

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

RENAN VINICIUS FIORINI SUZIN

**DESENVOLVIMENTO DE UM CONJUNTO DE JOGOS
EDUCACIONAIS INFANTIS PARA ENSINO BÁSICO DA
MATEMÁTICA**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2017**

RENAN VINICIUS FIORINI SUZIN

**DESENVOLVIMENTO DE UM CONJUNTO DE JOGOS
EDUCACIONAIS INFANTIS PARA ENSINO BÁSICO DA
MATEMÁTICA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Vinicius Pegorini

**PATO BRANCO
2017**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento
de Sistemas



TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

DESENVOLVIMENTO DE UM CONJUNTO DE JOGOS EDUCACIONAIS INFANTIS PARA ENSINO BÁSICO DA MATEMÁTICA

por

RENAN VINICIUS FIORINI SUZIN

Este trabalho de conclusão de curso foi apresentado no dia 26 de junho de 2017, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof. Me. Vinicius Pegorini
Orientador

Profª Me. Samantha De Castro Canete

Profª Drª Beatriz Terezinha Borsoi

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Profª Drª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Que Deus perdoe essas pessoas ruins.

Adriano Leite Ribeiro

RESUMO

SUZIN, Renan Vinicius Fiorini. Desenvolvimento de um conjunto de jogos educacionais infantis para ensino básico da matemática. 2017. 55f. Monografia (Trabalho de Conclusão de Curso) - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

O uso da tecnologia como ferramenta de ensino tem ganhado cada vez mais espaço nas escolas. Essa tendência faz parte da atualização do modelo de ensino, visando acompanhar as novas gerações que desde os anos iniciais de vida tem contato direto com dispositivos eletrônicos. A utilização dessas novas ferramentas vem ao encontro do uso de jogos educacionais, que por sua vez podem desempenhar um papel relevante no processo de aprendizagem, pois proporcionam uma forma que pode ser mais atraente e intuitiva de aprender. Este trabalho tem o objetivo de desenvolver um conjunto de jogos para dispositivos móveis para auxílio no ensino da matemática. Os jogos são voltados para crianças entre 6 e 12 anos e poderão ser utilizados tanto em sala de aula como em casa e trabalham assuntos como: operações básicas, associação de unidades de medidas e resolução de frações.

Palavras-chave: Aplicativo. Dispositivos Móveis. Jogo. Educacional.

ABSTRACT

SUZIN, Renan Vinicius Fiorini. Development of sets of children's educational games for basic education in mathematics. 2017. 55f. Monografia (Trabalho de Conclusão de Curso) - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

The use of technology as a teaching tool has been gaining more and more space in schools and is part of updating the current teaching model to accompany the new generations that since the early years of life have contact with electronic devices. The use of these new tools comes with the use of educational games, which play a very important part in the learning process they provide a more attractive and intuitive way of learning. This work aims to develop a set of games for mobile devices to aid in the teaching of mathematics. The games are appropriate for children between 6 and 12 years old can be used both in the classroom and at home and include subjects such as: basic operations. association of units of measurement and resolution of fractions.

Keywords: Games. Mobile Devices. Teaching. Learning.

LISTA DE FIGURAS

Figura 1 - Diagrama de casos de uso do aplicativo Space Bingo	22
Figura 2 - Fluxograma do aplicativo Space Bingo	23
Figura 3 - Tela de configurações do aplicativo Space Bingo	24
Figura 4 - Tela principal do aplicativo Space Bingo.....	25
Figura 5 - Tela principal do aplicativo Space Bingo com a cartela do adversário em destaque	26
Figura 6 - Animação apresentada em tela após a vitória do adversário.....	27
Figura 7 - Tela apresentada ao final do jogo do aplicativo Space Bingo.....	28
Figura 8 - Estrutura de pastas e arquivos do aplicativo Space Bingo	29
Figura 9 - Diagrama de casos de uso do aplicativo Desafio das Medidas	34
Figura 10 - Fluxograma do aplicativo Desafio das Medidas	35
Figura 11 - Tela de configurações do aplicativo Desafio das Medidas.....	36
Figura 12 - Tela de pergunta do aplicativo Desafio das Medidas.....	36
Figura 13 - Tela de perguntas após uma resposta errada do aplicativo Desafio das Medidas.....	37
Figura 14 - Tela de contagem regressiva do aplicativo Desafio das Medidas.....	37
Figura 15 - Tela principal do aplicativo Desafio das Medidas	38
Figura 16 - Estrutura de pastas e arquivos do aplicativo Desafio das Medidas	38
Figura 17 - Diagrama de casos de uso do aplicativo Memória Matemática	44
Figura 18 - Fluxograma do aplicativo Memória Matemática	44
Figura 19 - Tela principal do aplicativo Memória Matemática	45
Figura 20 - Tela de configurações do aplicativo Memória Matemática.....	46
Figura 21 - Tela principal do aplicativo Memória Matemática na configuração Fácil	46
Figura 22 - Tela principal do aplicativo Memória Matemática na configuração Médio	47
Figura 23 - Tela principal do aplicativo Memória Matemática na configuração Difícil	47
Figura 24 - Tela principal do aplicativo Memória Matemática com a opção de combinar imagens marcada	48
Figura 25 - Tela principal do aplicativo Memória Matemática com a opção de combinar imagens desmarcada	48
Figura 26 - Tela apresentada ao usuário após o fim de jogo do aplicativo Memória Matemática.....	49
Figura 27 - Tela de ajuda do aplicativo Memória Matemática	49
Figura 28 - Estrutura de pastas e arquivos do aplicativo Memória Matemática	50

LISTA DE QUADROS

Quadro 1 - Ferramentas utilizadas no desenvolvimento do trabalho	16
Quadro 2 - Fluxos e iterações de trabalho	18
Quadro 3 - Space Bingo - Requisitos Funcionais.....	21
Quadro 4 - Space Bingo - Requisitos não funcionais	22
Quadro 5 - Desafio das Medidas - Requisitos funcionais.....	33
Quadro 6 - Desafio das Medidas - Requisitos não funcionais.....	34
Quadro 7 - Memória Matemática - Requisitos funcionais.....	43
Quadro 8 – Memória Matemática - Requisitos não funcionais	44

LISTAGEM DE CÓDIGO

Listagem 1 - Classe principal do aplicativo Space Bingo	30
Listagem 2 - Declaração dos objetos de menu do aplicativo Space Bingo	30
Listagem 3 - Método de abertura da janela de configurações do aplicativo Space Bingo	31
Listagem 4 - Método de criação da cartela de respostas do aplicativo Space Bingo	31
Listagem 5 - Método de criação dos resultados possíveis das multiplicações do aplicativo Space Bingo	32
Listagem 6 - Classe principal do aplicativo Desafio das Medidas	39
Listagem 7 - Trecho do método de configuração de jogo do aplicativo Desafio das Medidas	39
Listagem 8 - Método de movimentação da carroça do usuário do aplicativo Desafio das Medidas	40
Listagem 9 - Método de inicialização do desafio do aplicativo Desafio das Medidas	40
Listagem 10 - Declaração das lista das unidades de medida do aplicativo Desafio das Medidas	40
Listagem 11 - Método executado após o click do usuário em tela do aplicativo Desafio das Medidas	41
Listagem 12 - Trecho do código do método que adiciona uma penalidade ao usuário após uma resposta errada do aplicativo Desafio das Medidas	42
Listagem 13 - Método de inicialização de jogo do aplicativo Memória Matemática ...	50
Listagem 14 - Método de configuração de jogo do aplicativo Memória Matemática	50
Listagem 15 - Trecho do código de criação das cartas do aplicativo Memória Matemática	51
Listagem 16 - Trecho do código de posicionamento das cartas do aplicativo Memória Matemática	52
Listagem 17- Método chamado após click do usuário sob uma carta do aplicativo Memória Matemática	52
Listagem 18 - Método de verificação de par do aplicativo Memória Matemática	53

LISTA DE SIGLAS

AIR *Adobe Integrated Runtime*
CC *Creative Cloud*
IOS *iPhone Operating System*

SUMÁRIO

1 INTRODUÇÃO	11
1.1 CONSIDERAÇÕES INICIAIS.....	11
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral.....	12
1.2.2 Objetivos Específicos.....	12
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO	13
2 REFERENCIAL TEÓRICO.....	14
2.1 O ENSINO DA MATEMÁTICA POR MEIO DO LÚDICO	14
2.2 O USO DA TECNOLOGIA EM SALA DE AULA E A ADAPTAÇÃO DO SISTEMA DE ENSINO PARA AS NOVAS GERAÇÕES DE ALUNOS.....	14
3 MATERIAIS E MÉTODO.....	16
3.1 MATERIAIS.....	16
3.1.1 Adobe Animate CC.....	16
3.1.2 ANDROID.....	17
3.1.3 IOS	17
3.1.4 Action Script 3	17
3.1.5 Gliffy	17
3.2 MÉTODO	18
4 RESULTADOS.....	20
4.1 SPACE BINGO.....	20
4.1.1 Escopo.....	20
4.1.2 Modelagem	21
4.1.3 Apresentação	23
4.1.4 Implementação.....	28
4.1.5 Testes	32
4.2 DESAFIO DAS MEDIDAS	32
4.2.1 Escopo.....	32
4.2.2 Modelagem	33
4.2.3 Apresentação	35
4.2.4 Implementação.....	38
4.2.5 Implementação.....	42
4.3 MEMÓRIA MATEMÁTICA	42
4.3.1 Escopo.....	42
4.3.2 Modelagem	43
4.3.3 Apresentação	44
4.3.4 Implementação.....	50
4.3.5 Testes	53
5 CONCLUSÃO	54

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. Subsequentemente é feita a apresentação dos capítulos que compõem o texto.

1.1 CONSIDERAÇÕES INICIAIS

O ensino da matemática começa a ser introduzido no dia a dia escolar da criança a partir dos 4,5 anos, quando começam a ser elaboradas noções básicas de operações matemáticas, lógica e geometria. É muito importante que essa base seja sólida e bem trabalhada para auxiliar no aprendizado dos conhecimentos futuros. É de grande importância criar um ambiente que favoreça e estimule a vontade de aprender matemática. E os jogos e as brincadeiras têm papel muito importante nesse processo. Segundo Piaget (1967, p.25) “O jogo não pode ser visto apenas como brincadeira para gastar energia, pois ele favorece o desenvolvimento físico, cognitivo, afetivo e moral”. Por meio de jogos e brincadeiras a criança pode desenvolver de forma lúdica o raciocínio lógico-matemático e manifestar sua criatividade.

Neste trabalho serão desenvolvidos três jogos para dispositivos móveis, compatíveis com os sistemas operacionais IOS e Android. Os nomes escolhidos para eles são: *Space Bingo*, *Desafio das Medidas* e *Memória Matemática*.

O *Space Bingo* é um jogo de bingo adaptado para a matemática em que um número será sorteado e o usuário deverá encontrar em um painel uma multiplicação cujo resultado seja o número anteriormente sorteado, caso encontre deverá clicar sobre ela. O objetivo é completar uma linha ou uma coluna para assim vencer o jogo. O jogo contará com uma tela de configuração na qual o usuário poderá escolher os limites da multiplicação assim como o tempo que ele terá para encontrar a multiplicação.

O *Desafio das Medidas* é baseado em um *quiz* em que o usuário deverá clicar sobre um herói o mais rápido possível para que ele puxe uma carroça e a cada determinada distância uma pergunta deverá ser respondida. Essas perguntas são sorteadas pelo aplicativo e trabalham com a conversão de unidades de medidas

básicas como metro, centímetros, quilogramas e toneladas. Caso acerte a pergunta, o usuário ganha uma vantagem. Caso erre, a carroça retrocede para o ponto anterior. O objetivo é levar a carroça até o último ponto antes do inimigo, que é controlado pelo aplicativo. A dificuldade das perguntas e a velocidade do inimigo são configuráveis.

Memória Matemática é uma espécie de jogo da memória no qual o usuário deverá relacionar uma fração com o seu resultado numérico. Na tela de configuração do jogo poderá ser selecionado o número de cartas bem como se as imagens contidas nas cartas formarão pares ou não.

Para o desenvolvimento dos aplicativos a plataforma de desenvolvimento utilizada foi a Adobe Animate *Creative Cloud/Adobe Integrated Runtime (CC/AIR)* que oferece os recursos necessários para a construção do aplicativo. A codificação foi feita com a linguagem Action Script 3.0 que é a linguagem nativa da plataforma.

1.2 OBJETIVOS

A seguir serão apresentados o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Desenvolver um conjunto de jogos para auxílio do ensino da matemática básica.

1.2.2 Objetivos Específicos

- Trabalhar de maneira lúdica e intuitiva, por meio dos jogos desenvolvidos, questões como:
- Associação de números fracionários com números inteiros;
- Conversão de unidades de medidas;
- Operações básicas (adição, subtração, multiplicação e divisão).

