caso de matrícula existente é iterado sobre classificação e resolução e todas suas fases e incrementando a quantidade de vitórias a mais que foi concluído das fases no jogo e em seguida o arquivo é atualizado no banco do mLab.

4.6 Ambiente de desenvolvimento

Falar a respeito do emulador utilizado para o desenvolvimento, outros 2 celulares além do emulador foram utilizados. Como abrir o emulador e o servidor do react-native. Como fazer a build do app no celular e como olhar o logs para analisar e debuggar. Citar das funcionalidades de feedback que não foram finalizadas, então foi solicitado aos alunos que enviassem qualquer problema ou *bug* encontrado para o número pessoal. Apesar de testar em outras telas e versões de celular, ainda foi relatado um celular entre os alunos que utilizaram o jogo no qual houve problema para rodar o módulo 2 (classificação). Colocar as fotos relatado pelo aluno.

4.7 Acesso ao código

O código do servidor está hospedado no github do heroku, disponível em <<https://dashboard.heroku.com/apps/servidor-aprendo>>.

Todo o código está hospedado no github. O banco de equações e a aplicação para android estão disponíveis no seguinte link <<https://github.com/LeonardoRk/TCC-2>>.

O APP publicado no google play pode ser encontrado através do link: <<https://play.google.com/store/apps/details?id=projeto.aprEnDO&rdid=projeto.aprEnDO>>

E a biblioteca do Wolfram Alpha em javascript pode ser baixada no link: <<https://products.wolframalpha.com/api/libraries.html>>. Porém a biblioteca já está inclusa no github dentro da pasta banco.

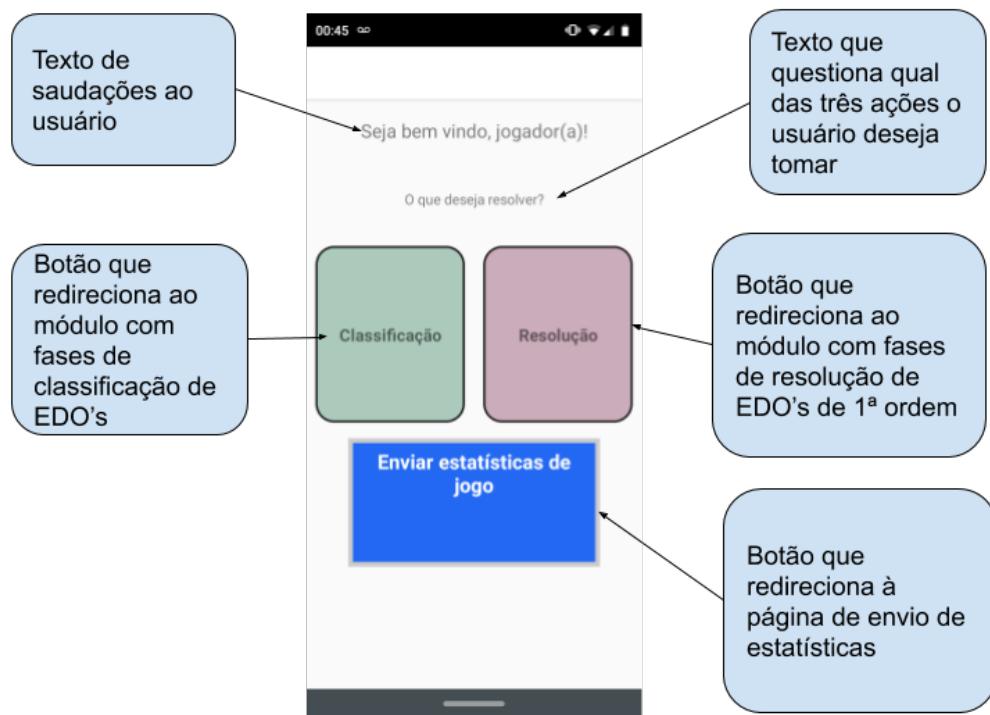
O jogo é empacotado para criar um arquivo .apk, e este que é instalado nos celulares. Para submeter o jogo ao Google Play para os jogadores poderem baixá-lo é necessário criar o .apk. O mesmo só é criado sempre que tem alguma nova atualização no jogo, seja no banco de equações ou manutenção corretiva ou evolutiva do jogo.

O empacotamento do jogo ocorre dentro da pasta do projeto react native. É utilizado o comando 'npm run android'.

5 Manual do aprEnDO

O jogo é composto em 2 pacotes: o de exercícios e o de estatísticas. É na página inicial que você pode escolher o que deseja fazer.

Figura 16 – Tela inicial



Fonte: do próprio autor

5.1 Pacote 1 - Jogo

No pacote de exercícios temos dois módulos, o de classificação e o de resolução.

5.1.1 Módulo 1: Classificação

O módulo de classificação objetiva fixar o reconhecimento e a classificação de EDs. Este é composto de seis fases: tipo, ordem, homogeneidade, linearidade, separável e exata. Essas fases foram classificadas como sendo da mais fácil para a mais complexa. Para navegar entre as fases basta arrastar a tela para a direita. Cada fase tem o domínio das suas perguntas e o número fixo de 20 equações que são selecionadas aleatoriamente de acordo com a fase.

As perguntas de tipo são:

- Escolha a ED ordinária
- Escolha a ED parcial

As perguntas de ordem são:

- Escolha a ED de ordem 1
- Escolha a ED de ordem 2
- Escolha a ED de ordem 3
- Escolha a ED de ordem superior

As perguntas de homogeneidade são:

- Escolha a ED homogênea
- Escolha a ED NÃO homogênea

As perguntas de linearidade são:

- Escolha a ED linear
- Escolha a ED NÃO linear

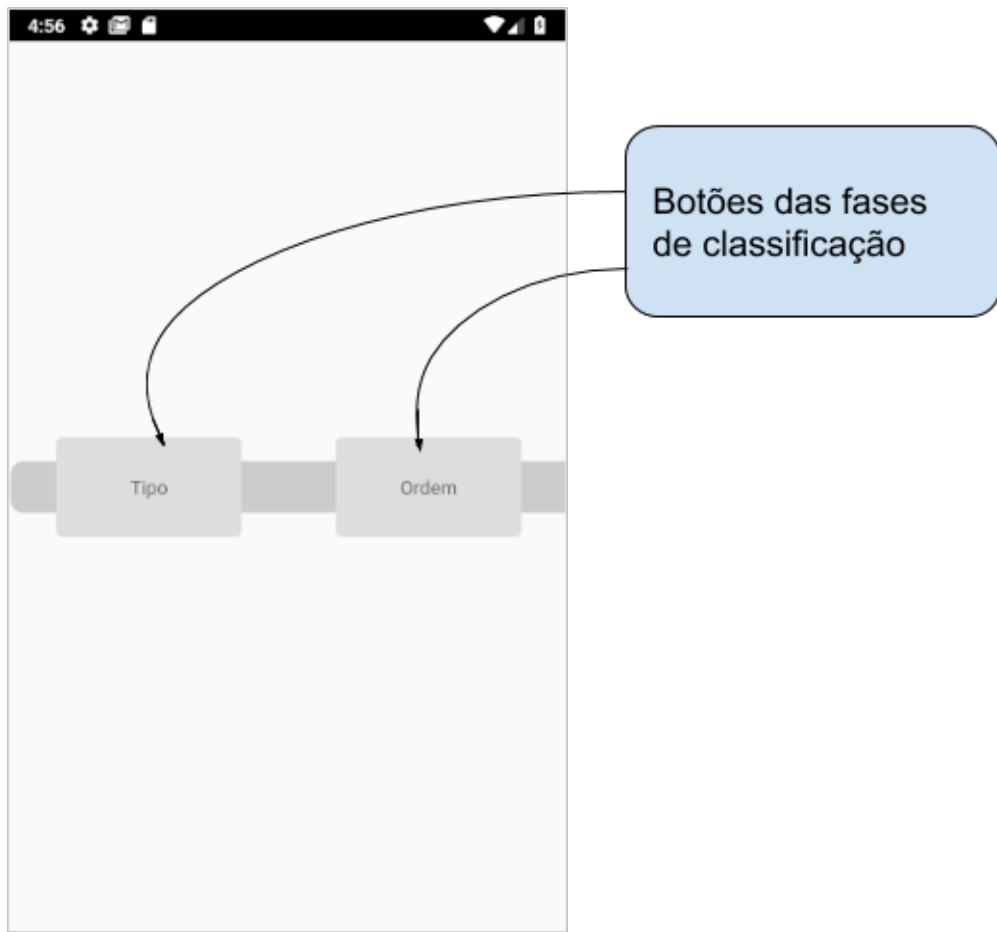
As perguntas para a fase separável são:

