



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

GEOFRANGITE CÂMARA DA SILVA

**ARTEMIS: REALIDADE AUMENTADA E GAMIFICAÇÃO PARA ENSINO DE
ARQUITETURA DE COMPUTADORES COM MIPS**

MOSSORÓ

2019

GEOFRANGITE CÂMARA DA SILVA

**ARTEMIS: REALIDADE AUMENTADA E GAMIFICAÇÃO PARA ENSINO DE
ARQUITETURA DE COMPUTADORES COM MIPS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Linha de Pesquisa: Projeto de Sistemas e Circuitos

Orientador: Sílvia Roberto Fernandes de Araújo, Prof. Dr.

Co-orientador: Leiva Casemiro Oliveira, Prof. Dr.

MOSSORÓ

2019

GEOFRANGITE CÂMARA DA SILVA

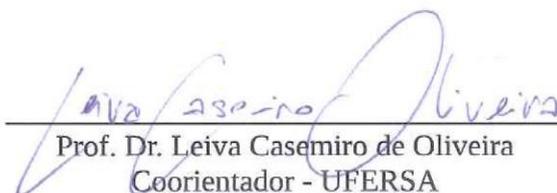
**ARtEMIS: Realidade Aumentada e Gamificação para Ensino de Arquitetura de
Computadores com MIPS**

Dissertação apresentada ao Programa de Pós-Graduação
em Ciência da Computação para a obtenção do título de
Mestre em Ciência da Computação.

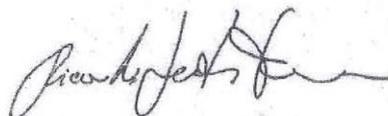
APROVADA EM: 29 / 05 / 2019



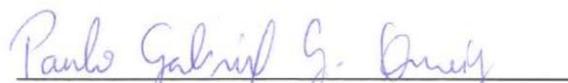
Prof. Dr. Sílvio Roberto Fernandes de Araújo
Orientador e Presidente da Banca



Prof. Dr. Leiva Casemiro de Oliveira
Coorientador - UFERSA



Prof. Dr. Ricardo dos Santos Ferreira
Examinador Externo - UFV



Prof. Dr. Paulo Gabriel Gadelha Queiroz
Examinador Interno - UFERSA

Aos meus pais, Dona Graça e Seu Netinho, e a todos os outros membros da minha família por tudo que fizeram e fazem por mim.

AGRADECIMENTOS

Agradeço a Deus primeiramente, por ter me dado saúde e meios para dar mais esse importante passo na caminhada da vida acadêmica.

Meu muito obrigado à minha família por acreditar em mim e por todo o apoio e incentivo dado durante esses dois anos. Essa conquista é nossa.

Agradeço aos meus orientadores, professor Sílvio Fernandes e professor Leiva Casemiro, pelos conhecimentos transmitidos, confiança, motivação e paciência. Muito obrigada pela oportunidade de trabalhar com vocês.

Sou grata a todos os amigos que convivi durante o mestrado. Amigos que estavam ali para dizer que “Vai dar certo” e com quem pude contar com a ajuda para fazer dar certo; Amigos que viveram comigo o sufoco de PAA (Projeto e Análise de Algoritmos) e me ajudaram a sobreviver a ele; Amigos que sempre me receberam de portas abertas em suas casas; Amigos que me ajudaram quando precisei e com os quais vivi bons momentos e que nas horas vagas compartilharam histórias para descontrair da tensão do mestrado. Agradeço em especial a Ademar Neto, Alexandre Adler, Antonio Queiróz, Arthur Domingues, Cynthia Maia, Daniel Vieira, Elizeu Sandro, Érico Gomes, Etiene Júnior, Igor Fernandes, Igor Melo, Jesaías Carvalho, Johnattan Douglas, Julio Cartier, Otilia Santos, Rayana Rocha, Thyago Sobreira e Wedson Carlos.

Agradeço aos colegas do laboratório GESyCA (*Group of Embedded Systems and Computer Architecture*) que participam dos projetos de ensino de robótica, em especial a Alfredo Lopes, Igo Joctan, Isabelle Oliveira, Laura Rodrigues e Vinícius Sales, pela agradável convivência e pelas histórias compartilhadas que alegraram muitos de meus dias no laboratório.