1.3 JUSTIFICATIVA

Vivemos numa sociedade cercada de aparatos tecnológicos e o uso da tecnologia tem ganhado cada vez mais espaço na educação. Lima Junior (2007, p 67) destaca que: “Nossas escolas, que visam contribuir para que os indivíduos participem ativa e criticamente da dinâmica social, podem e devem investir na nova eficiência e competência, baseadas numa lógica do virtualizante”.

O modelo tradicional de ensino da matemática se baseia numa forma mecanizada de realização de exercícios, na memorização de fórmulas que muitas vezes não tem ligação com situações do cotidiano. A interação entre o aluno e o professor é reduzida, geralmente o aluno é apenas espectador em sala de aula. O uso de aplicativos pode ajudar na adequação do modelo de ensino a nova realidade da sociedade. Eles proporcionam uma forma mais pessoal e intuitiva de apresentar o conteúdo, além de promover a interação em sala de aula fazendo com que o aluno se sinta parte do processo.

Sendo assim, a proposta deste trabalho é desenvolver um conjunto de jogos que ofereça uma maneira intuitiva e atraente de ensinar matemática para crianças de 6 a 10 anos, por meio de atividades já presentes no cotidiano da criança e que possa ser usada tanto em sala de aula quanto em casa.

1.4 ESTRUTURA DO TRABALHO

Este trabalho foi organizado em 5 capítulos. No Capítulo 1 foram apresentadas as considerações iniciais, os objetivos e a justificativa do trabalho, no Capítulo 2 é apresentado o referencial teórico, sendo abordados os temas de ensino da matemática por meio do lúdico, o uso da tecnologia em sala de aula e a adaptação do sistema de ensino para as novas gerações de alunos. No Capítulo 3 são apresentados os métodos e os materiais utilizados no desenvolvimento do trabalho. No Capítulo 4 é apresentada a modelagem dos aplicativos, seus requisitos funcionais e não funcionais, os diagramas de caso de uso assim como o fluxograma de funcionamento dos jogos, telas dos jogos e as principais partes do código fonte. Por fim, o Capítulo 5 apresenta a conclusão.

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os conceitos que fundamentam o trabalho.

2.1 O ENSINO DA MATEMÁTICA POR MEIO DO LÚDICO

As atividades lúdicas têm um papel muito importante na educação infantil, pois elas trazem situações nas quais a criança deve tomar as decisões para chegar a um objetivo. Dessa maneira ela aprende a desenvolver diferentes estratégias perante o desafio proposto.

“Podemos dizer que o jogo serve como meio de exploração e invenção, reduz a consequência dos erros e dos fracassos da criança, permite que ela desenvolva sua iniciativa, sua autoconfiança, sua autonomia. No fundo, o jogo é uma atividade séria que não tem consequência frustrante para a criança.” (SMOLE, 1996, p. 138).

Para Kishimoto (1994), o jogo estimula a exploração e a solução de problemas e, por ser livre de pressão, cria um clima adequado para a investigação e a busca de soluções.

Pode-se observar que de forma geral existe certa resistência a aprender matemática, isso se dá pela forma com que a matemática pode ter sido apresentada nas etapas iniciais do aprendizado.

No ensino da matemática os jogos ajudam a criar o gosto pelos números e apresentam de forma mais interessante a associação de operações matemáticas às situações comuns do cotidiano. Isso ajuda a romper a resistência e a falta de interesse no aprendizado da matemática que tem sido observada nos alunos.

2.2 O USO DA TECNOLOGIA EM SALA DE AULA E A ADAPTAÇÃO DO SISTEMA DE ENSINO PARA AS NOVAS GERAÇÕES DE ALUNOS

A sociedade está diante de um grande questionamento de como tornar a escola mais atraente para os alunos. Segundo Prensky (2001b, p. 1) “nossos alunos mudaram radicalmente. Os alunos de hoje não são os mesmos para os quais o nosso sistema educacional foi criado”. Prensky (2001) define como “nativos digitais”

peças nascidas nos anos 2000. Essas pessoas cresceram cercadas pela mídia digital, desde cedo possuem contato com a internet, *smartphones*, *tablets*, videogames e redes sociais, ou seja, essas tecnologias já fazem parte de seu cotidiano.

Segundo Tapscot (1999), muitas escolas lutam com a realidade de alunos que dominam o espaço cibernético mais que o professor. No entanto, o uso das tecnologias digitais na educação não se trata mais de um diferencial e sim de uma necessidade. Não há como separar o processo de ensino e aprendizagem da disseminação de novas tecnologias.

Os nativos digitais não querem mais ser apenas observadores e ouvintes em sala de aula, eles sentem a necessidade de interagir e o *tablet*, usado juntamente com aplicativos educacionais, se apresenta como uma ferramenta para possibilitar essa interação. Para Rojo (2012, p.37) a presença das tecnologias digitais cria novas possibilidades de comunicação e expressão e assim como a tecnologia da escrita devem ser aprendidas. Segundo o Currículo em Movimento da Secretaria de Estado de Educação do Distrito Federal (BRASIL, 2013), o acesso às tecnologias digitais e a formação dos estudantes em torno dessas tecnologias são fundamentais e devem ser desenvolvidos.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e as tecnologias utilizadas para o desenvolvimento dos aplicativos e as principais atividades realizadas para desenvolvê-los.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias utilizadas no desenvolvimento do trabalho.

Ferramenta / Tecnologia	Versão	Finalidade
Adobe Animate CC	2015.2	Desenvolvimento dos desenhos e código fonte dos aplicativos.
Android	4.5	Sistema operacional alvo dos aplicativos.
IOS	9	Sistema operacional alvo dos aplicativos.
Gliffy		Criação dos fluxogramas e diagramas de casos de uso.
Action Script 3	3	Linguagem de programação.

Quadro 1 - Ferramentas utilizadas no desenvolvimento do trabalho

3.1.1 Adobe Animate CC

A Adobe Animate CC é uma plataforma desenvolvida pela empresa Adobe Systems Incorporated Inc., sucessora do Adobe Flash Professional, que oferece diversas ferramentas para o desenvolvimento de conteúdo multimídia. Essa plataforma conta com recursos para elaboração dos desenhos das animações e da codificação nas linguagens Actions Script 3.0 e JavaScript e oferece suporte para publicação do projeto para as plataformas *Desktop, Android e IOS*.

3.1.2 ANDROID

Atualmente o Android é a plataforma móvel mais utilizada no mundo, Segundo Lecheta (2013, p. 22) consiste de uma plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux, e que possui diversas aplicações já instaladas. Desenvolvida pela Google, ela teve a sua primeira versão publicada em 2009 com o nome de “Eclair”, atualmente se encontra na versão 7.0.

A plataforma Android é compatível com a maioria dos dispositivos presentes no mercado e sua licença é livre, o que permite customização.

3.1.3 IOS

O IOS é um sistema operacional desenvolvido pela Apple teve sua primeira versão publicada em 2007 que foi desenvolvida para o iPhone, posteriormente foi adaptada para os demais produtos da empresa. A Apple (2016) define o IOS como o “mais avançado do mundo”, seu desenvolvimento foi baseado do MAC OS X, sistema operacional utilizado pelos computadores da Apple.

3.1.4 Action Script 3

O Action Script 3 é uma linguagem de programação orientada a objetos que permite a criação de animações simples até aplicativos complexos. Foi lançada pela Adobe em 2006 juntamente com o Adobe Flash Player.

3.1.5 Gliffy

Gliffy é uma ferramenta *on-line* para criação de diagramas e fluxogramas. Oferece a possibilidade de salvar os diagramas criados na nuvem, podendo ser acessado de qualquer dispositivo.

3.2 MÉTODO

Para a modelagem e implantação dos aplicativos foi seguido o modelo sequencial linear. No Quadro 2 serão apresentados os fluxos de trabalho e as iterações.

Processos	1ª iteração	2ª iteração	3ª iteração
Requisitos	Levantamento dos requisitos necessários para o desenvolvimento dos aplicativos..	Complementação dos requisitos.	Revisão dos requisitos levantados na 1ª iteração.
Análise dos aplicativos	Definição da estrutura dos aplicativos, da apresentação do conteúdo e interação do usuário.	Definição do ciclo de vida.	Revisão das análises feitas na 1ª e na 2ª iteração.
Projeto	Definição das regras do jogo e do processo de geração das questões.	Ajustes nas regras do jogo e na geração das questões.	Revisão dos processos realizados na 1ª e na 2ª iteração.

Quadro 2 - Fluxos e iterações de trabalho

A seguir serão descritas mais detalhadamente cada uma das três etapas que compõem o desenvolvimento dos aplicativos. Essas etapas foram utilizadas para cada um dos três aplicativos desenvolvidos.

a) Requisitos

O processo de levantamento de requisitos procurou identificar os requisitos necessários para o desenvolvimento de um conjunto de aplicativos, visando prender a atenção do usuário, que apresentem o conteúdo de forma didática e intuitiva e sejam compatíveis com as plataformas *Android* e *IOS*. Os requisitos foram levantados por meio de pesquisa em artigos e aplicativos semelhantes e pela experiência do autor em já ter atuado profissionalmente na área de desenvolvimento de aplicações educacionais. Com os dados coletados foi gerada a primeira versão dos requisitos.

b) Análise dos aplicativos

Na etapa da análise, com base no levantamento dos requisitos, foi definida a estrutura dos aplicativos, a forma com que o conteúdo será apresentado e como o usuário interagirá. Nessa etapa também foram definidos os personagens e o ciclo de vida de cada um dos aplicativos.

c) Projeto

Após a análise concluída e revisada foram definidas as regras do jogo e do

processo de geração das questões que serão apresentadas para o usuário durante o jogo.

d) Implementação

O processo de implementação seguiu os requisitos definidos nas fases anteriores. Esse processo foi composto por, basicamente, duas fases: a) desenho dos personagens, objetos, menus e ambiente, b) codificação das regras e da interação do aplicativo.

e) Testes

Os testes foram conduzidos em paralelo à implementação, mas de forma mais acentuada após a sua conclusão. Os testes foram realizados utilizando o modo de *debug* da ferramenta Adobe Animate e um *tablet* com o sistema operacional Android.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos com o desenvolvimento deste trabalho, são apresentados trechos do código dos aplicativos assim como imagens das telas.

4.1 SPACE BINGO

O aplicativo Space Bingo é baseado em um jogo de bingo e o seu principal objetivo é oferecer uma ferramenta para auxiliar no ensino da operação de multiplicação.

4.1.1 Escopo

É um jogo de bingo no qual a cartela do usuário é composta por multiplicações e não por números, como tradicionalmente ocorre. A cada rodada um número é sorteado e o usuário deve procurar em seu painel se existe alguma multiplicação cujo resultado seja o número sorteado. Se encontrar, o usuário deve marcar a multiplicação clicando sobre ela. Ao marcar o aplicativo verifica se já existe alguma linha ou coluna completas, caso exista o usuário vence o jogo. Paralelamente à jogada do usuário, o aplicativo simula a ação de outro jogador adversário que também terá uma cartela gerada aleatoriamente e, da mesma forma, o aplicativo busca na cartela do adversário se existe alguma multiplicação correta.

Para que o jogo se torne mais interativo e atraente foram definidas as seguintes regras e funcionalidades:

- O usuário tem um tempo pré-determinado a cada rodada para encontrar a multiplicação, caso não encontre dentro desse tempo, um novo número é sorteado e uma nova rodada é iniciada.
- Simultaneamente à jogada do usuário, o aplicativo simula a jogabilidade de um adversário. Um indicador do progresso de conclusão da cartela do adversário é exibido na tela principal do jogo

com o propósito de aumentar a competitividade.

- O adversário simulado pelo aplicativo também tem a possibilidade de vencer o jogo. Se isso acontecer uma animação é exibida caracterizando que o adversário venceu o desafio.
- As regras de interação do adversário são ajustadas para que ele possua um determinado percentual de erro ao encontrar a multiplicação. Essa funcionalidade simula a possibilidade do usuário não encontrar a multiplicação dentro do tempo determinado da rodada.
- O aplicativo conta com uma tela de configuração, na qual podem ser escolhidos os multiplicadores máximos e mínimos que o aplicativo utilizará para criação das cartelas e dos números a serem sorteados.

Não será necessária conexão com a internet para o funcionamento do aplicativo.

4.1.2 Modelagem

O Quadro 3 apresenta a listagem dos requisitos funcionais identificados para o aplicativo Space Bingo.

Identificação	Descrição
[RF-01] Configurações de jogo	O aplicativo conterà uma tela para configuração do jogo, podendo ser configurados os multiplicadores máximos e mínimos e o tempo de cada rodada.
[RF-02] Jogabilidade	O aplicativo garantirá que sejam sorteados apenas números possíveis de serem obtidos pelas multiplicações realizadas considerando os limites definidos na tela de configuração.
[RF-03] Conclusão do jogo	O aplicativo finalizará o jogo sempre que o usuário ou o adversário completarem uma linha ou uma coluna de sua cartela. Deverá exibir a animação respectiva a quem ganhou o jogo e exibir uma tela pedindo se o usuário deseja iniciar um novo jogo.
[RF-04] Botão pular	Um botão deve ser adicionado, na tela principal de jogo, para que o usuário possa pular a rodada caso não possua nenhuma multiplicação em sua cartela que satisfaça o número sorteado.
[RF-05] Adversário	O aplicativo simulará a jogabilidade de um adversário, contando com uma taxa de erro para que o adversário “não encontre” em sua cartela a multiplicação que satisfaça o número sorteado.
[RF-06] Tela de ajuda	O aplicativo conterà uma tela com instruções de como jogar. Essas instruções devem estar disponíveis na tela inicial do aplicativo e deve ser acessível em qualquer momento do jogo.