- Escolha a ED separável
- Escolha a ED NÃO separável

As perguntas se exata são:

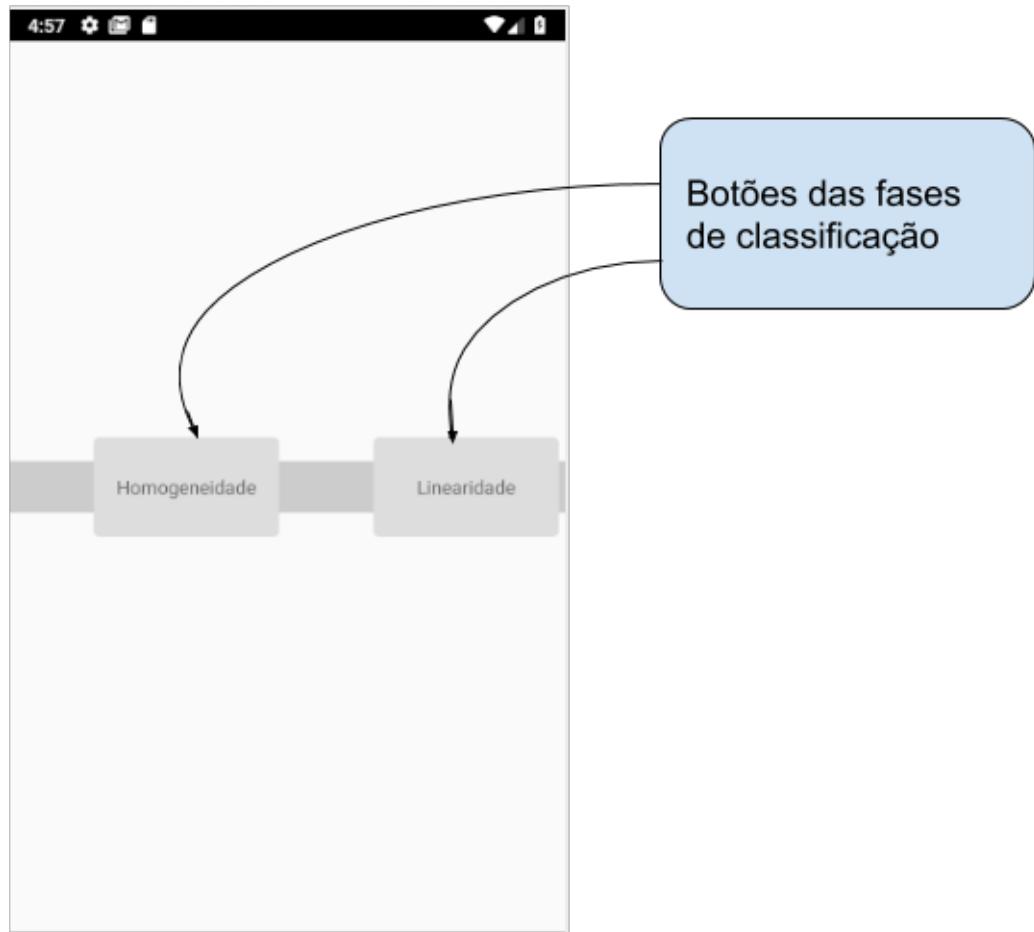
- Escolha a ED exata
- Escolha a ED NÃO exata

Figura 17 – Modo de classificação parte 1



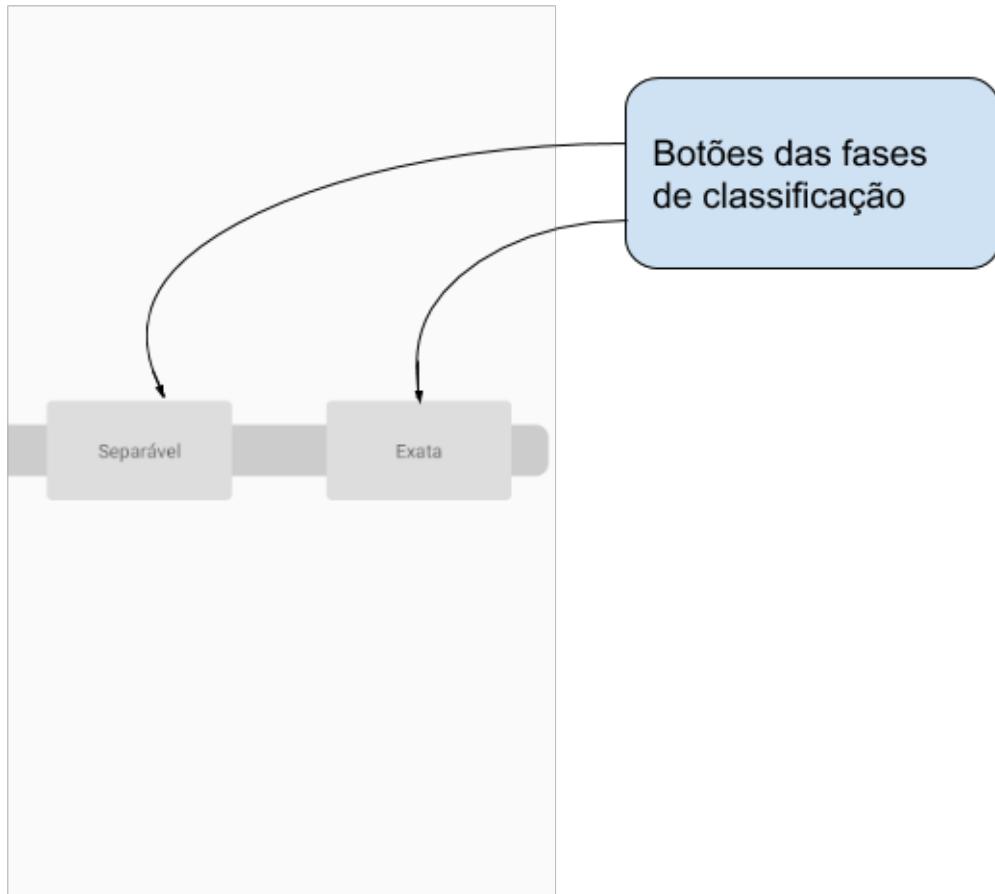
Fonte: do próprio autor

Figura 18 – Modo de classificação parte 2



Fonte: do próprio autor

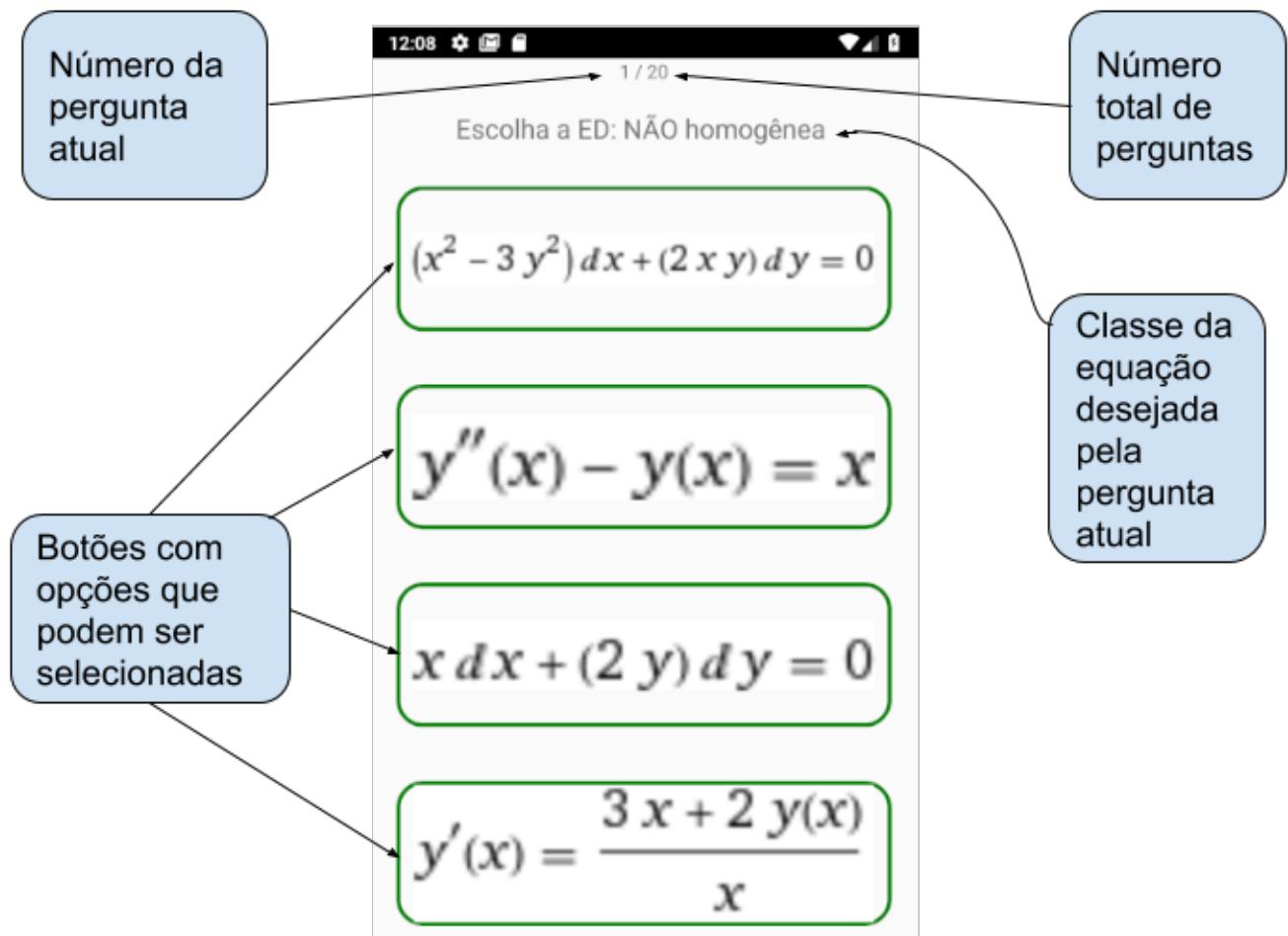
Figura 19 – Modo de classificação parte 3