Agradeço a todo corpo docente do Programa de Pós-Graduação em Ciência da Computação (PPgCC), pela contribuição para minha formação. Esse agradecimento se estende também aos demais funcionários que contribuem para o funcionamento do programa.

À CAPES, pelo apoio financeiro e a Universidade do Estado do Rio Grande do Norte (UERN) e Universidade Federal Rural do Semi-Árido (UFERSA), por todo suporte dado para a realização deste trabalho.

Sou grata a todos que participaram da validação do ARtEMIS.

Por fim, agradeço a todos que direta ou indiretamente contribuíram para a realização deste trabalho. Sintam-se citados.

Só existem dois dias no ano que nada pode ser feito. Um se chama ontem e o outro se chama amanhã, portanto hoje é o dia certo para amar, acreditar, fazer e principalmente viver.

Dalai Lama

RESUMO

Os simuladores são ferramentas desenvolvidas com o intuito de aumentar o nível de abstração de conteúdos. Eles têm sido de grande importância nos cursos de ciência da computação para o ensino de conteúdos de disciplinas que tratam do *hardware*, tidas como complexas em termos de aprendizado. No contexto de Organização e Arquitetura de Computadores vários simuladores foram e continuam sendo desenvolvidos, no entanto, eles não apresentam integração com o material de estudo (livro). Neste trabalho é apresentado um simulador da arquitetura do processador MIPS que tem como objetivo permitir maior integração entre o material de estudo e as ferramentas de apoio ao aprendizado. Para isso, o simulador desenvolvido reconhece, por meio da realidade aumentada, as figuras do livro “Organização e Projeto de Computadores” de Patterson e Hennessy, 3ed e as recria como modelos interativos com os quais é possível interagir para facilitar o aprendizado por meio da obtenção de informações e visualização do funcionamento interno do MIPS. Também é proposta uma metodologia de uso com base em gamificação. Foi realizada uma validação, a partir da qual foi verificado que o uso do simulador é capaz de promover uma melhora no desempenho dos alunos. Além disso, os participantes da validação o consideraram muito útil para auxiliar no aprendizado de Organização e Arquitetura de Computadores, acharam que aprenderam muito usando-o e consideraram importante a inclusão da RA e da gamificação para estimular o aprendizado.

Palavras-chave: Realidade Aumentada, Gamificação, Arquitetura de Computadores, MIPS, Computação e Educação, Ferramenta Educacional.

ABSTRACT

Simulators are tools developed to increase the level of content abstraction. They have been of great importance in the courses of computer science for the teaching of contents of disciplines that deal with the hardware, considered as complex in terms of learning. In the context of Organization and Computer Architecture several simulators have been and continue to be developed, however, do not present integration with the study material (book). This work presents a simulator of the architecture of the MIPS processor that aims to allow greater integration between the study material and the tools to support learning. For this, the developed simulator recognizes, through augmented reality, the figures in Patterson and Hennessy's book "Computer Organization and Design", 3ed and recreates them as interactive models with which it is possible to interact to facilitate learning by obtaining information and visualization of the internal workings of MIPS. A methodology of use based on gamification is also proposed. A validation was performed in which it was verified that the simulator is able to promote improvement in students' performance. In addition, the validation participants considered it very useful to assist in the learning of OCA, felt that they had learned a lot using it and considered it important to include RA and gamification to stimulate learning.

Keywords: Augmented Reality, Gamification Computer Architecture, MIPS, Computing and Education, Educational Tool.