Quadro 3 - Space Bingo - Requisitos Funcionais

O Quadro 4 apresenta a listagem dos requisitos não funcionais identificados

para o aplicativo Space Bingo.

Identificação	Descrição
[NF-01] Ambientação	O jogo será ambientado no espaço, com efeitos sonoros e objetos que remetam a planetas, estrelas e naves espaciais.
[NF-02] Interface	A interface será adequada para uso em dispositivos móveis, contando com botões grandes e pouco texto.
[NF-03] Orientação	A orientação do aplicativo será na horizontal.

Quadro 4 - Space Bingo - Requisitos não funcionais

O diagrama de casos de uso é representado pela Figura 1.

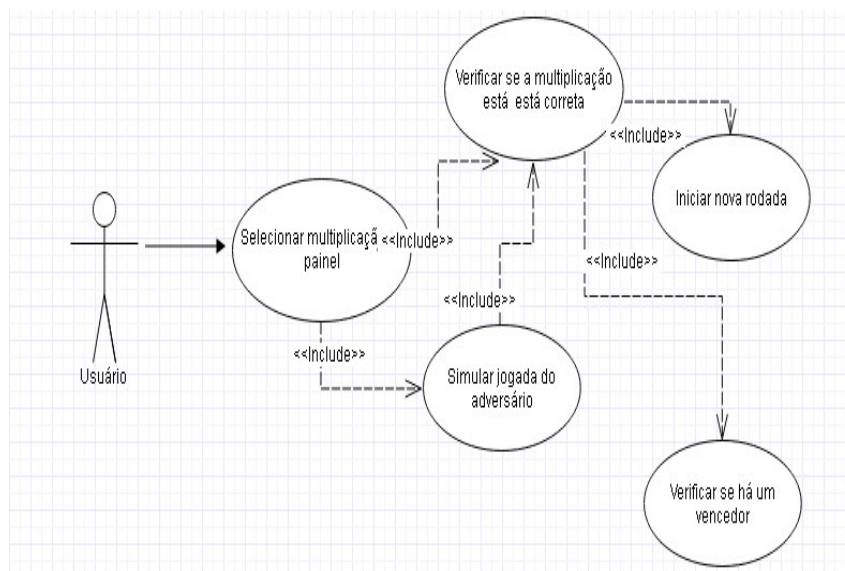


Figura 1 - Diagrama de casos de uso do aplicativo Space Bingo

Na Figura 2 está representado o fluxograma do aplicativo.

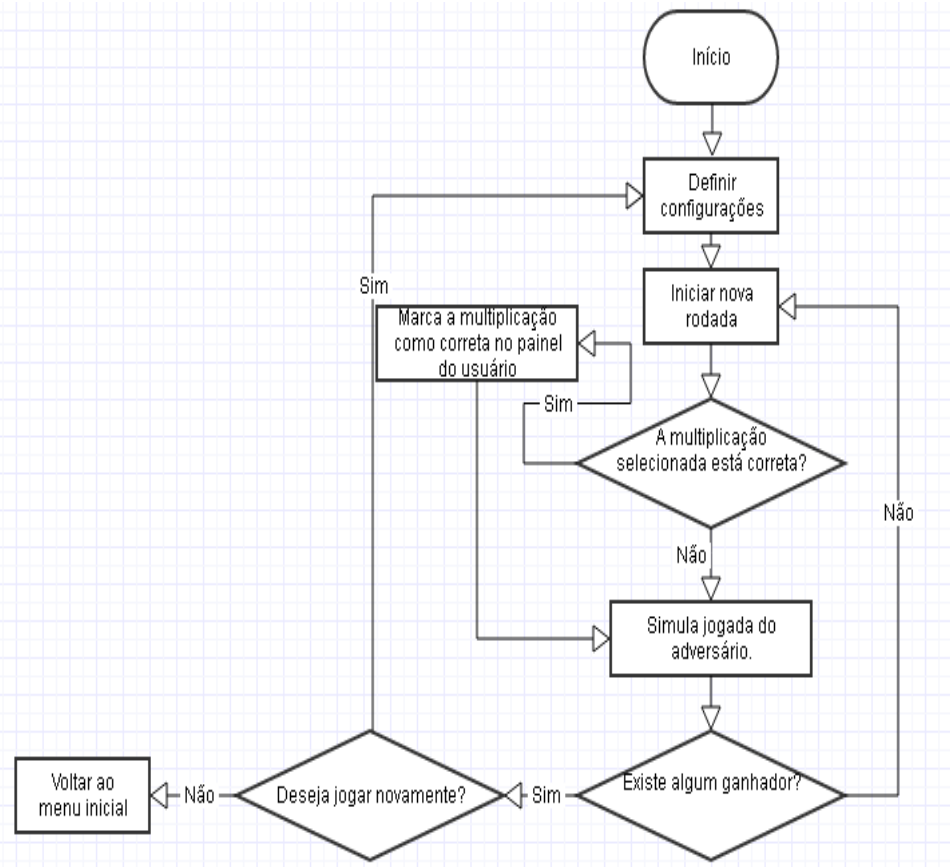


Figura 2 - Fluxograma do aplicativo Space Bingo

4.1.3 Apresentação

A Figura 3 representa a tela de configurações do jogo. Nessa tela, o usuário pode definir os multiplicadores máximo e mínimo e o tempo de cada rodada. Os limites definidos pelo usuário são utilizados pelo jogo para criar as multiplicações apresentadas na cartela do usuário. O tempo informado é o tempo máximo que o usuário tem para marcar a multiplicação na cartela, passado esse tempo uma nova rodada é iniciada.

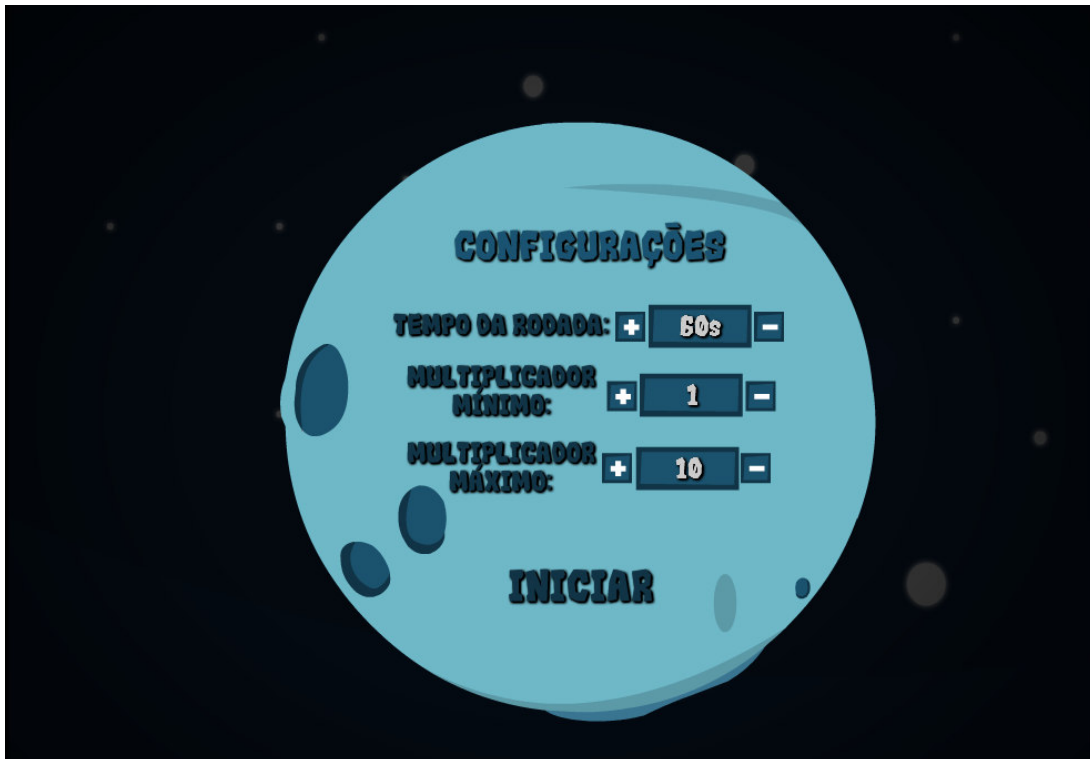


Figura 3 - Tela de configurações do aplicativo Space Bingo

Na Figura 4 é apresentada a tela principal do aplicativo, na qual é possível clicar sobre as multiplicações apresentadas no painel do usuário. A tela conta ainda com o botão “Pular” que reinicia o tempo e sorteia um novo número.

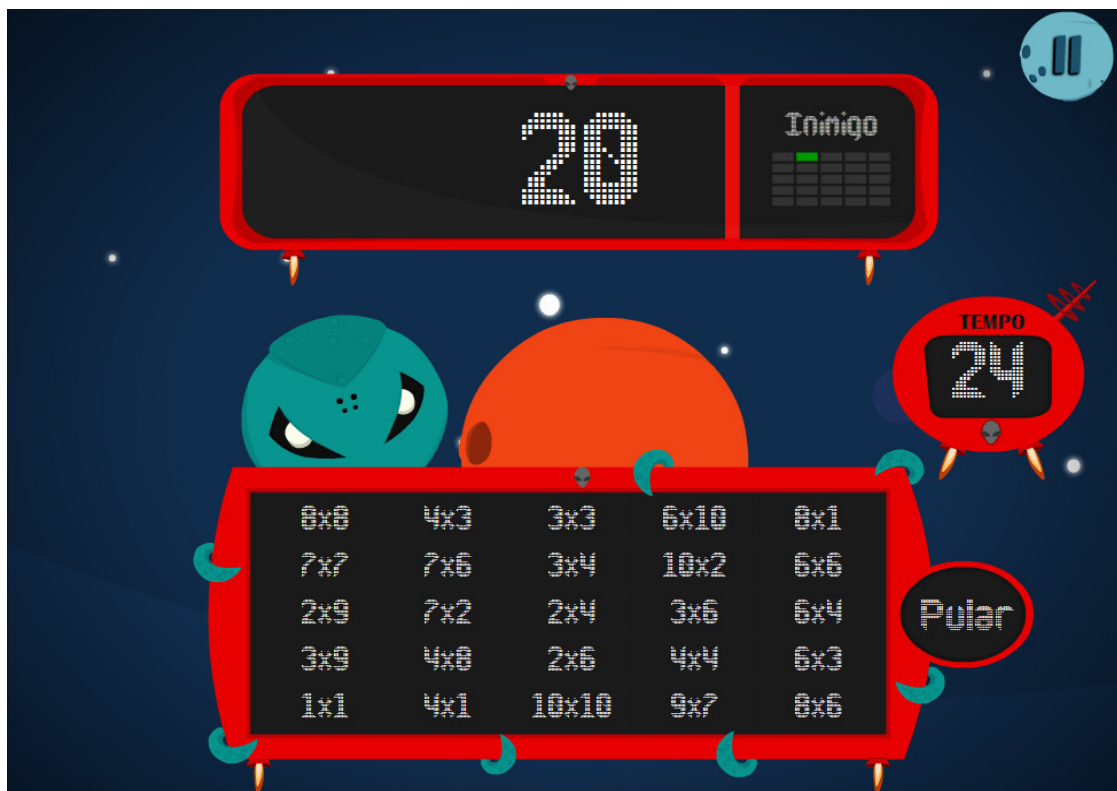


Figura 4 - Tela principal do aplicativo Space Bingo

Na Figura 5, pode-se observar o progresso de jogo do adversário. Ao final de cada rodada o aplicativo verifica se existe alguma multiplicação presente na cartela do adversário. Para tornar a jogabilidade mais justa foram implementadas regras de interação que definem uma chance de 20% da multiplicação mesmo estando correta não ser marcada. Essa regra foi implementada a fim de simular a situação na qual o usuário não encontra a multiplicação na cartela antes do tempo de rodada terminar.