Fonte: do próprio autor

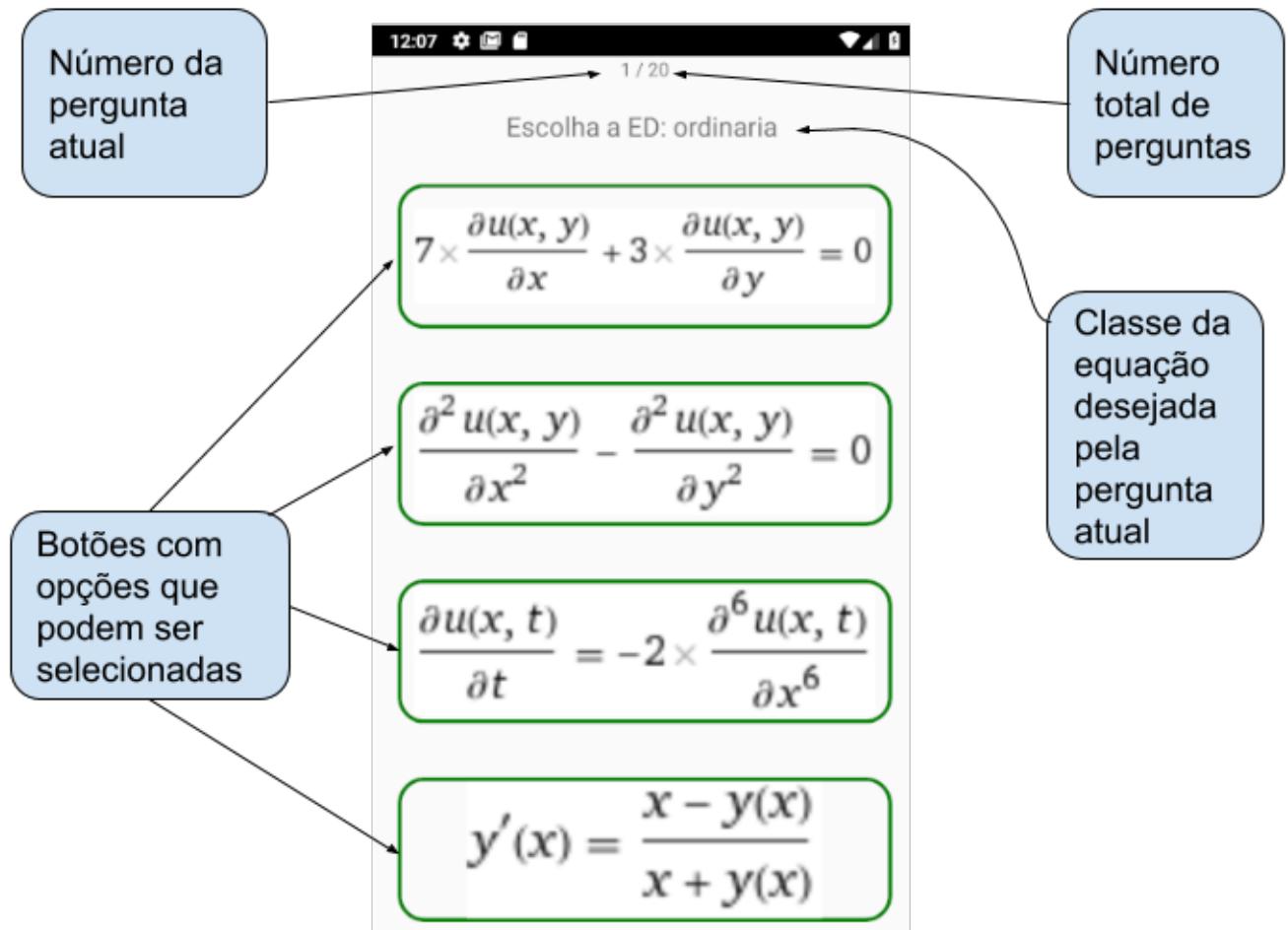
As fases de classificação apresentam quatro equações como opção sendo que apenas 1 é a correta. Ao pressionar cada opção por um tempo aparecerá a imagem da equação em uma janela modal para tentar melhorar a visualização. Para escolher uma opção basta dar um toque na opção desejada. Em caso de acerto da equação carregará automaticamente a próxima pergunta. Em caso de erro aparecerá o *feedback* da escolha da opção inválida. Ao fim da fase será redirecionado novamente para o modo de classificação, após a parabenização do jogador, onde é possível escolher outra fase para jogar ou voltar para trocar o módulo.