LISTA DE FIGURAS

Figura 1 – Taxa de aprovação e reprovação em OAC entre os nos semestres de 2010.2 a 2014.1.....	21
Figura 2 – Motivos das reprovações em OAC entre os nos semestres de 2010.2 a 2014.1	22
Figura 3 – Caminho de dados do processador MIPS. Em azul estão destacadas as partes do caminho por onde transitam os sinais de controle com destaque para os barramentos ligados ao controle da ULA.....	33
Figura 4 – Contínuo de virtualidade.....	36
Figura 5 – Esquema de funcionamento da RA.....	37
Figura 6 – Etapas seguidas na condução da revisão.....	45
Figura 7 – Etapas do processo de seleção	48
Figura 8 – Distribuição dos artigos por nível de escolaridade	51
Figura 9 – Distribuição dos artigos por área e grande área de conhecimento	52
Figura 10 – Formas de agrupamento identificadas. Ag1 – grupo de teste e grupo de controle; Ag2 – único grupo realiza tarefas com apenas um tipo de RA; Ag3 – único grupo realiza tarefas com e sem RA; Ag4 – 2 ou mais grupos, cada um realiza tarefas um tipo de RA; e Ag5 – 2 grupos que realizam tarefas tanto com RA quanto sem RA, mas em ordens diferentes	53
Figura 11 – Formas de aplicação identificadas	54
Figura 12 – Tipos dos testes aplicados.....	55
Figura 13 – Estratégias de Avaliação da RA identificadas	56
Figura 14 – Fluxo de desenvolvimento do ARtEMIS	58
Figura 15 – Ilustrações 5.1, 5.2, 5.17 e 5.24 do livro texto (PATTERSON; HENNESSY, 2005) utilizadas como marcadores pela RA do ARtEMIS. Em cada ilustração, os componentes acrescentados em relação a ilustração anterior estão destacados em vermelho.	63
Figura 16 – Subfases que constituem as fases com as instruções que abordam.....	64
Figura 17 – Missão da subfase 1-1	64
Figura 18 – Telas com informações sobre o progresso do usuário no jogo. a) Informações sobre a pontuação adquirida e fase alcançada pelo usuário. b) Conquistas disponíveis para o usuário. As destacadas em azul estão liberadas e as em cinza estão pendentes	66
Figura 19 – Tela de cadastro do usuário.....	67
Figura 20 – Tela inicial do ARtEMIS. a) Apenas a subfase 1-1 está disponível. b) Todas as subfases estão liberadas. c) Opções de ir para o desafio da subfase ou reconhecer a ilustração do livro.....	67

Figura 21 – Tela de reconhecimento antes da ilustração ser reconhecida.....	68
Figura 22 – Descrição da subfase 1-1	69
Figura 23 – Tela de reconhecimento após a ilustração ser reconhecida. Destaque para os botões de funcionalidades	69
Figura 24 – Tipos de desafios disponíveis. a) Questões de verdadeiro ou falso. b) Questões de múltipla escolha com textos	71
Figura 25 – Tipos de desafios disponíveis. a) Questões de múltipla escolha com caminhos de dados. b) Questões para configurar os sinais do controle	72
Figura 26 – Tela de resultado. Exemplo de mensagem para quando o usuário acerta todas as questões do desafio	73
Figura 27 – Esquema de cores e valores dos barramentos	73
Figura 28 – Reconhecimento da ilustração 5.1 pelo ARtEMIS. Detalhe para o balão informativo com informações sobre a unidade banco de registradores	74
Figura 29 – Funcionamento do reconhecimento da ilustração 5.2 mostrando a simulação da execução de uma instrução do tipo <i>add</i>	76
Figura 30 – Caminho de dados da ilustração 5.17 modificado. Os componentes adicionados estão destacados em vermelho. a) Extensão do caminho de dados para dá suporte as instruções <i>sll</i> e <i>srl</i> . b) Extensão do caminho de dados para dá suporte a instrução <i>bne</i>	77
Figura 31 – Caminho de dados da ilustração 5.24 modificado. Os componentes adicionados estão destacados em vermelho. a) Extensão do caminho de dados para dá suporte as instruções <i>sll</i> e <i>srl</i> . b) Extensão do caminho de dados para dá suporte a instrução <i>bne</i> . c) Extensão do caminho de dados para dá suporte a instrução <i>jr</i> . d) Extensão do caminho de dados para dá suporte a instrução <i>jal</i>	78
Figura 32 – Abas Código e Código de máquina. a) Aba código inicialmente exibida ao usuário. b) Aba código após o usuário selecionar um tipo de instrução para inserir. c) Aba código de máquina exibindo o binário das instruções inseridas	79
Figura 33 – Tela para visualizar/alterar os valores da memória de dados e dos registradores.....	79
Figura 34 – Resultado obtidos para as questões relacionadas ao perfil dos participantes. a) Resultado do GC. b) Resultado do GP	82
Figura 35 – Percentuais de participantes que acertaram cada questão do questionário pré-teste. a) Percentuais do GC. b) Percentuais do GP	83
Figura 36 – Distribuição dos participantes em relação a quantidade de questões que acertaram. a) Distribuição do GC. b) Distribuição do GP	84
Figura 37– Percentuais de participantes que acertaram cada questão do ARtEMIS.....	85
Figura 38 – Distribuição dos participantes em relação a quantidade de questões que acertaram	86