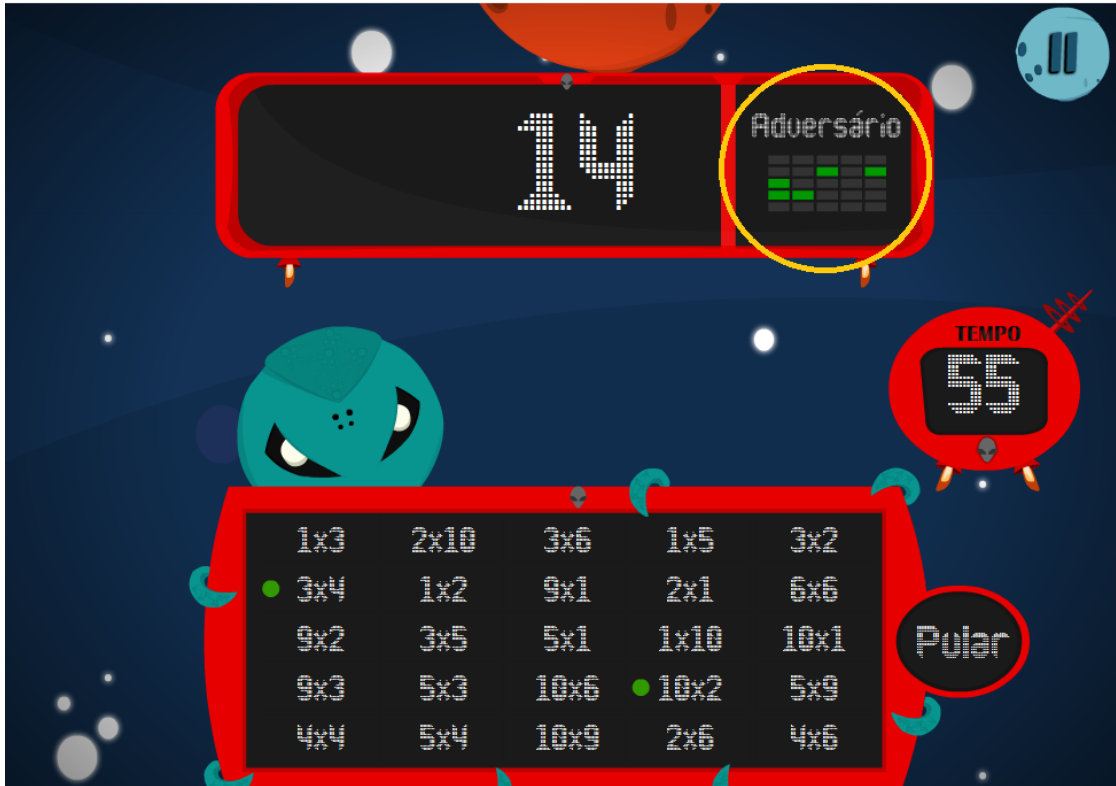


Figura 5 - Tela principal do aplicativo Space Bingo com a cartela do adversário em destaque

Na Figura 6 é possível visualizar a animação mostrada em tela caso o adversário vença o jogo.

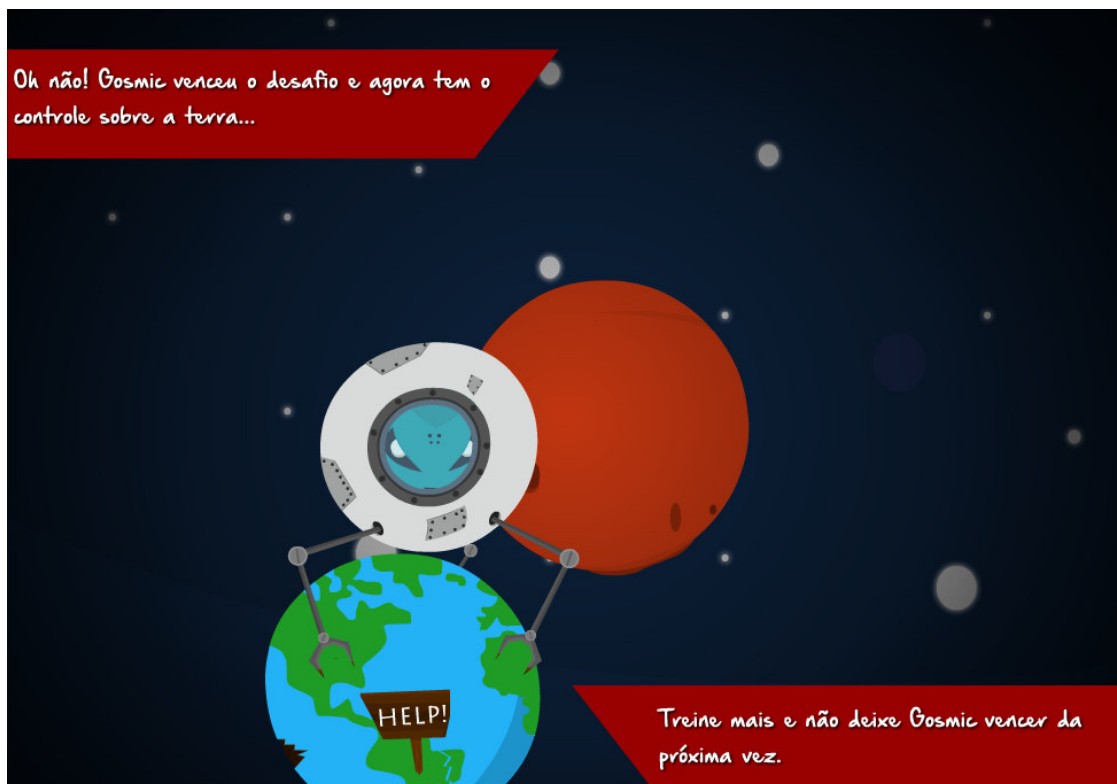


Figura 6 - Animação apresentada em tela após a vitória do adversário.

Após o fim da animação uma tela pedindo se o usuário deseja jogar novamente é apresentada, conforme ilustrado na Figura 7.



Figura 7 - Tela apresentada ao final do jogo do aplicativo Space Bingo

4.1.4 Implementação

São apresentados a seguir alguns trechos da codificação do aplicativo, assim como a estrutura do projeto e seus arquivos principais.

Na Figura 8 é possível observar a estrutura de pastas e arquivos do aplicativo. Na pasta “code” são armazenados os arquivos do tipo ActionScript que contém a codificação do aplicativo e na pasta “sons” são armazenados os arquivos de som executados durante o jogo. O arquivo “SpaceBingo.apk” é gerado pelo Adobe Animate ao exportar a aplicação para Android, esse arquivo é utilizado para a instalação no dispositivo.

code	15/09/2015 12:34	Pasta de arquivos	
release	16/09/2015 12:36	Pasta de arquivos	
sons	15/09/2015 12:56	Pasta de arquivos	
Bingo.apk	16/09/2015 08:30	Arquivo APK	668 KB
Bingo	23/05/2017 21:41	Documento do A...	7.874 KB
Bingo	16/09/2015 12:38	Chrome HTML Do...	2 KB
Bingo	23/05/2017 21:54	Flash Movie	871 KB
Bingo-app	23/05/2017 21:54	Documento XML	2 KB
Certificado de publicação Android	16/09/2015 08:29	Troca de Informaç...	2 KB
Space Bingo.apk	18/09/2015 08:45	Arquivo APK	10.012 KB

Figura 8 - Estrutura de pastas e arquivos do aplicativo Space Bingo

Na Listagem 1 está a classe principal do projeto. Nessa classe é feita a declaração do objeto “interacao” que faz o controle geral do aplicativo, assim como o objeto “menus” que controla a exibição de todas as janelas do jogo. Nessa listagem é realizada a declaração do objeto “timerSplash” com duração de quatro segundos. Com a chamada “timerSpash.start()” o timer é disparado e por meio do evento “TIMER_COMPLETE” o método de inicialização do app “iniciarApp” é executado. Então é chamada a janela principal do aplicativo e é iniciada a execução do som ambiente em *loop*.

```

package code{

import flash.display.MovieClip;
import lib.utils.Audio;
import flash.utils.Timer;
import flash.events.TimerEvent;

public class BingoMain extends MovieClip
{
public var interacao:Interacao;
public var menus:Menus;
public var dirBase:String="";
private var timerSplash:Timer = new Timer(4000,1);

public function BingoMain()
{
interacao = _interacao;
menus = _menus;
interacao.live(this);

interacao.visible = false;

menus.live(this);

timerSplash.addEventListener(TimerEvent.TIMER_COMPLETE, iniciarApp);
timerSplash.start();
//iniciarApp();
}

private function iniciarApp(e:TimerEvent):void
{
//chamada da função iniciarJogo() temporaria aqui, depois quem irá chamar será a classe do menu
_splash.visible=false;
menus.abrirJanelaPrincipal();
Audio.loopSom(dirBase+"sons/ambiente1.mp3");
}

}
}

```

Listagem 1 - Classe principal do aplicativo Space Bingo

A Listagem 2 apresenta a declaração dos objetos de todos os menus do jogo.

```

11
12 public class Menus extends MovieClip
13 {
14     //OBJETOS DE CONTROLE
15     public var owner:BingoMain;
16
17     //OBJETOS
18     private var janelaAtual:*;
19     private var janelaPrincipal:JanelaPrincipal;
20     private var janelaConfiguracoes:JanelaConfiguracoesJogo;
21     private var janelaConfirmaSaida:JanelaConfirmaSaida;
22     private var janelaPausa:JanelaPause;
23     private var janelaJogarNovamente:JanelaJogarNovamente;
24     private var janelaAjuda:JanelaAjuda;
25     private var janelaSobre:JanelaSobre;

```

Listagem 2 - Declaração dos objetos de menu do aplicativo Space Bingo

O método `abrirJanelaConfiguracoes()` é responsável por instanciar um novo objeto do tipo `JanelaConfiguracoesJogo` e adicioná-lo na tela por meio do método nativo `addChild()`. Esses dois métodos são exibidos na Listagem 3.

```

54     public function abrirJanelaConfiguracoes():void
55     {
56         janelaConfiguracoes = new JanelaConfiguracoesJogo();
57         janelaConfiguracoes.live(this);
58         janelaAtual = janelaConfiguracoes;
59         addChild(janelaConfiguracoes);
60     }
61

```

Listagem 3 - Método de abertura da janela de configurações do aplicativo Space Bingo

Na Listagem 4 está o método criarGridRespostas() que é responsável por criar o *grid* de respostas do jogador. A criação é realizada a partir dos limites máximo e mínimo definidos para o multiplicador, na tela de configuração do jogo. Nesse método são definidos o número de linhas e de colunas, a altura e a largura de cada objeto e a posição x e y inicial do *grid*. Para cada posição é criado um novo objeto do tipo btnOpcaoResposta, que é adicionado na lista arrBtnOpcaoResposta.

```

123     private function criarGridRespostas():void
124     {
125         var auxPosicaoGrid:Number=0;
126         btnOpcaoResposta = new BtnOpcaoResposta();
127
128         arrBtnOpcoesResposta = new Array();
129
130         matrizOpcoesRespostas = new Array();
131         var contLinha:Number=0;
132         var contColuna:Number=0;
133
134         gridOpcoesResposta.linhas=5;
135         gridOpcoesResposta.colunas=5;
136         gridOpcoesResposta.alturaTotal=btnOpcaoResposta.height*gridOpcoesResposta.linhas+20;
137         gridOpcoesResposta.larguraTotal=btnOpcaoResposta.width*gridOpcoesResposta.colunas+20;
138         gridOpcoesResposta.x0=(this.width-(gridOpcoesResposta.larguraTotal))/2.6;
139         gridOpcoesResposta.y0=(this.height-(gridOpcoesResposta.alturaTotal+(gridOpcoesResposta.alturaTotal/2.5)));
140         gridOpcoesResposta.gerarPosicoes();
141
142         for (var i:Number=0;i<gridOpcoesResposta.getTotalPosicoes();i++)
143         {
144             var posicaoGrid:Point;
145
146             posicaoGrid = gridOpcoesResposta.vPontosVazias[auxPosicaoGrid];
147             btnOpcaoResposta = new BtnOpcaoResposta();
148             btnOpcaoResposta.x = posicaoGrid.x;
149             btnOpcaoResposta.y = posicaoGrid.y;
150             btnOpcaoResposta.addEventListener(MouseEvent.CLICK, getResposta);
151
152             arrBtnOpcoesResposta.push(btnOpcaoResposta);
153             addChild(btnOpcaoResposta);
154             auxPosicaoGrid++;
155         }
156
157         criarMatrizRespostas();
158         criarListaResultadosPossiveis();
159     }

```

Listagem 4 - Método de criação da cartela de respostas do aplicativo Space Bingo

A Listagem 5 apresenta o método criarListaResultadosPossiveis que armazena todos os resultados possíveis da multiplicação dos multiplicadores máximo e mínimo definidos na tela de configuração do jogo. Os resultados são armazenados nas listas arrRespostas e arrRespostasInimigo para posteriormente serem utilizadas no sorteio da matriz de resposta do usuário e do adversário.


```

161 private function criarListaResultadosPossiveis():void
162 {
163     for(var i:Number=multiplicadorMinimo;i<=multiplicadorMaximo;i++)
164     {
165         for(var j:Number=multiplicadorMinimo;j<=multiplicadorMaximo;j++)
166         {
167             var result:Number=i*j;
168             if(arrResultadosPossiveis.indexOf(result)==-1)
169             {
170                 arrResultadosPossiveis.push(result);
171             }
172             arrRespostas.push(i.toString() + "x" + j.toString());
173             arrRespostasInimigo.push(i.toString() + "x" + j.toString());
174         }
175     }
176
177     //trace("ArrResultadosPossiveis: " + arrResultadosPossiveis);
178     //trace("Total resultados possiveis: " + arrResultadosPossiveis.length);
179     //trace("ArrOpcoesResposta: " + arrRespostas);
180     criarListaRespostas();
181 }

```

Listagem 5 - Método de criação dos resultados possíveis das multiplicações do aplicativo Space Bingo

4.1.5 Testes

Os testes do aplicativo Space Bingo aconteceram durante a sua implementação. Os testes foram realizados utilizando o modo *debug* da ferramenta *Adobe Animate*. Foram realizados testes em um *tablet* com o sistema operacional Android. Os testes foram focados em encontrar erros na criação das matrizes de resposta e na simulação de jogo do adversário, também foram observados pontos como a usabilidade e as animações do jogo.