Figura 20 – Exemplo da fase de classificação de homogeneidade



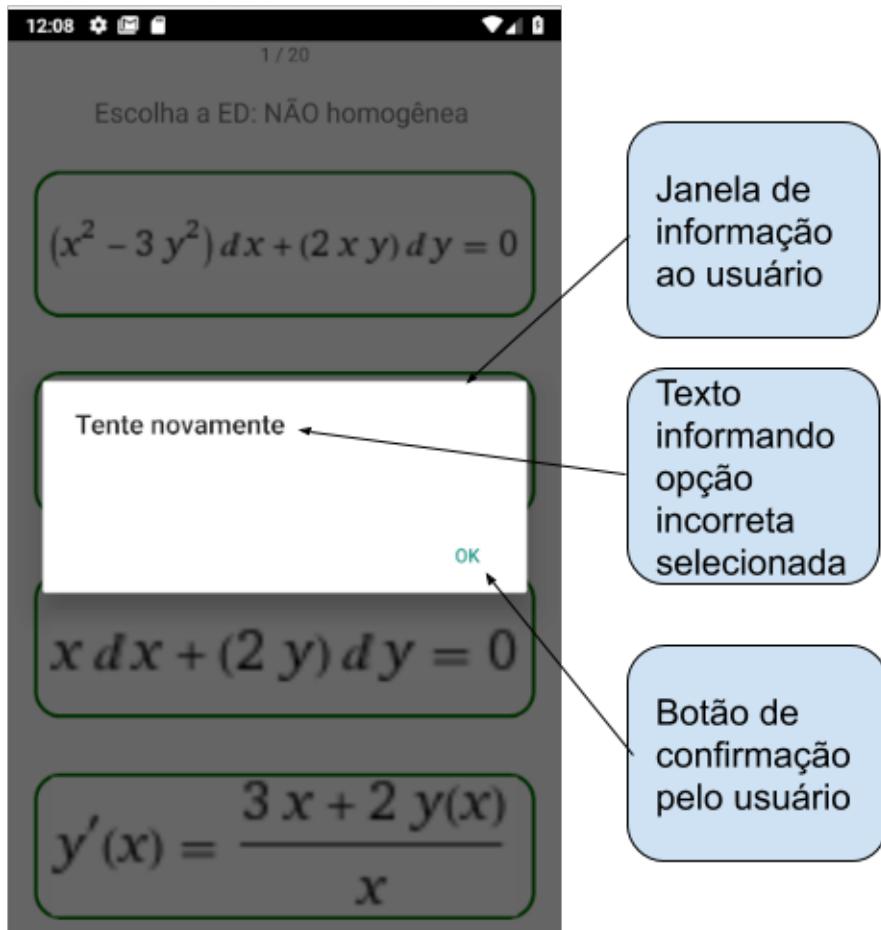
Fonte: do próprio autor

Figura 21 – Exemplo da fase de classificação de tipo



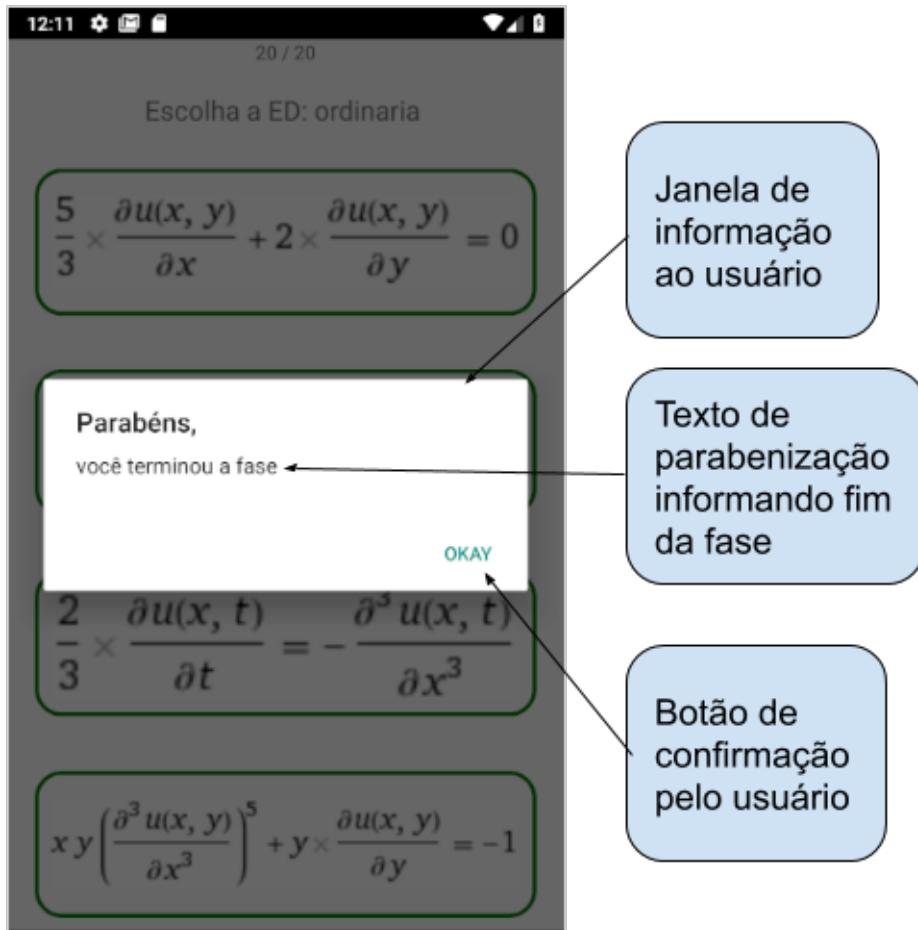
Fonte: do próprio autor

Figura 22 – Feedback fornecido à uma tentativa inválida



Fonte: do próprio autor

Figura 23 – Tela de finalização do modo de classificação

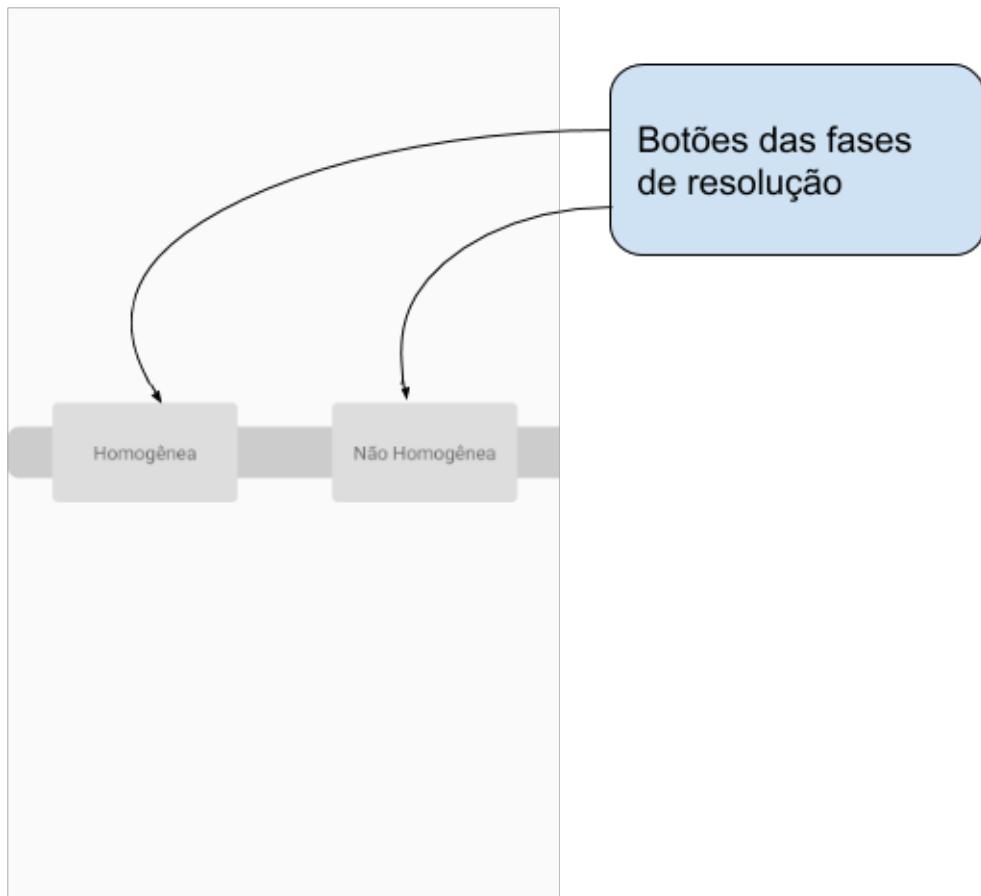


Fonte: do próprio autor

5.1.2 Módulo 2: Resolução

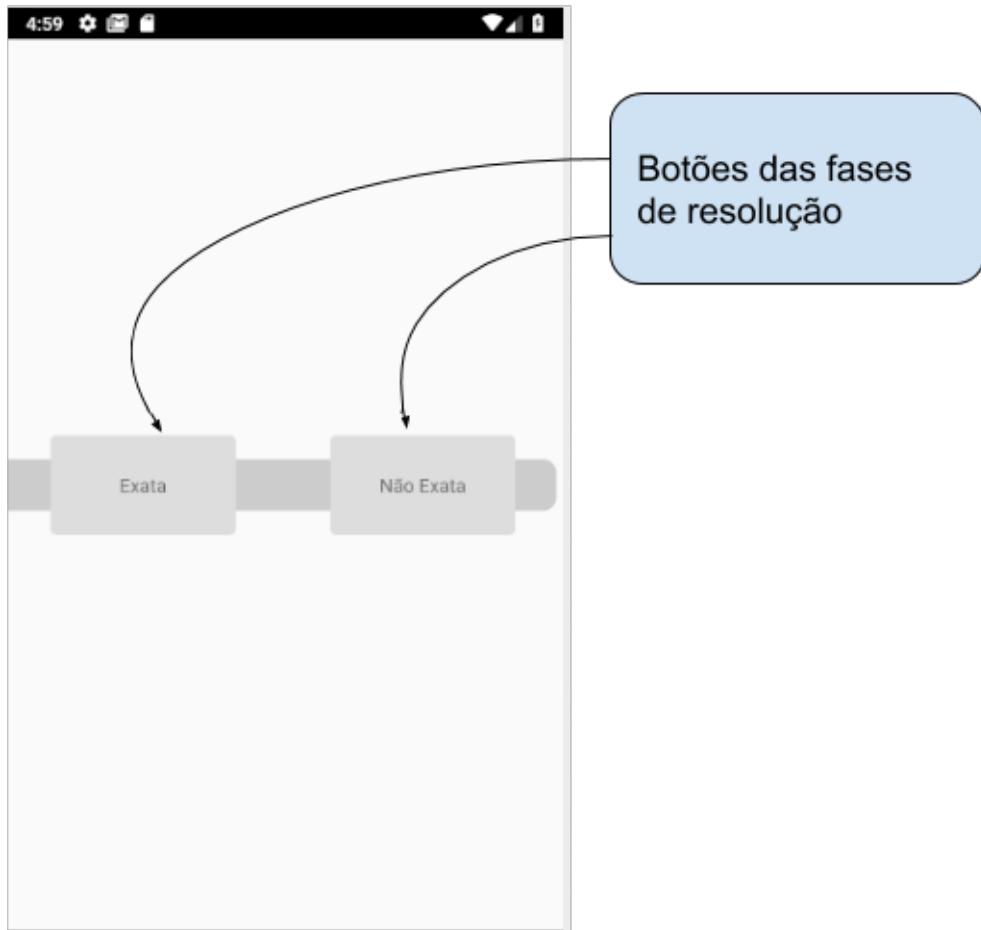
O módulo de resolução visa fixar o conhecimento de resolução de EDOs de 1^a e tem quatro fases. Sendo elas homogênea, NÃO homogênea, exatas e NÃO exatas. Neste módulo estão presentes APENAS equações diferenciais ordinárias de primeira ordem que contenham respostas.