Figura 39 – Comparação dos percentuais de acertos dos participantes obtidos no pré-teste e usando o ARtEMIS.....	87
Figura 40 – Comparação do desempenho médio dos participantes que cursaram OAC com o dos participantes que não cursaram	88
Figura 41 – Comparação do desempenho médio dos participantes que não estudaram o processador MIPS com o dos participantes que estudaram.....	89
Figura 42 – Comparação do desempenho médio dos participantes que cursaram OAC mas não estudaram o MIPS com o dos participantes que cursaram OAC e estudaram o MIPS	90
Figura 43 – Questão 01: Quão fácil é aprender a usar o aplicativo; quão clara são as etiquetas/ícones do menu e instruções de uso?.....	91
Figura 44 – Questão 02: A movimentação entre as telas é lógica/precisa/apropriada/ininterrupta; todos os links de tela necessários estão presentes?.....	92
Figura 45 – Questão 03: As interações (toques/swipes/pinches/scrolls) são consistentes e intuitivas em todos os componentes/telas?	92
Figura 46 – Questão 04: A organização e o tamanho dos botões/ícones/menus/conteúdo na tela são apropriados ou ampliáveis (zoom) se necessário?	93
Figura 47 – Questão 05: Qual é a qualidade/resolução dos gráficos usados para os botões/ícones/menus/conteúdo?.....	93
Figura 48 – Questão 06: Quão boa é a aparência do aplicativo?.....	94
Figura 49 – Questão 01: Quão importante você achou a inclusão da gamificação (inserção dos elementos de jogo fases, pontos e conquistas) no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores?.....	95
Figura 50 – Questão 02: Quão importante você achou a inclusão da Realidade Aumentada no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores? .	95
Figura 51 – Questão 03: Quão útil você achou o ARtEMIS para aprender sobre Organização e Arquitetura de Computadores?.....	96
Figura 52 – Questão 04: O reconhecimento das figuras foi preciso?.....	96
Figura 53 – Questão 05: O reconhecimento das figuras foi rápido?	97
Figura 54 – Questão 01: Quanto você acha que aprendeu respondendo o questionário pré-teste?.....	97
Figura 55 – Questão 02: Quanto você acha que aprendeu usando o ARtEMIS?	98
Figura 56 – Questão 03: Quão difíceis foram as questões apresentadas no questionário pré-teste?.....	98
Figura 57 – Questão 04: Quão difíceis foram as questões apresentadas no ARtEMIS?	99
Figura 58 – Diagrama de casos de uso.....	108

Figura 59 – Diagrama de Atividades para o caso de uso “Cadastrar-se”	109
Figura 60 – Diagrama de Atividades para o caso de uso “Alterar Configurações”	110
Figura 61 – Diagrama de Atividades para o caso de uso “Ver Informações”	111
Figura 62 – Diagrama de Atividades para o caso de uso “Jogar”	112

LISTA DE TABELAS

Tabela 1 – Conjunto de registradores do MIPS	28
Tabela 2 – Conjunto de instruções básicas da arquitetura MIPS.....	30
Tabela 3 – Formatos de instruções da arquitetura MIPS.....	31
Tabela 4 – Elementos de jogos ligados à gamificação.....	38
Tabela 5 – Comparação dos trabalhos relacionados com o simulador ARtEMIS.....	44
Tabela 6 – Palavras-chave e seus sinônimos.....	46
Tabela 7 – <i>Strings</i> aplicadas na busca realizada na IEEE Xplore	49
Tabela 8 – <i>Strings</i> aplicadas na busca realizada na ACM Digital Library.....	50