4.2 DESAFIO DAS MEDIDAS

O Aplicativo Desafio das Medidas é baseado em um Quiz. Esse jogo tem como principal objetivo trabalhar com a associação e a conversão de unidades básicas de medida

4.2.1 Escopo

O jogo simula uma espécie de cabo de guerra, no qual o usuário deverá clicar sobre o seu personagem para que ele puxe uma carroça até determinado ponto. No percurso existem 4 divisões, no momento em que a carroça alcançar essas divisões o usuário deverá responder uma pergunta relacionada à conversão de medidas gerada aleatoriamente pelo aplicativo. Simultaneamente, o aplicativo simulará a jogabilidade de um adversário que terá uma chance de erro previamente definida

para responder as perguntas. Para vencer o jogo o usuário deverá trazer a sua carroça até o final do percurso mais rápido que o adversário.

Foram definidas as seguintes regras e funcionalidades para o jogo:

- A cada erro nas respostas o usuário sofrerá uma desvantagem relacionada à unidade de medida que era assunto da pergunta, por exemplo: caso o usuário ou o adversário erre uma pergunta sobre peso, a carroça se tornará “mais pesada” avançando menos no cenário a cada clique na tela.
- O aplicativo contará com uma tela de seleção de dificuldade de jogo que interferirá na velocidade com que o adversário puxa a carroça e a chance que ele tem de errar as respostas das perguntas.

4.2.2 Modelagem

O Quadro 5 apresenta a listagem dos requisitos funcionais identificados para o aplicativo Desafio das Medidas.

Identificação	Descrição
[RF-01] Configurações de jogo	O aplicativo conterà uma tela para configurar a dificuldade de jogo.
[RF-02] Jogabilidade	O aplicativo garantirá que a carroça avance a mesma distância a cada clique do usuário.
[RF-03] Conclusão do jogo	O aplicativo finalizará o jogo sempre que o usuário ou o adversário puxarem a carroça até a linha final, apresentando uma animação referente a quem venceu o jogo e uma tela pedindo se o usuário deseja iniciar uma nova partida.
[RF-04] Perguntas	O aplicativo exibirá uma pergunta gerada aleatoriamente a cada vez que a carroça atingir as marcações determinadas no cenário de jogo. As perguntas devem conter 2 respostas incorretas e 1 resposta correta. O usuário terá duas chances de responder corretamente.
[RF-05] Desvantagem	Sempre que houver um erro ao responder as perguntas o aplicativo aplicará uma desvantagem ao usuário. Essa desvantagem deve ser relacionada ao tipo da pergunta respondida, como peso e distância.
[RF-06] Tela de ajuda	O aplicativo conterà uma tela com instruções de como jogar que deve estar disponível na tela inicial do aplicativo e ser acessível a qualquer momento do jogo.

Quadro 5 - Desafio das Medidas - Requisitos funcionais

O Quadro 6 apresenta a listagem dos requisitos não funcionais identificados para o aplicativo Desafio das Medidas.

Identificação	Descrição
[NF-01] Ambientação	O jogo será ambientado em uma arena de gladiadores.
[NF-02] Interface	A interface será adequada para uso em dispositivos móveis, contando com botões grandes e pouco texto.
[NF-03] Orientação	A orientação do aplicativo será na horizontal.

Quadro 6 - Desafio das Medidas - Requisitos não funcionais

A Figura 9 mostra o diagrama de casos de uso.



Figura 9 - Diagrama de casos de uso do aplicativo Desafio das Medidas

Na Figura 10 está representado o fluxograma do aplicativo Desafio das Medidas.

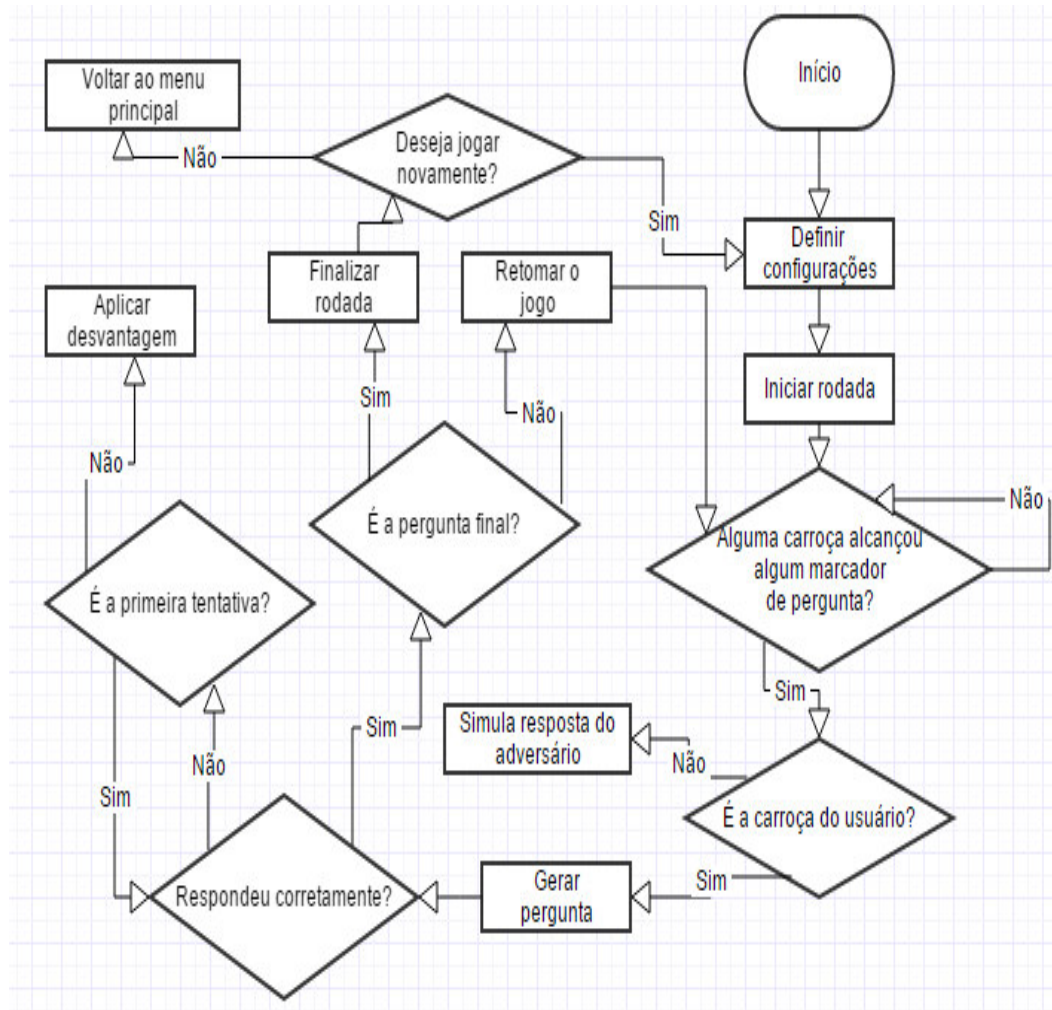


Figura 10 - Fluxograma do aplicativo Desafio das Medidas

4.2.3 Apresentação

Na Figura 11 é exibida a tela de configurações do aplicativo Desafio das Medidas, sendo possível selecionar a dificuldade de jogo. A dificuldade interferirá na velocidade em que o adversário movimentará a sua carroça e a chance que ele tem de errar a pergunta simulada, no nível fácil a chance é de 70%, no nível médio é de 60% e no nível difícil é de 50%.



Figura 11 - Tela de configurações do aplicativo Desafio das Medidas

A Figura 12 representa a tela de perguntas por meio da qual o usuário realiza a conversão proposta. O usuário tem duas chances de responder corretamente, após o primeiro erro a opção selecionada erroneamente é removida da tela, restando somente as outras duas, conforme pode ser observado na Figura 13.

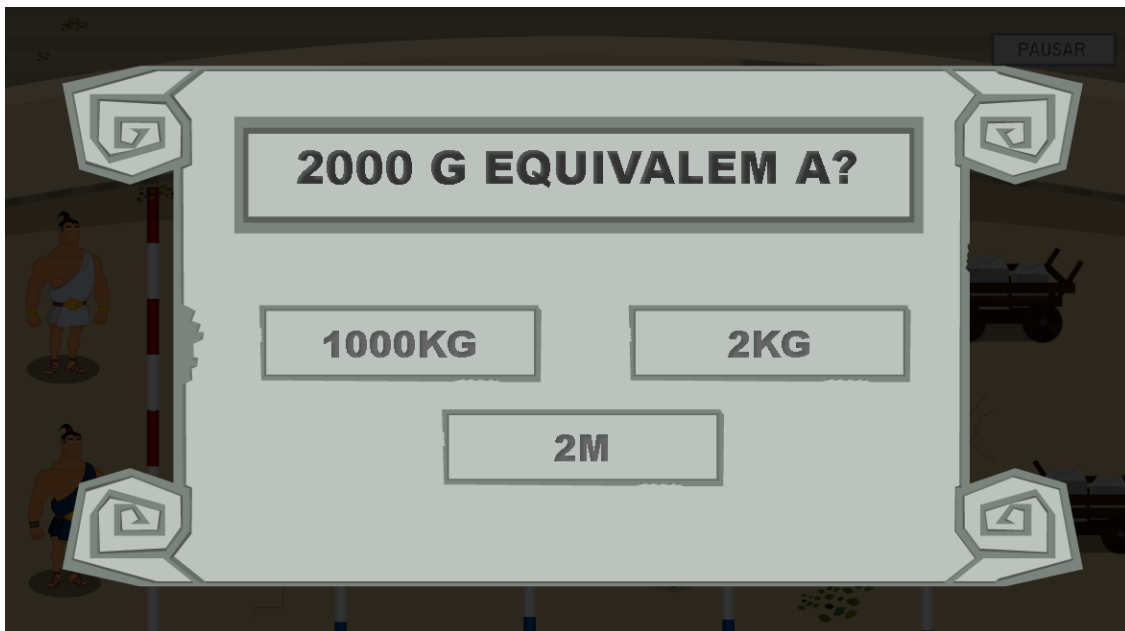


Figura 12 - Tela de pergunta do aplicativo Desafio das Medidas

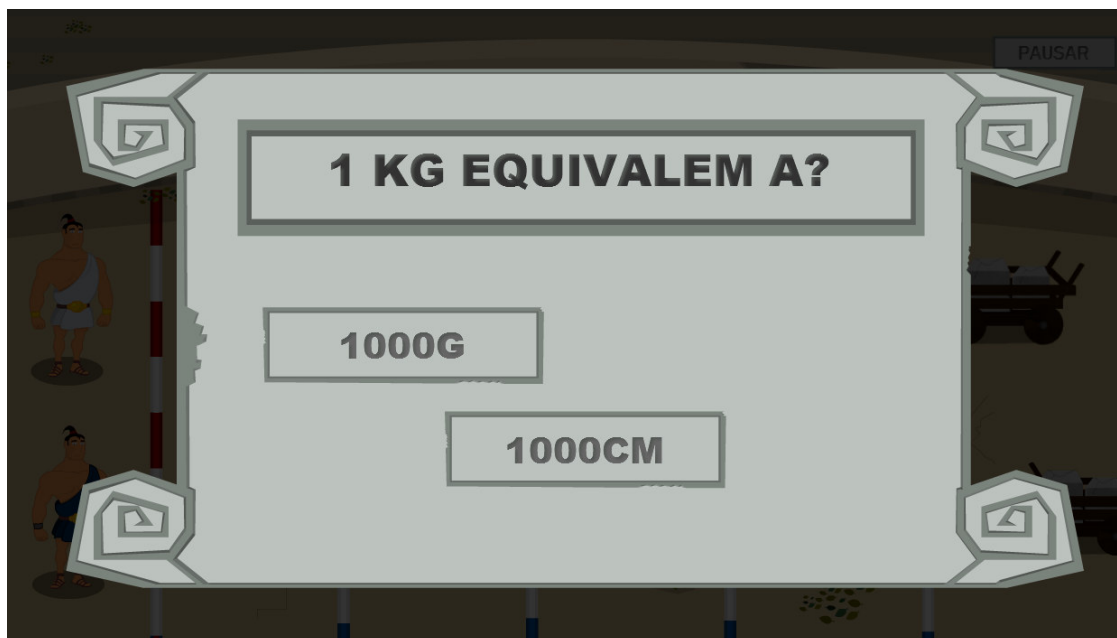


Figura 13 - Tela de perguntas após uma resposta errada do aplicativo Desafio das Medidas

Na Figura 14 pode-se visualizar a contagem regressiva que é apresentada na tela após o usuário responder a pergunta. Essa funcionalidade foi implementada para que o jogador tenha tempo de se preparar para a continuação do jogo e tem a duração de 3 segundos.