Figura 24 – Modo de resolução parte 1



Fonte: do próprio autor

Figura 25 – Modo de resolução parte 2

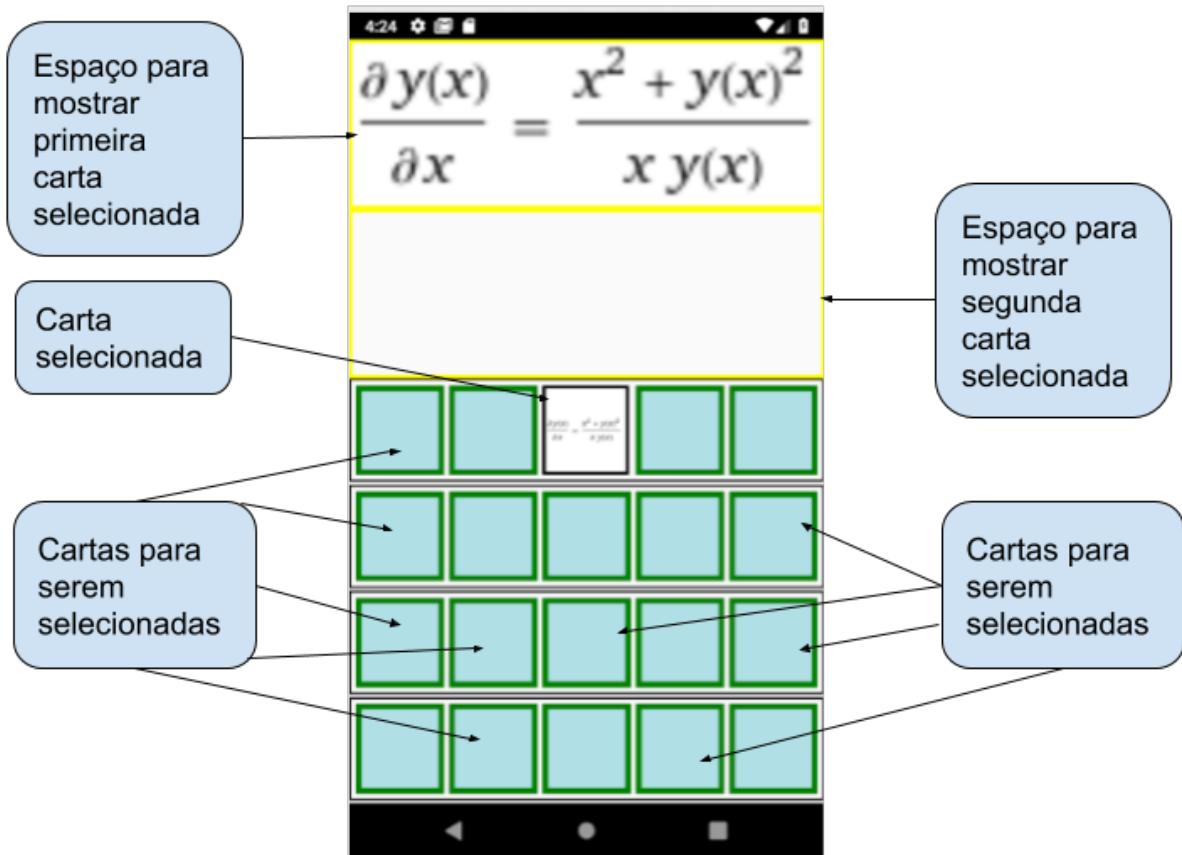


Fonte: do próprio autor

O módulo de resolução trata-se de um jogo da memória, o qual tem o objetivo de encontrar as cartas gêmeas. Uma carta contém a EDO proposta e a carta gêmea contém a solução correta para a equação proposta.

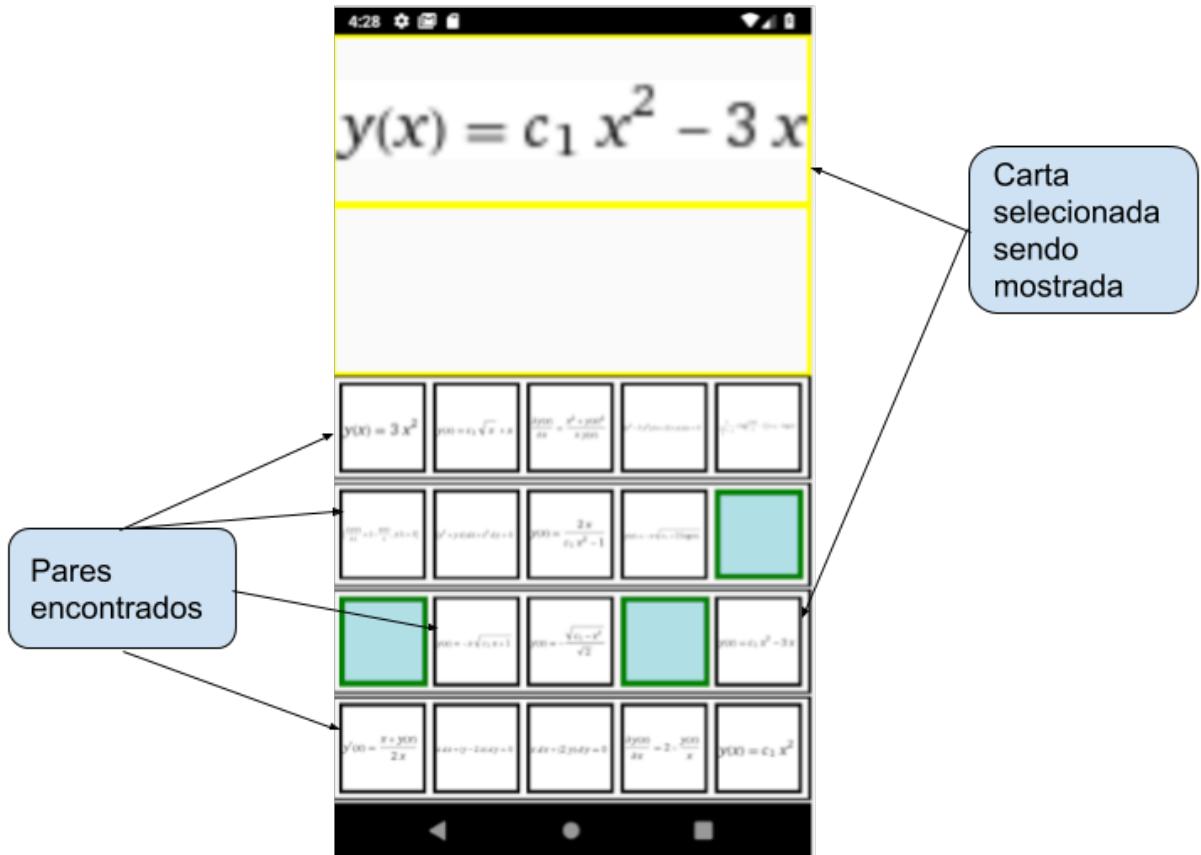
Existem dois campos reservados na tela para mostrar as cartas selecionadas. O jogo é composto de vinte cartas (dez EDOs e dez soluções, uma de cada EDO). Cada carta é selecionada com um toque sobre ela, onde sua face será mostrada e seu conteúdo se mostrará em um dos dois campos reservados. Cada clique inverte o lado da carta que está sendo mostrado. Ao selecionar duas cartas haverá a comparação se são complementares (ou seja, pergunta e resposta). Em caso afirmativo, as cartas congelarão não sendo mais possível interagir com elas, em caso negativo apenas esconderão sua face para que inicie outra tentativa do jogador.