LISTA DE ABREVIATURAS E SIGLAS

Ag1	Grupo de teste e grupo de controle
Ag2	Único grupo realiza tarefas com apenas um tipo de RA
Ag3	Único grupo realiza tarefas com e sem RA
Ag4	2 ou mais grupos, cada um realiza tarefas um tipo de RA
Ag5	2 grupos que realizam tarefas tanto com RA quanto sem RA, mas em ordens diferentes
APK	<i>Android Application Pack</i>
ARMS	<i>Augmented Reality MIPS Simulator</i> (nome inicial dado ao projeto do ARtEMIS)
ARtEMIS	<i>Augmented Reality MIPS Simulator</i>
DIMIPSS	<i>Didact Interactive MIPS Simulator</i>
FA1	Apenas Pós-teste
FA2	Pré-teste e Pós-teste
FA3	Pré-teste, Pós-teste e teste de retenção
IDE	<i>Integrated Development Environment</i>
ISA	<i>Instruction Set Architecture</i>
GC	Grupo Completo
GESyCA	<i>Group of Embedded Systems and Computer Architecture</i>
GP	Grupo Parcial
GUI	<i>Graphical User Interface</i>
MARS	<i>Mips Assembly and Runtime Simulator</i>
MIPS	<i>Microprocessor without interlocked pipeline stages</i>
OA	Objeto de Aprendizagem
OAC	Organização e Arquitetura de Computadores
PAA	Projeto e Análise de Algoritmos
PC	Contador de Programa
PPgCC	Programa de Pós-Graduação em Ciência da Computação
RA	Realidade Aumentada
RISC	<i>Reduced Instruction Set Computing</i>
RM	Realidade Misturada
RSL	Revisão Sistemática da Literatura
SO	Sistema Operacional

SDK	<i>Software Development Kit</i>
UERN	Universidade do Estado do Rio Grande do Norte
UFERSA	Universidade Federal Rural do Semi-Árido
VA	Virtualidade Aumentada
WTCC	Workshop Técnico-Científico de Computação
XP	Pontos de Experiência

SUMÁRIO

1 INTRODUÇÃO	21
1.1 Objetivos	24
1.2 Metodologia	25
1.3 Organização do Trabalho.....	25
2 REFERENCIAL TEÓRICO.....	26
2.1 Organização e Arquitetura de Computadores.....	26
2.2 O Processador MIPS	26
2.2.1 Arquitetura do Conjunto de Instruções	28
2.2.2 Formato de Instruções	30
2.2.3 Caminho de dados MIPS	32
2.3 Realidade Aumentada	36
2.4 Gamificação	37
2.5 Trabalhos relacionados	38
2.5.1 MARS	38
2.5.2 WEBMIPS.....	39
2.5.3 VISIMIPS	40
2.5.4 DIMIPSS	40
2.5.5 DRMIPS.....	41
2.5.6 MIPSFPGA.....	41
2.5.7 RaVI	42
2.5.8 UCOMIPSIM 2.0.....	42
2.5.9 RA para ensinar sobre a arquitetura do processador Neander	43
2.5.10 VirtualComponent.....	43
2.6 Considerações Finais	43
3 uso de realidade aumentada na educação	45
3.1 Metodologia	45