Figura 14 - Tela de contagem regressiva do aplicativo Desafio das Medidas

A tela principal do jogo pode ser visualizada na Figura 15. Nela é possível

observar os dois personagens, o do usuário (em branco) e o adversário (em azul). No lado oposto aos personagens estão as duas respectivas carroças. O usuário deverá clicar sobre o seu personagem a fim de que a carroça se movimente. As faixas listradas em branco e azul identificam os pontos em que o usuário responderá uma pergunta. A faixa listrada em branco e vermelho representa o ponto final do jogo.

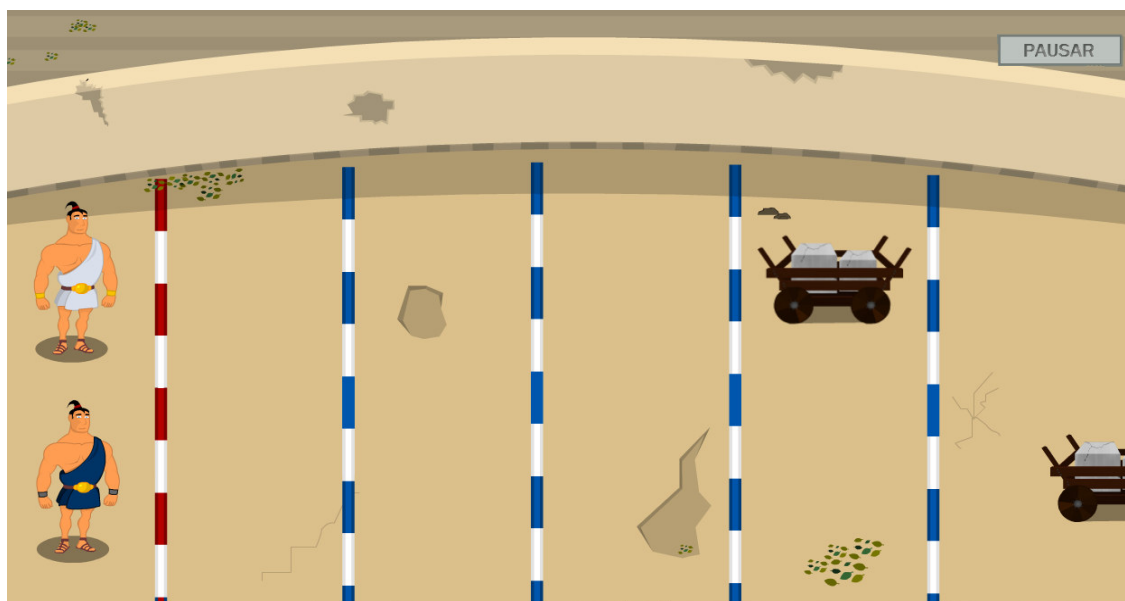


Figura 15 - Tela principal do aplicativo Desafio das Medidas

4.2.4 Implementação

A seguir serão apresentados alguns trechos da codificação do aplicativo Desafio das Medidas assim como a estrutura de pastas e arquivos do projeto. A Figura 16 exibe os arquivos principais do projeto. Na pasta “code” estão os arquivos do tipo ActionScript que contém a codificação do jogo, o arquivo “Desafio das Medidas.fla” é o arquivo principal do projeto.

Nome	Data de modificaç...	Tipo	Tamanho
code	03/11/2015 00:47	Pasta de arquivos	
DesafioDasMedidas	02/06/2017 23:23	Documento do A...	1.632 KB
DesafioDasMedidas	04/04/2016 15:09	Chrome HTML Do...	3 KB
DesafioDasMedidas	02/06/2017 19:52	Flash Movie	268 KB
DesafioDasMedidas	19/10/2015 22:30	Documento XML	1 KB
DesafioDasMedidas-app	04/04/2016 15:09	Documento XML	2 KB

Figura 16 - Estrutura de pastas e arquivos do aplicativo Desafio das Medidas

Na Listagem 6 é exibido o código da classe principal do aplicativo. Pode-se observar a declaração do objeto timerSplash do tipo Timer. Por meio da chamada

timerSplash.start() o timer é iniciado, e ao fim de 1,5 segundos o método iniciar() é executado chamando a janela principal do aplicativo.

```

3      import flash.display.MovieClip;
4      import flash.events.MouseEvent;
5      import flash.utils.Timer;
6      import flash.events.TimerEvent;
7
8
9      public class DesafioDasMedidasMain extends MovieClip
10     {
11         //Variáveis de controle
12         public var interacao:Interacao;
13         public var menus:Menus;
14
15         public function DesafioDasMedidasMain()
16         {
17             trace("Desafio das medidas main");
18             menus = _menus;
19             menus.live(this);
20
21             interacao = _interacao;
22             interacao.live(this);
23
24             var timerSplash:Timer = new Timer(1500,1);
25             timerSplash.addEventListener(TimerEvent.TIMER_COMPLETE, function(){
26                 _splash.visible = false;
27                 iniciar();
28             });
29
30             timerSplash.start();
31         }
32
33         private function iniciar():void
34         {
35             menus.abrirJanelaPrincipal();
36         }
37     }
38
39 }

```

Listagem 6 - Classe principal do aplicativo Desafio das Medidas

A Listagem 7 representa o método configurarJogo(), o qual define a chance de erro e a velocidade do adversário conforme configuração de dificuldade selecionada pelo usuário na tela de configuração do jogo. As variáveis chanceErroInimigo e velPadraoInimigo são valorizadas para posteriormente serem utilizadas na interação do jogo.

```

108     private function configurarJogo()
109     {
110         carrocaJogador = new Carroca();
111
112         if(dificuldade=="facil")
113         {
114             trace("Fácil");
115             chanceErroInimigo = 70;
116             velPadraoInimigo = 1.3;
117         }
118
119         else if(dificuldade=="medio")
120         {
121             trace("Médio");
122             chanceErroInimigo = 60;
123             velPadraoInimigo = 1.5;
124         }
125
126         else
127         {
128             trace("Difícil");
129             chanceErroInimigo = 50;
130             velPadraoInimigo = 1.9
131         }

```

Listagem 7 - Trecho do método de configuração de jogo do aplicativo Desafio das Medidas

Para iniciar o movimento da carroça é necessário chamar o método puxarCarroca() que é disparado por meio do clique do usuário no personagem, o código desse método é exibido na Listagem 8. Nesse método, a animação da roda

da carroça é iniciada pela chamada do método `carrocaJogador_roda.play()`. A movimentação da carroça é realizada por meio da manipulação da propriedade `x` do objeto `carrocaJogador`. Nesse método também é feita a verificação da colisão entre a carroça do jogador e os marcadores da tela.

```

163     private function puxarCarroca(e:MouseEvent)
164     {
165         //trace("Puxar carroca");
166         if(carrocaJogador._roda.isPlaying==false)
167         {
168             carrocaJogador._roda.play();
169             carrocaJogador._rodaTras.play();
170         }
171         carrocaJogador.x=carrocaJogador.getDeslocamento();
172         for(var i:Number=0;i<arrMarcadoresDesafioJogador.length;i++)
173         {
174             if((carrocaJogador._hitador.hitTestObject(arrMarcadoresDesafioJogador[i])) &&
175                |(carrocaJogador.getHitHabilitado()))
176             {
177                 iniciarDesafio(null);
178                 arrMarcadoresDesafioJogador.shift();
179             }
180         }
181
182         if(carrocaJogador._hitador.hitTestObject(_marcadorFinal))
183         {
184             finalizarJogoHitLinhaChegada(1);
185         }
186     }
187

```

Listagem 8 - Método de movimentação da carroça do usuário do aplicativo Desafio das Medidas

A Listagem 9 representa o método `iniciarDesafio()` que é executado no momento em que a carroça do usuário atinge algum dos marcadores da tela. É nesse método que é realizado o sorteio de qual unidade de medida será utilizada para formular a pergunta. As unidades já foram previamente divididas por grandeza e armazenadas em listas, como pode ser observado na Listagem 10. As respostas da pergunta serão invariavelmente 1 correta e 2 incorreta.

```

218     public function iniciarDesafio(e:Event)
219     {
220         var auxSortUnidade:Number = Math.floor(Math.random()*3);
221         var auxSortValorUnidade:Number = Math.floor(Math.random()*10);
222         auxSortValorUnidade++;
223         var auxSortUnidade2:Number = Math.floor(Math.random()*2);
224         tipoPergunta = auxSortUnidade2;

```

Listagem 9 - Método de inicialização do desafio do aplicativo Desafio das Medidas

```

28     private var arrUnidadesMajores:Array = new Array("KM","TON");
29     private var arrUnidadesMedias:Array = new Array("M","KG");
30     private var arrUnidadesMenores:Array = new Array("CM","G");

```

Listagem 10 - Declaração das lista das unidades de medida do aplicativo Desafio das Medidas

A Listagem 11 apresenta o método `onClickResposta()` que é chamado pelo *click* do usuário em uma das alternativas de resposta da pergunta apresentada. Nesse método é verificado se a resposta escolhida está correta ou incorreta, caso esteja incorreta o aplicativo verifica se é a primeira tentativa. Caso seja, o jogo

permite mais uma tentativa, removendo a tentativa escolhida. Caso seja a segunda tentativa, a tela da pergunta é fechada e o jogo continua. Então uma penalidade é aplicada ao jogador, referente ao tipo de pergunta que foi respondida.

```

46     private function onClickResposta(e:MouseEvent):void
47     {
48         strRespostaUsuario = Botao(e.target).getTextoBotao();
49
50         if(strRespostaUsuario==strRespostaCorreta)
51         {
52             acertou=true;
53             _iconeResposta.gotoAndStop("acerto");
54             timerFecharDesafio.start();
55             botaoResposta1.desativarBotao();
56             botaoResposta2.desativarBotao();
57             botaoResposta3.desativarBotao();
58             e.target.alpha=1;
59         }
60         else
61         {
62             contErros++;
63             if(contErros==1)
64             {
65                 removeChild(Botao(e.target));
66             }
67             else
68             {
69                 //mostrarRespostaCorreta;
70                 acertou=false;
71                 _iconeResposta.gotoAndStop("erro");
72                 Botao(e.target).desativarBotao();
73                 timerFecharDesafio.start();
74             }
75         }
76     }

```

Listagem 11 - Método executado após o click dom usuário em tela do aplicativo Desafio das Medidas

Na Listagem 12 pode-se observar a implementação da funcionalidade que adiciona uma penalidade ao jogador quando uma pergunta sobre distância não é respondida corretamente. O sistema verifica qual é a posição em tela da carroça do jogador e retorna a mesma para o marcador mais próximo, dessa maneira a distância percorrida entre a última pergunta e a atual é perdida. Respondendo novamente a pergunta de forma incorreta nenhuma penalidade é aplicada.

```

398     if(acertou==false)
399     {
400         //pergunta sobre distancia
401         if(tipoPergunta==0)
402         {
403             var xOnErro:Number;
404             if(carrocaJogador.x<950 && carrocaJogador.x>920)
405             {
406                 //carrocaInimigo.x = 1130;
407                 xOnErro = 1130;
408             }
409
410             if(carrocaJogador.x<750 && carrocaJogador.x>720)
411             {
412                 //carrocaInimigo.x = 931;
413                 xOnErro = 931;
414             }
415
416             if(carrocaJogador.x<550 && carrocaJogador.x>520)
417             {
418                 //carrocaInimigo.x = 731;
419                 xOnErro = 731;
420             }
421             else if(carrocaJogador.x<350 && carrocaJogador.x>320)
422             {
423                 //carrocaInimigo.x = 330;
424                 xOnErro = 330;
425             }
426             var tweenErro:Tween = new Tween(carrocaJogador, "x", None.easeIn, carrocaJogador.x, xOnErro,
427             _personagemJogador.addEventListener(MouseEvent.CLICK, puxarCarroca);
428             carrocaJogador._roda.stop();
429             carrocaJogador._rodaTras.stop();
430             tweenErro.addEventListener(TweenEvent.MOTION_FINISH, fimTweenErroJogador);
431         }

```

Listagem 12 - Trecho do código do método que adiciona uma penalidade ao usuário após uma resposta errada do aplicativo Desafio das Medidas

4.2.5 Implementação

Os testes do aplicativo Desafio das Medidas foram realizados utilizando o modo *debug* da ferramenta Adobe Animate. Os testes foram focados em observar erros na movimentação das carroças e na simulação da jogabilidade do adversário. Observados os resultados, erros foram corrigidos e a simulação ajustada para atender aos objetivos propostos.