Figura 26 – Jogo da memória com 1 carta selecionada



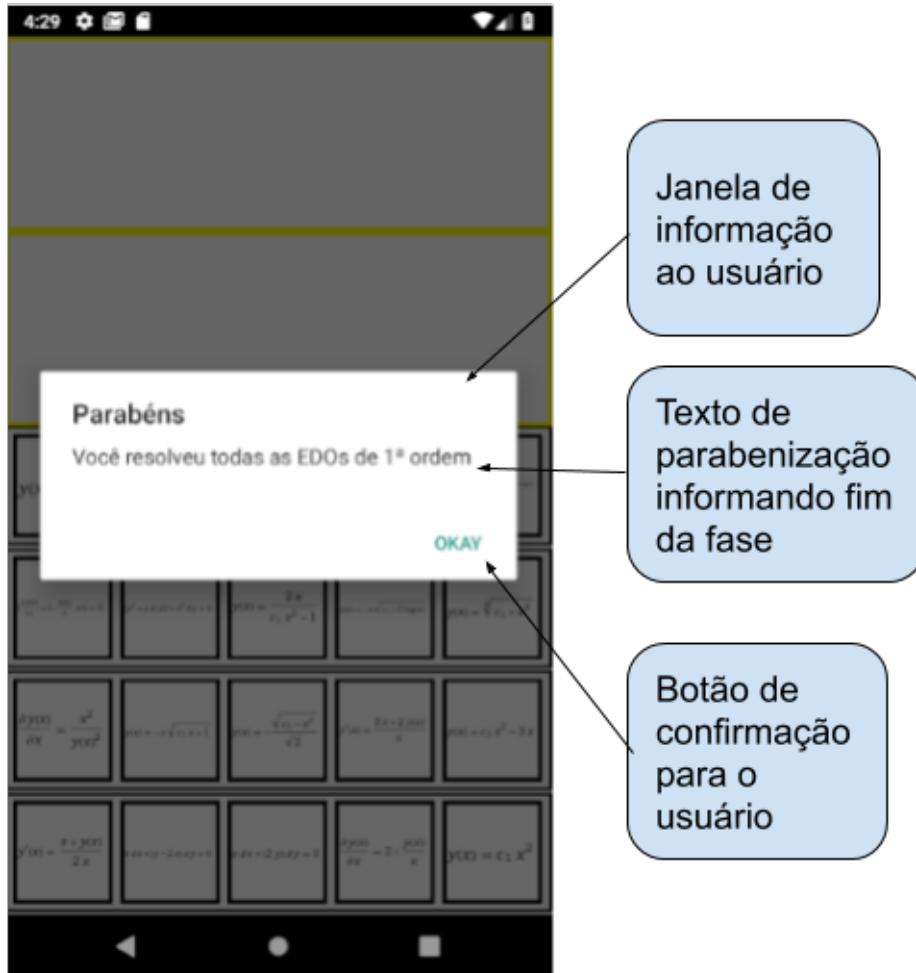
Fonte: do próprio autor

Figura 27 – Jogo da memória com alguns pares encontrados



Fonte: do próprio autor

Figura 28 – Vitória de jogo da memória



Fonte: do próprio autor

Ao terminar qualquer fase do jogo da memória, após a confirmação o jogo retornará à tela de seleção da fase a ser jogada.

5.2 Pacote 2 - Servidor

O pacote de estatísticas está relacionado à coleta de dados para a avaliação da quantidade de jogadores que usaram o jogos e dentre os que usaram, quais fases eles completaram e quantas vezes eles as completaram. O envio das estatísticas para o servidor não ocorrerá sem a ação do usuário, depende do jogador enviar os dados de acordo com a sua vontade de colaboração através de um botão na tela inicial e a confirmação do envio com a sua matrícula.

Ao se clicar no botão para enviar estatísticas do jogo, deve-se redirecionar para a tela igual a Figura 29.

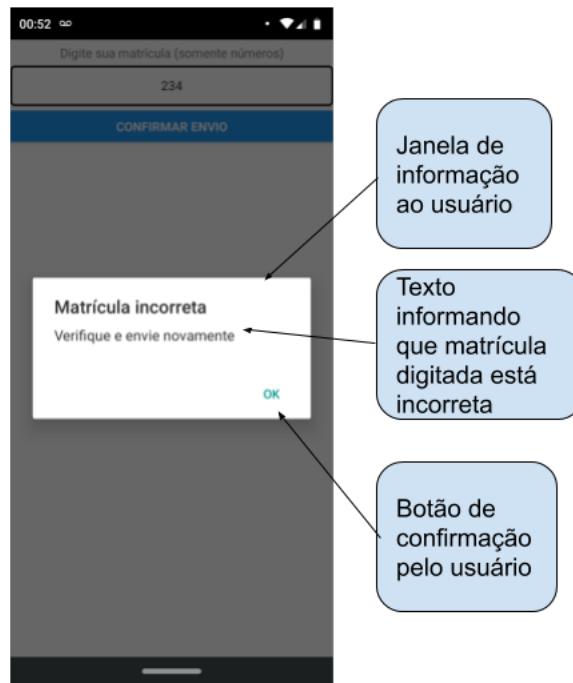
Figura 29 – Tela de envio de estatísticas



Fonte: do próprio autor

A Figura 30 apresenta a mensagem de erro ao digitar a matrícula incorreta. A checagem é feita conferindo a quantidade de dígitos que deve ser igual a 8.

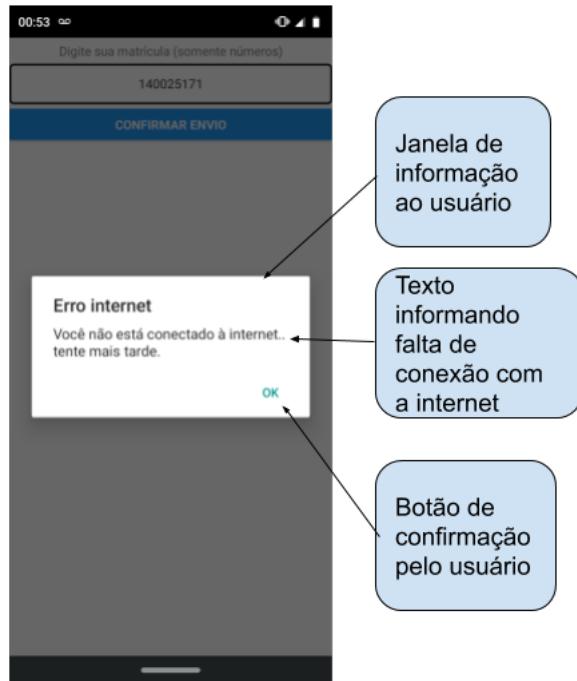
Figura 30 – Alerta de matrícula inválida



Fonte: do próprio autor

A Figura 31 apresenta a mensagem de erro quando não há conexão do celular com a internet para o envio dos dados ao servidor.