4.3 MEMÓRIA MATEMÁTICA

O aplicativo Memória Matemática visa desenvolver a habilidade do usuário em relacionar as frações com números decimais, ao mesmo tempo o jogo visa ajudar no desenvolvimento da memória e da atenção.

4.3.1 Escopo

O aplicativo se baseia em um jogo da memória no qual o par será formado por uma fração e seu resultado decimal. As cartas possuem imagens de animais e

por meio de uma tela de configuração o usuário terá a possibilidade de optar se o desenho presente nas cartas combinará entre os pares ou não. Essa funcionalidade visa aumentar a dificuldade de jogo, pois exigirá mais atenção do usuário para encontrar os pares.

Foram definidas as seguintes regras e funcionalidades para o jogo:

- O número de cartas presentes no tabuleiro será configurável na tela de configurações.
- Um cronômetro será apresentado na tela para que o usuário possa acompanhar o tempo que levou para completar o jogo.
- O usuário poderá configurar se o desenho presente nas cartas combinará entre os pares ou não.
- As frações serão randomizadas pelo aplicativo, dessa forma as cartas possivelmente não serão iguais entre um jogo e outro.

4.3.2 Modelagem

O Quadro 7 apresenta a listagem dos requisitos funcionais identificados para o aplicativo Memória Matemática.

Identificação	Descrição
[RF-01] Configurações de jogo	O aplicativo conterà uma tela para configurar o número de cartas presentes no tabuleiro e se o desenho das cartas combinará entre os pares.
[RF-02] Jogabilidade	O aplicativo garantirá que toda carta tenha o seu par de fração.
[RF-04] Conclusão do jogo	O aplicativo finalizará o jogo sempre que o usuário encontrar todos os pares do tabuleiro.
[RF-05] Tela de ajuda	O aplicativo conterà uma tela com instruções de como jogar. Essa tela deve estar disponível na tela inicial do aplicativo e ser acessível a qualquer momento do jogo.

Quadro 7 - Memória Matemática - Requisitos funcionais

O Quadro 8 apresenta a listagem dos requisitos não funcionais identificados para o aplicativo Memória Matemática.

Identificação	Descrição
[NF-01] Ambientação	O jogo será ambientado em uma floresta.
[NF-02] Interface	A interface será adequada para uso em dispositivos móveis, contando com botões grandes e pouco texto.

[NF-03] Orientação	A orientação do aplicativo será na horizontal.
-----------------------	--

Quadro 8 – Memória Matemática - Requisitos não funcionais

A Figura 17 representa o diagrama de casos de uso.

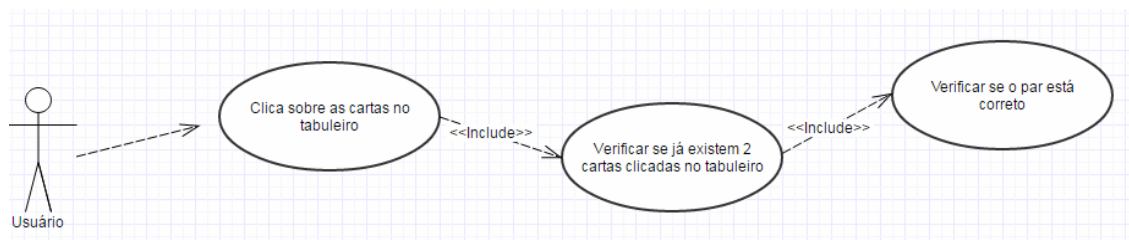


Figura 17 - Diagrama de casos de uso do aplicativo Memória Matemática

A Figura 18 representa o fluxograma do aplicativo.

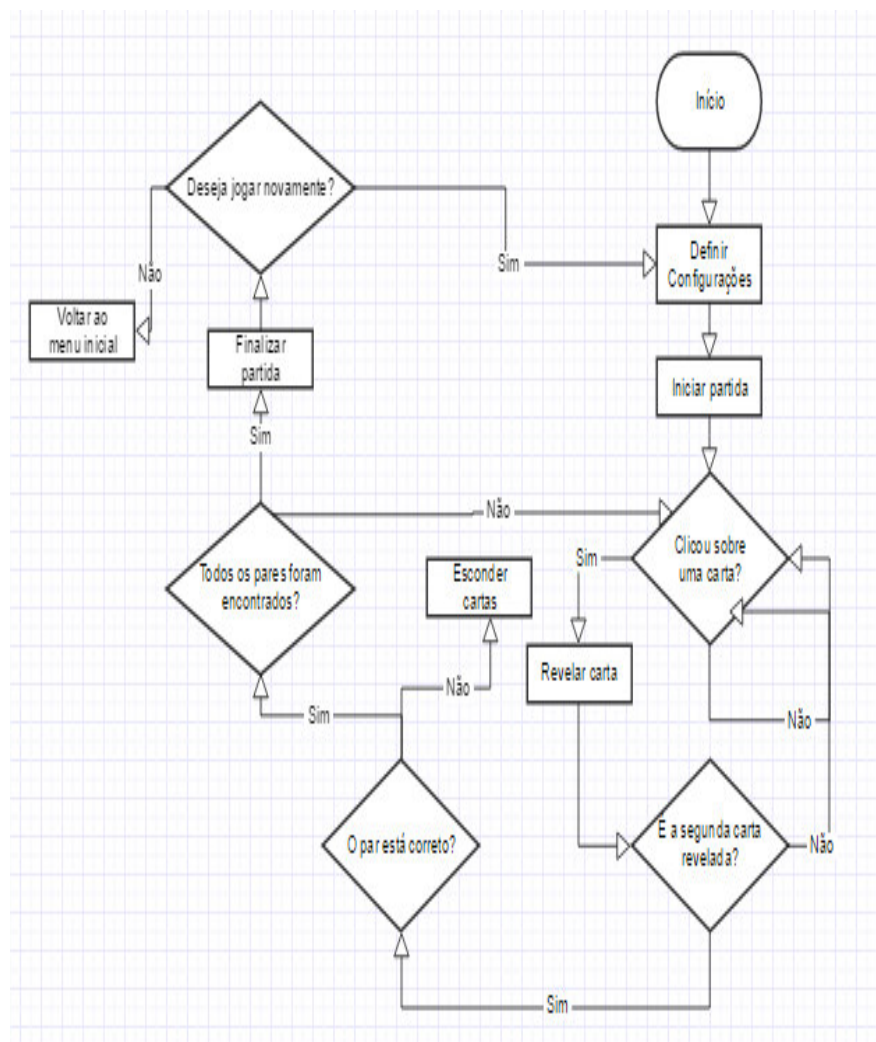


Figura 18 - Fluxograma do aplicativo Memória Matemática

4.3.3 Apresentação

Na Figura 19 está representada a tela principal do aplicativo Memória Matemática. No lado esquerdo da tela existe um cronômetro marcando o tempo de jogo do usuário. As cartas são dispostas em tela em linhas e colunas e o usuário deve clicar sobre elas para que sejam reveladas, formando o par corretamente ele fica exibido em tela até o tabuleiro ser completo.



Figura 19 - Tela principal do aplicativo Memória Matemática

A Figura 20 apresenta a tela de configurações do aplicativo Memória Matemática. Nessa tela o usuário poderá escolher a dificuldade do jogo, assim como a opção de “Combinar imagens” a qual influenciará na montagem dos pares.



Figura 20 - Tela de configurações do aplicativo Memória Matemática

Nas Figuras 21, 22 e 23, pode-se observar a tela principal do jogo Memória Matemática nas três configurações possíveis de dificuldade. A Figura 21 representa a dificuldade Fácil, a Figura 22 a dificuldade Médio e a Figura 23 a dificuldade Difícil.



Figura 21 - Tela principal do aplicativo Memória Matemática na configuração Fácil



Figura 22 - Tela principal do aplicativo Memória Matemática na configuração Médio



Figura 23 - Tela principal do aplicativo Memória Matemática na configuração Difícil

Vale ressaltar que além do número de cartas dispostas no tabuleiro e do número de animais disponíveis para montagem das cartas, não há nenhuma outra variação no jogo em virtude da dificuldade escolhida.

Nas Figuras 24 e 25 é apresentado a diferença da montagem dos pares em relação à configuração “Combinar imagens” ser selecionada ou não no menu de configurações do jogo. Na Figura 24, além do resultado da fração estar correto, as

imagens são as mesmas nas duas cartas. Já na Figura 25 por mais que o resultado seja correto, as imagens são diferentes. Essa opção foi adicionada para criar dificuldade e manter a atenção do jogador na resolução das frações e não somente nas cartas.



Figura 24 - Tela principal do aplicativo Memória Matemática com a opção de combinar imagens marcada



Figura 25 - Tela principal do aplicativo Memória Matemática com a opção de combinar imagens desmarcada

A Figura 26 apresenta a tela que é exibida para o usuário após todos os

pares serem encontrados no tabuleiro. Clicando na opção “Sim” o usuário é levado para a tela de configuração do jogo, clicando na opção “Não” ele retorna ao Menu principal do jogo.

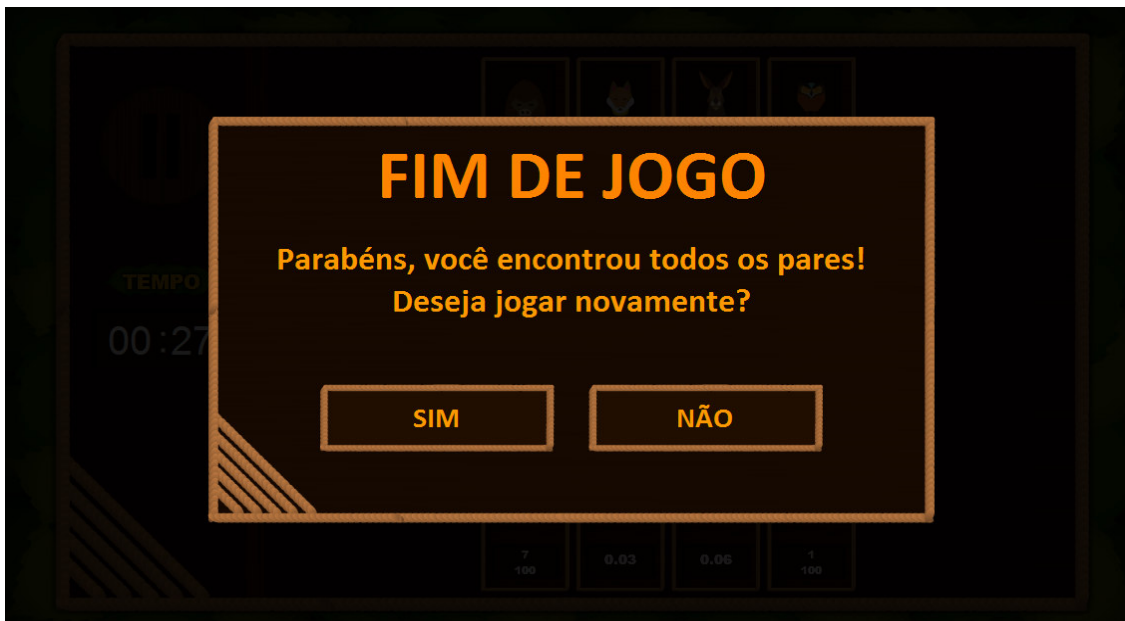


Figura 26 - Tela apresentada ao usuário após o fim de jogo do aplicativo Memória Matemática

A Figura 27 apresenta a tela de ajuda do jogo, que pode ser acessada por meio de um atalho do menu principal do jogo clicando no botão “Ajuda”. Nessa tela há um texto explicando as regras e o objetivo do jogo.

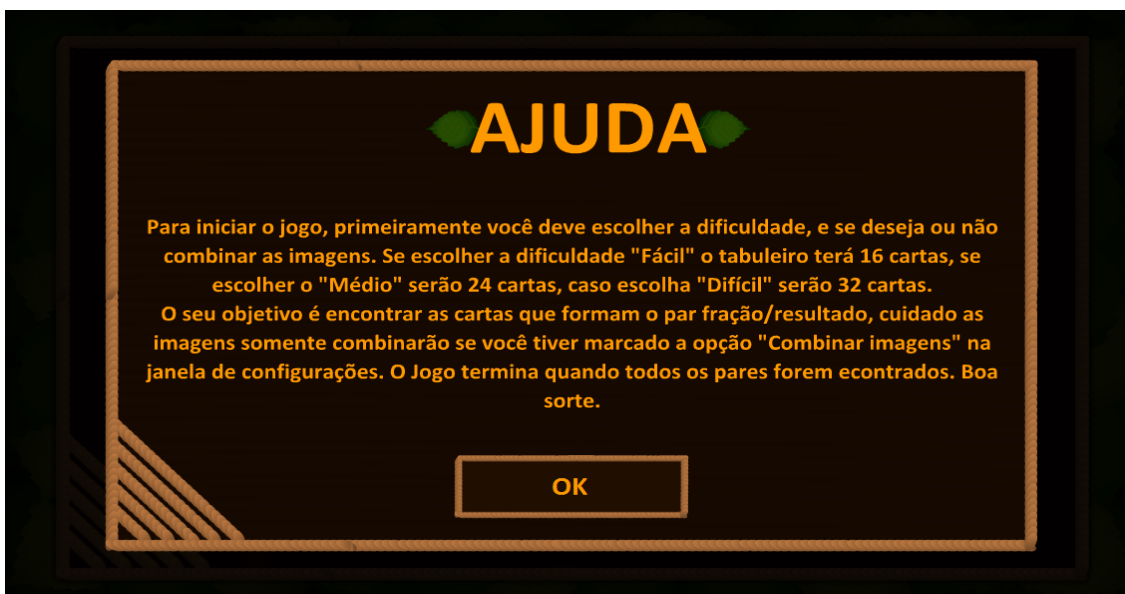


Figura 27 - Tela de ajuda do aplicativo Memória Matemática

4.3.4 Implementação

A seguir estão trechos da codificação do aplicativo, assim como sua estrutura de pastas e arquivos. A Figura 28 apresenta a pasta principal do aplicativo Memória Matemática. Na pasta “code” estão os arquivos do tipo “ActionScript” que contém a codificação do jogo, já o arquivo MemoriaMatematica.apk é gerado pela ferramenta Adobe Animate e é utilizado para instalar a aplicação em sistemas operacionais Android.