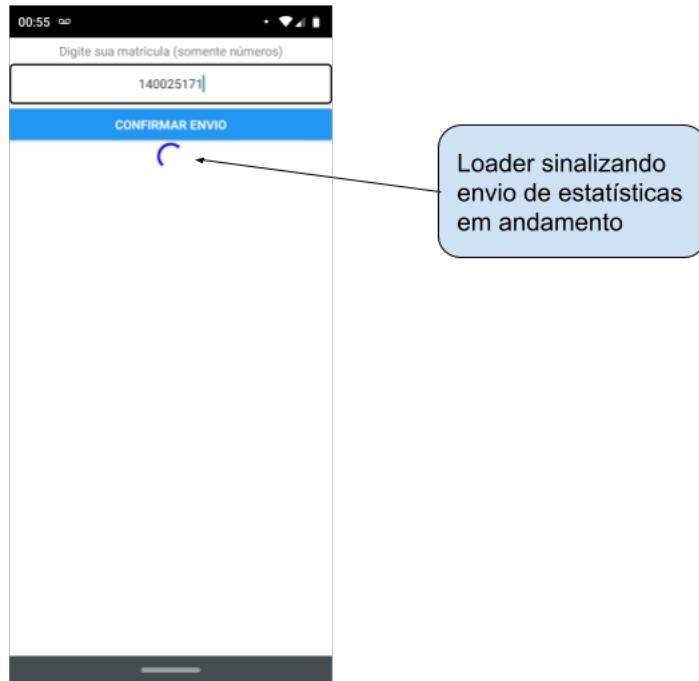
Figura 31 – Alerta de falta de internet



Fonte: do próprio autor

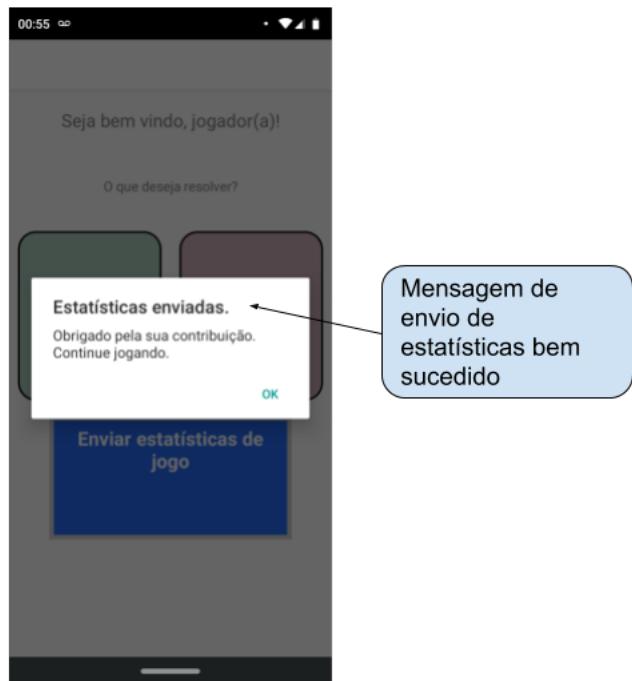
Ao ser digitado a matrícula correta e realizada a submissão do envio das estatísticas, um loader aparecerá enquanto o procedimento não termina, como mostra a Figura 32 e ao terminar o envio, uma mensagem é apresentada, como mostra a Figura 33.

Figura 32 – Loader visível enquanto envia estatísticas



Fonte: do próprio autor

Figura 33 – Alerta de envio com sucesso



Fonte: do próprio autor

Ao fim do envio, é redirecionado para à tela inicial [16](#).

6 AprEnDO: o relato de experiência e a análise

6.1 Relato de experiência

Antes de iniciar a aplicação e a disponibilização do jogo com os alunos houve uma apresentação em sala de aula realizada pela professora. Ela apresentou o aluno que aplicaria o jogo e explicou a proposta para a semana seguinte. Foi pedido aos alunos para fornecer o feedback de como estava sendo a aplicação e caso fosse encontrado algum erro, defeito ou falha. Neste dia o aplicador foi adicionado ao grupo de WhatsApp da turma onde algumas dúvidas sobre a aplicação também foram sanadas.

No primeiro dia de aplicação foram bem menos alunos do que a quantidade dos que estavam matriculados na disciplina. Um deles até havia conversado por WhatsApp perguntando se teria como jogar de forma remota pois devido à grande distância entre a casa e a FGA optaria por não ir. No segundo e terceiro dia a quantidade de alunos presente em sala de aula chegou praticamente a zero. Sendo que os que estavam indo era com o objetivo de estudar outros conteúdos da matéria (Resolução de EDO de segunda ordem).

O servidor de dados estava sendo checado os logs dos dados enviados mais de uma vez por dia, para garantir que o servidor estava disponível para receber qualquer envio de dados do jogo. O mesmo ficou online o tempo inteiro para a recepção dos dados enviados.

Com aproximadamente dois meses de aplicação e coleta de dados percebeu-se a contribuição de poucos estudantes em relação aos matriculados.

Alguns dos motivos propostos para estes resultados é que o jogo não foi testado com uma amostra do público alvo e pode apresentar baixa usabilidade. A resolução gráfica do jogo está não muito atrativa e com poucos efeitos visuais e de animação. Falta de som para animação e sem história para envolver o jogador. E por fim o jogo pode estar muito difícil para se jogar.

6.2 Análise

De 90 alunos para cima na turma, até o momento da última checagem no servidor de dados do *heroku*, contabilizou-se 13 alunos voluntários que enviaram estatísticas, porém não se sabe quantos jogaram. Sabe-se que dos 13 alunos que enviaram, 10 completaram pelo menos uma fase do módulo de classificação e 3 pessoas enviaram as estatísticas sem

ter completado nenhuma fase de nenhum módulo. Não se sabe quantas vezes eles tentaram jogar as fases e nem se eles apenas enviaram as estatísticas sem tentar.

A Figura 34 apresenta o resultado de classificação dos 13 alunos que contribuíram enviando os dados. Onde está escrito falso significa que não foi vencido nenhuma vez a fase (maioria dos casos). E alguns alunos jogaram apenas 1 vez, outros 2 e um aluno jogou 8 vezes.

Figura 34 – Estatísticas colhidas de classificação

Alunos	C_exata	C_homogeneidade	C_linearidade	C_ordem	C_separável	C_tipo
1	1	1	FALSO	1	FALSO	1
2	1	1	FALSO	1	FALSO	1
3	1	FALSO	FALSO	1	FALSO	1
4	FALSO	1	1	1	1	2
5	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
6	FALSO	FALSO	FALSO	FALSO	FALSO	1
7	FALSO	FALSO	FALSO	1	FALSO	FALSO
8	FALSO	1	1	1	FALSO	1
9	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
10	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
11	FALSO	FALSO	FALSO	1	FALSO	1
12	FALSO	FALSO	FALSO	1	FALSO	1
13	FALSO	FALSO	FALSO	8	FALSO	FALSO

A Figura 35 apresenta o resultado de resolução dos 13 alunos que contribuíram enviando os dados. Onde está escrito falso significa que não foi vencido nenhuma vez a fase (maioria dos casos). E alguns alunos jogaram apenas 1 vez algumas fases, porém a minoria venceu este módulo, percebe-se que sua aceitação não foi muito boa por algum motivo. Apenas 3 alunos jogaram pelo menos 1 vez 1 fase.

Figura 35 – Estatísticas colhidas de resolução

Alunos	R_homogêneo	R_não_homogêneo	R_exato	R_não_exato
1	FALSO	FALSO	FALSO	FALSO
2	FALSO	FALSO	FALSO	FALSO
3	1	FALSO	1	FALSO
4	FALSO	FALSO	FALSO	FALSO
5	FALSO	FALSO	FALSO	FALSO
6	FALSO	FALSO	FALSO	FALSO
7	1	FALSO	FALSO	FALSO
8	FALSO	FALSO	1	FALSO
9	FALSO	FALSO	FALSO	FALSO
10	FALSO	FALSO	FALSO	FALSO
11	FALSO	FALSO	FALSO	FALSO
12	FALSO	FALSO	FALSO	FALSO
13	FALSO	FALSO	FALSO	FALSO

Foi planejado coletar mais dados do servidor, porém por falta de organização do

tempo não foi possível coletar todos os dados conforme o planejado.

Foram planejados desenvolver casos de teste, porém também com o mau planejamento de tempo, a grande dificuldade em desenvolver e executar os testes unitários e de integração automatizados estes acabaram não sendo desenvolvidos.

Abaixo segue uma breve descrição dos testes que foram planejados testar:

Testes de servidor no heroku:

- Garantir que cada matrícula seja única. **OBS:** Apesar do teste não ter sido implementado, existe a checagem na própria função do arquivo para garantir que a matrícula é nula. Foram realizados testes manuais para confirmar que a funcionalidade está funcionando, porém não há testes escritos.
- Garantir que os dados recebidos sejam salvos no servidor em arquivo. **OBS 1:** inicialmente acreditava-se que salvaria os dados no heroku em arquivo, mas percebeu-se que a cada novo envio uma instância era iniciada e os dados antigos não eram persistidos, a ideia foi migrar para a utilização de um banco de dados na nuvem com a ajuda do mLab. **OBS 2:** Apesar do teste não ter sido implementado, garante-se que os dados recebidos estão sendo salvos no servidor, porém não foram desenvolvidos testes para melhores confirmações e indicações de falhas.

Testes de banco de equações:

- Garantir que toda resposta tenha uma pergunta **OBS:** no próprio desenvolvimento do código da aplicação existe o filtro de selecionar apenas equações com respostas, porém não há testes para a confirmação que o mesmo está funcionando sempre de acordo com o esperado, apesar das muitas validações manuais.

Testes da aplicação react-native:

Pacote 1

- Garantir que carregue todas fases do módulo classificação. **OBS:** apenas testes manuais e exaustivos foram realizados.
- Garantir que carregue todas fases do módulo resolução. **OBS:** apenas testes manuais e exaustivos foram realizados.
- Testar que quantidade de perguntas de classificação é 20. **OBS:** apenas testes manuais foram realizados.
- Testar que quantidade de cartas de resolução sejam 20. **OBS:** apenas testes manuais foram realizados.