Nome	Data de modificação...	Tipo	Tamanho
code	02/11/2015 15:03	Pasta de arquivos	
release	02/11/2015 16:45	Pasta de arquivos	
Certificado de publicação Android	16/09/2015 08:29	Troca de Informaç...	2 KB
MemoriaMatematica.apk	02/11/2015 17:27	Arquivo APK	9.478 KB
MemoriaMatematica	02/11/2015 21:24	Documento do A...	2.267 KB
MemoriaMatematica	02/11/2015 17:27	Chrome HTML Do...	3 KB
MemoriaMatematica	05/06/2017 18:51	Flash Movie	270 KB
MemoriaMatematica-app	05/06/2017 18:51	Documento XML	2 KB

Figura 28 - Estrutura de pastas e arquivos do aplicativo Memória Matemática

Na Listagem 13 pode-se observar o método iniciarApp() que é chamado por meio do click no botão “Iniciar” da tela de configurações do jogo. Esse método recebe dois parâmetros que indicam as configurações selecionadas pelo usuário.

```

57     public function iniciarApp(dif:String, combImg:Boolean):void
58     {
59         dificuldade = dif;
60         combinarImagens = combImg;
61
62         configurarJogo();
63         relógio.iniciarRelógio();
64         jogoIniciado=true;
65     }
66

```

Listagem 13 - Método de inicialização de jogo do aplicativo Memória Matemática

A Listagem 14 representa um trecho do método configurarJogo(). Nesse método é definida a quantidade de cartas do tabuleiro conforme a configuração selecionada pelo usuário: Fácil são 16 cartas, Médio são 24 cartas e Difícil são 32 cartas.

```

67     private function configurarJogo():void
68     {
69         if(dificuldade=="facil")
70         {
71             numCartas = 16;
72         }
73         else if(dificuldade=="medio")
74         {
75             numCartas = 24;
76         }
77         else
78         {
79             numCartas = 32;
80         }

```

Listagem 14 - Método de configuração de jogo do aplicativo Memória Matemática

Na Listagem 15 pode-se observar a montagem das cartas conforme a configuração de combinar imagens. Caso seja verdadeira, somente uma imagem é sorteada da lista previamente definida e ela é adicionada nas duas cartas que formam o par. Caso seja falsa, para cada carta um animal diferente é sorteado. As imagens também foram divididas em dificuldade, para cada nível o número de imagens disponíveis aumenta.

```

159         if(combinarImagens==true)
160         {
161             if(j==0)
162             {
163                 var randomAnimal:Number = Math.floor(Math.random()*arrImgs.length);
164                 var strAnimal = arrImgs[randomAnimal]
165                 arrImgs.splice(randomAnimal,1);
166             }
167             carta.setAnimal(strAnimal);
168         }
169         else
170         {
171             var randomAnimal2;
172             var strAnimal2;
173
174             if(dificuldade=="facil")
175             {
176                 randomAnimal2 = Math.floor(Math.random()*arrImgs.length);
177                 strAnimal2 = arrImgs[randomAnimal2];
178                 arrImgs.splice(randomAnimal2,1);
179             }
180             else if(dificuldade=="medio")
181             {
182                 randomAnimal2 = Math.floor(Math.random()*arrImgs2.length);
183                 strAnimal2 = arrImgs2[randomAnimal2];
184                 arrImgs2.splice(randomAnimal2,1);
185             }
186             else
187             {
188                 randomAnimal2 = Math.floor(Math.random()*arrImgs3.length);
189                 strAnimal2 = arrImgs3[randomAnimal2];
190                 arrImgs3.splice(randomAnimal2,1);
191             }
192
193             carta.setAnimal(strAnimal2);
194         }

```

Listagem 15 - Trecho do código de criação das cartas do aplicativo Memória Matemática

A Listagem 16 representa um trecho do método `posicionarCartas()` que tem por objetivo posicionar corretamente as cartas em tela conforme o número estabelecido pela configuração de dificuldade. Inicialmente é definida uma posição fixa inicial chamada de "xBase". Após a primeira carta ser adicionada, as subsequentes são adicionadas conforme a posição da anterior acrescida pela largura do objeto, esse cálculo assegura que nenhuma carta é criada sobre outra.

```

252         for (var j: Number=0; j<arrAux.length; j++)
253         {
254             if (j==0)
255             {
256                 //trace("J0primeiradalinha:"+j);
257                 arrAux[j].x=xBase;
258                 arrAux[j].y=yBase;
259                 _tabuleiro.addChild(arrAux[j]);
260             }
261             if ((j<numCartasLinha) && (j!=0))
262             {
263                 //trace("J0:"+j);
264                 arrAux[j].x=(xBase+arrAux[j-1].width*j);
265                 arrAux[j].x = arrAux[j].x+(7*j);
266                 arrAux[j].y=yBase;
267                 _tabuleiro.addChild(arrAux[j]);
268             }

```

Listagem 16 - Trecho do código de posicionamento das cartas do aplicativo Memória Matemática

Na Listagem 17 está representado o método `onClickCarta()` que é chamado pelo *click* do usuário em uma carta no tabuleiro. Esse método verifica se o *click* foi realizado sobre uma carta ainda não revelada, após isso a carta é guardada em uma lista para que previamente seja realizada a comparação entre as duas clicadas. Um contador é iniciado em zero e incrementado a cada *click* para identificar se existem duas cartas exibidas em tela para que o aplicativo faça a comparação..

```

205     private function onClickCarta(e:Event)
206     {
207         if ((Carta(e.target).currentFrameLabel=="esconder") && (clickHabilitado==true))
208         {
209             contCliques++;
210             arrCartasClicadas.push(Carta(e.target));
211             if (contCliques==1)
212             {
213                 //idCartaPrimeiroClick = e.target(Carta).getIdCarta();
214                 idCartaPrimeiroClick = Carta(e.target).getIdCarta();
215                 Carta(e.target).gotoAndStop("mostrar");
216                 //trace("Id carta 1:" + idCartaPrimeiroClick);
217             }
218             else
219             {
220                 idCartaSegundoClick = Carta(e.target).getIdCarta();
221                 Carta(e.target).gotoAndStop("mostrar");
222                 //trace("Id carta 1:" + idCartaSegundoClick);
223                 verificarPar(idCartaPrimeiroClick, idCartaSegundoClick);
224             }
225         }
226     }
227
228

```

Listagem 17- Método chamado após click do usuário sob uma carta do aplicativo Memória Matemática

Na Listagem 18 é possível observar o método `verificaPar()` que recebe por parâmetro o "id" das duas cartas reveladas pelo usuário. Esse método verifica se o par está correto, estando correto é verificado se todos os pares já foram encontrados para que um novo jogo seja iniciado. Caso o par esteja incorreto, as duas cartas voltam a serem escondidas e o jogo reinicia.

```

339     private function verificarPar(idCarta1:Number, idCarta2:Number):void
340     {
341         contCliques=0;
342         if(idCarta1==idCarta2)
343         {
344             //trace("Acertou o par!");
345             arrCartasClicadas[0].mouseEnabled=false;
346             arrCartasClicadas[1].mouseEnabled=false;
347             arrCartasClicadas = new Array();
348             paresEcontrados++;
349             //trace("Pares encontrados:" + paresEcontrados);
350             //trace("Total pares: " + totalPares);
351             if(paresEcontrados==totalPares)
352             {
353                 //trace("TOTAL PARES");
354                 var timerNovoJogo = new Timer(1000,1);
355                 relógio.pausarRelógio();
356                 timerNovoJogo.addEventListener(TimerEvent.TIMER_COMPLETE, function(){
357                     owner.menus.abrirJanelaJogarNovamente();
358                 });
359                 timerNovoJogo.start();
360             }
361         }
362         else
363         {
364             trace("Errou o par");
365             timerEscondeCarta = new Timer(1000,1);
366             removerEventoClickCartas();
367             timerEscondeCarta.addEventListener(TimerEvent.TIMER_COMPLETE, function(){
368                 arrCartasClicadas[0].gotoAndStop("esconder");
369                 arrCartasClicadas[1].gotoAndStop("esconder");
370                 arrCartasClicadas = new Array();
371                 adicionarEventoClickCartas();
372             });
373             timerEscondeCarta.start();
374         }
375     }

```

Listagem 18 - Método de verificação de par do aplicativo Memória Matemática

4.3.5 Testes

Os testes do aplicativo Memória Matemática foram realizados paralelamente ao desenvolvimento e utilizando o modo *debug* da ferramenta Adobe Animate. Os testes estiveram focados em encontrar erros na montagem dos pares e no posicionamento das cartas em tela.

5 CONCLUSÃO

O objetivo principal deste trabalho foi desenvolver um conjunto de jogos para auxílio no ensino da matemática. Foram desenvolvidos jogos que podem ser utilizados com ou sem a supervisão de um professor, pois possuem uma forma bastante intuitiva de jogar e contam com uma janela de ajuda que apresenta as regras do jogo.

Após concluído o levantamento dos requisitos foram definidas as regras e a jogabilidade de cada aplicativo, também foram desenvolvidas as telas principais. A plataforma de desenvolvimento escolhida foi a Adobe Animate CC, pois ela oferece diversas ferramentas necessárias para o desenvolvimento jogos.

Durante o processo de desenvolvimento, primeiramente foi realizado o desenho de todos os elementos de compõe os jogos. Terminada a fase de desenho e ambientação foi realizada a codificação da jogabilidade e das regras de interação. Os testes foram conduzidos parte em paralelo ao desenvolvimento e após o término. Durante os testes buscou-se identificar erros de lógica e interação.

O maior obstáculo encontrado durante o desenvolvimento foi o de definir o modelo, a ambientação dos jogos e as regras de interação dos adversários.

Após a conclusão desse trabalho os aplicativos serão aprimorados, também existe a intenção da inclusão dos jogos em um ambiente escolar a fim de avaliar a sua usabilidade e o seu comportamento.

O desenvolvimento desses jogos foi uma oportunidade de aprendizado. Pretende-se, futuramente, desenvolver outros jogos para a área educacional, que podem ser de outras disciplinas.

REFERÊNCIAS

APPLE. **O que é o iOS?**. Disponível em: <<http://www.apple.com/br/ios/>>. Acesso em: jun. 2016.

BRASIL, Ministério da educação - Secretaria de educação fundamental - **PCN'S Parâmetros curriculares nacionais**. Brasília: MEC/SEF, 1998.

KISHIMOTO, Tizuco Mochida. **Jogo, brinquedo, brincadeiras e a educação**. 4ª Ed. São Paulo, Editora Cortez: 2000.

LECHETA, Ricardo. **Google Android. Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3ª Edição. Editora Novatec: 2013.

LIMA JUNIOR, A. S.. **A escola no contexto das tecnologias de comunicação e informação: do dialético ao virtual**. Salvador: EDUNEB, 2007.

PIAGET, Jean. **A psicologia da inteligência**. Editora Fundo de Cultura S/A. Lisboa, 1967.

PRENKSY, Marc. **Digital Natives, digital immigrants. On the Horizon**, v. 9, n. 5, out. 2001b.

ROJO, Roxane. **Pedagogia dos multiletramentos: diversidade cultural e de linguagens na escola**. In: ROJO, Roxane. Moura, Eduardo (Orgs.). **Multiletramentos na escola**. São Paulo: Parábola, 2012. p. 11-31.

SMOLE, Kátia Cristina Stocco. **A Matemática na Educação Infantil. A teoria das inteligências múltiplas na prática escolar**. Porto Alegre, Editora Artes Médicas: 1996.

TAPSCOTT, Dan. **Geração Digital: a crescente e irreversível ascensão da geração net**. Trad. Ruth Gabriela Bahr. São Paulo: Makron Books, 1999.