- Testar que jogo da memória tenha apenas 10 equações perguntas. **OBS:** apenas testes manuais foram realizados.
- Testar que jogo da memória tenha apenas 10 equações respostas. **OBS:** apenas testes manuais foram realizados.
- Testar que cada pergunta do jogo tenha 1 e apenas uma resposta. **OBS:** existe apenas o filtro no código da aplicação sem testes escritos.
- Testar que apenas duas cartas estejam viradas para cima mostrando seu conteúdo. **OBS:** apenas testes manuais foram realizados.
- Testar em classificação que cada pergunta só tenha uma resposta correta. **OBS:** apenas testes manuais e exaustivos foram realizados.
- Testar em classificação que cada pergunta tenham 4 alternativas diferentes. **OBS:** apenas testes manuais e exaustivos foram realizados.

Pacote 2

- Garantir que ao enviar estatística a quantidade de vezes jogado zere para o módulo 1 e 2. **OBS:** apenas testes manuais e exaustivos foram realizados.
- Garantir que ao enviar estatística o tempo total de cada fase zere para o módulo 1 e 2. **OBS:** Não foi coletado o tempo jogado de cada fase.
- Garantir que ao enviar estatísticas ao servidor a informação a respeito das fases completas nos módulos permaneça a mesma e zere apenas quantas vezes foi vencido. **OBS:** apenas testes manuais e exaustivos foram realizados.

Não foi medido o tempo total gasto com o desenvolvimento dos códigos e nem o custo. O único custo que o projeto teve foi a licença da google play para poder publicar o jogo. O valor da licença foi de 25 dólares, aproximadamente 110 reais.

7 Conclusão

Como poucos alunos jogaram e apresentam-se os conceitos esperados nos mapas conceituais acredita-se que o resultado da pesquisa é inconclusivo, pois não se sabe se o jogo que ajudou a aprender ou foram apenas as aulas e os estudos por conta própria. A ideia inicial era avaliar a nota da questão da prova geral a qual os alunos realizarão para comparar a turma que jogou com turmas que não jogaram para avaliar se o jogo surtiu efeito ou não no aprendizado.

Poderiam ter sido feitos mais testes e realizá-los junto com o desenvolvimento das funcionalidades, onde os casos de teste encontram-se mais frescos na mente. Poderia ter tido um planejamento melhor com o levantamento das funcionalidades do jogo e a descrição dos requisitos. O código pode estar mais limpo e modularizado (refatorado) e o jogo poderia ter sido pensado melhor com mais detalhes em fases e animações.

Para trabalhos futuros podem ser pensadas em mais funcionalidades para o jogo, como a criação de um personagem e ganho de itens para utilizações no jogo.

Referências

- BENITTI, F. B. V.; MOLLéRI, J. S. Utilização de um rpg no ensino de gerenciamento e processo de desenvolvimento de software. **WEI - Workshop sobre Educação em Computação**, 07 2008. Citado na página 18.
- COELHO, V. M. **O jogo como prática pedagógica na escola inclusiva**. 2010. Acessado em: 22/10/2018. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/1485/Coelho_Vania_Maria.pdf?sequence=1>. Citado 2 vezes nas páginas 12 e 13.
- CORREIA, P. R. M.; VALLE, B. X. do; SILVA, J. G. R. J. A. C. da. Mapas conceituais como ferramenta de avaliação: Desafios e possibilidades de mudanças na sala de aula. **Encontro Nacional de Pesquisa em Educação em Ciências**, 2009. ISSN 21766940. Citado na página 16.
- COSTA, M. H. C. da; SILVA, G. M.; RIBEIRO, T. N. O uso de mapas conceituais como instrumento de avaliação no ensino de física: Um estudo a partir do tema as leis de newton. 2012. Disponível em: <http://educonse.com.br/2012/eixo_06/PDF/75.pdf>. Citado na página 17.
- DICHEVA, D. *et al.* Gamification in education: A systematic mapping study. **Educational Technology & Society**, v. 18, p. 75–88, 07 2015. Citado na página 14.
- DUPAUL, G. J.; STONER, G. **TDAH nas escolas - Estratégias de Avaliação e Intervenção**. 1ª edição. ed. São Paulo: M. Books do Brasil Editora Ltda, 2007. 130 p. ISBN 978-85-7680-017-0. Citado na página 12.
- FILHO, A. M. D. S. Software everywhere: sobre a demanda de software e da engenharia de software. 2015. ISSN 1519-6186. Disponível em: <<http://periodicos.uem.br/ojs/index.php/EspacoAcademico/article/view/29122/15124>>. Citado na página 17.
- FUKUSAWA, J. e; CARNIELLO, A. Jogos digitais no ensino e aprendizagem de engenharia. v. 1, p. 1–8, 2015. Citado 2 vezes nas páginas 12 e 13.
- GIACINTI, M. *et al.* A video game based on elementary differential equations. **Scientific Research**, 2013. Citado 2 vezes nas páginas 9 e 14.
- IEEE. Ieee standard glossary of software engineering terminology. 1990. Disponível em: <<https://ieeexplore-ieee-org.ez54.periodicos.capes.gov.br/stamp/stamp.jsp?tp=&arnumber=159342>>. Citado na página 17.
- INEP. **Resultados de Leitura e Matemática - equipe nacional**. 2015. 27-29 p. Accessado em: 26/10/2018. Disponível em: <http://download.inep.gov.br/acoes_internacionais/pisa/resultados/2015/pisa_apresentacao_leitura_e_matematica.pptx>. Citado na página 9.
- KATMADA, A.; MAVRIDIS, A.; TSIATSOS, T. Implementing a game for supporting learning in mathematics. v. 12, 2014. Citado 2 vezes nas páginas 13 e 19.

- MICHAELIS, D. **Dicionário Brasileiro da Língua Portuguesa.** 2019. Disponível em: <<http://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/jogo/>>. Citado na página 12.
- MONSALVE, E. S.; WERNECK, V. M. B.; CESAR, J. Simules-w : Um jogo para o ensino de engenharia de software. **Anais do III Fórum de Educação em Engenharia de Software**, p. 17–26, 2010. Citado 2 vezes nas páginas 14 e 18.
- NETO, J. C.; BLANCO, M. B.; SILVA, J. A. da. O uso de gamificação e dificuldades matemáticas: possíveis aproximações. **RENOTE - Revista Novas Tecnologias na Educação**, v. 15, n. 1, 2017. ISSN 1679-1916. Disponível em: <<https://seer.ufrgs.br/renote/article/download/75151/42586>>. Citado 2 vezes nas páginas 9 e 14.
- NOVAK, J. D. Aprender, criar e utilizar o conhecimento: mapas conceituais como ferramentas de facilitação nas escolas e empresas. 2000. Citado 2 vezes nas páginas 15 e 17.
- NUNES, I. F.; PARREIRA, P. A. Rpg4sorting - um jogo educacional para auxílio ao ensino de métodos de ordenação. 2015. Citado 2 vezes nas páginas 12 e 14.
- SANTOS, L. A. F. **Software gamificado para auxílio ao ensino e aprendizagem de matemática para crianças.** Brasília, Brazil, p. 75, 2017. Citado na página 12.
- SAVI, R.; ULBRICHT, D. V. R. Jogos digitais educacionais: Benefícios e desafios. 2008. Citado 2 vezes nas páginas 13 e 14.
- SILVA, V. *et al.* Proposta de um aplicativo gamificado para o ensino de cálculo. **Congresso Regional sobre Tecnologias na Educação**, Recife/PE - Brasil, 2016. Disponível em: <http://ceur-ws.org/Vol-1667/CtrIE_2016_AC_paper_14.pdf>. Citado 3 vezes nas páginas 9, 12 e 13.
- SOUZA, L. F. D. de. **Evasão do curso de Licenciatura em Matemática (Noturno) da Universidade de Brasília.** Brasília, 2016. Citado na página 9.
- SOUZA, M.; FRANÇA, C. O que explica o sucesso de jogos no ensino de engenharia de software? uma teoria de motivação. **WEI - 24º Workshop sobre Educação em Computação**, 08 2016. Citado 4 vezes nas páginas 12, 13, 14 e 18.
- SOUZA, N. A. de; BORUCHOVITCH, E. Mapa conceitual: seu potencial como instrumento avaliativo. v. 21, p. 173–192, 2009. Citado 3 vezes nas páginas 15, 16 e 17.
- SUBJACENTE, A. S. U. C. Marco antonio moreira. UFRGS, 2011. Citado 2 vezes nas páginas 15 e 22.