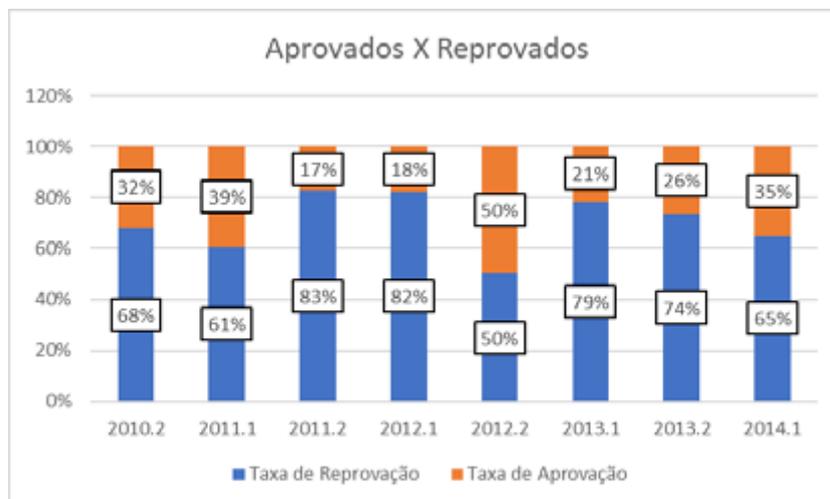
3.1.1 Planejamento da revisão.....	45
3.1.2 Condução da Revisão	49
3.2 Resultados da RSL	51
3.2.1 Questão de Pesquisa 1: Em quais níveis de escolaridade e áreas de conhecimento a RA tem sido mais aplicada?	51
3.2.2 Questão de Pesquisa 2: Quais estratégias de avaliação têm sido utilizadas para verificar o impacto da aplicação da RA no desempenho dos alunos?	52
3.2.3 Questão de Pesquisa 3: Quais estratégias têm sido utilizadas para permitir aos alunos avaliar a efetividade da RA como ferramenta facilitadora do aprendizado?.....	55
3.3 Considerações finais.....	56
4 O SIMULADOR ARTEMIS.....	58
4.1 Visão Geral	59
4.2 Implementação.....	60
4.3 Proposta de gamificação com ARTEMIS	61
4.4 Interface do ARtEMIS	66
4.5 Considerações finais.....	80
5 VALIDAÇÃO DO ARTEMIS	81
5.1 Metodologia	81
5.1.1 Etapa 01: Pré-teste.....	81
5.1.1.1 Perfil do participante.....	82
5.1.1.2 Conhecimentos sobre o MIPS	83
5.1.2 Etapa 02: uso do ARtEMIS e Pós-teste	84
5.1.3 Etapa 03: avaliação do processo	90
5.1.3.1 Avaliação da usabilidade do ARtEMIS.....	90
5.1.3.2 Avaliação das estratégias empregadas no desenvolvimento do ARtEMIS	94
5.1.3.3 Avaliação das atividades propostas	97
5.2 Considerações finais.....	99
6 Conclusão	100
6.1 Trabalhos Futuros	101

6.2 Publicações.....	102
REFERÊNCIAS.....	103
APÊNDICES.....	107
Apêndice A – Diagramas UML	108
Apêndice B – Questionário Pré-teste	113
Apêndice C – Questionário Pós-teste.....	118
Apêndice D – Questionário de Avaliação do ARtEMIS e das Atividades Propostas.....	124

1 INTRODUÇÃO

A grade curricular de qualquer curso de graduação deve abranger o conteúdo necessário para fornecer uma formação profissional que capacite o indivíduo para exercer a profissão escolhida. No tocante aos cursos de Ciência da Computação, embora tenham um enfoque maior nas áreas de conhecimento voltadas para *software*, esses cursos também apresentam, em suas grades, disciplinas que abordam conteúdo relacionado ao *hardware* pois, ao entender os aspectos físicos e funcionais das máquinas, os profissionais tornam-se melhores programadores e são capazes de tirar o melhor proveito dessas. No entanto, apesar da importância, Organização e Arquitetura de Computadores (OAC) é tida como complexa em termos de aprendizado o que a torna uma das disciplinas responsáveis por parte da retenção e da evasão dos alunos. Fernandes e Silva (2017) consideram o contexto do curso de Ciência da Computação da UFERSA (Universidade Federal Rural do Semi-Árido), que tem entrada semestral, nos semestres de 2010.2 a 2014.1. Um levantamento relativo à aprovações, reprovações e desistências na disciplina de OAC foi realizado entre esses semestres. Na Figura 1 pode-se observar uma predominância de uma taxa de reprovação de no mínimo 50%.

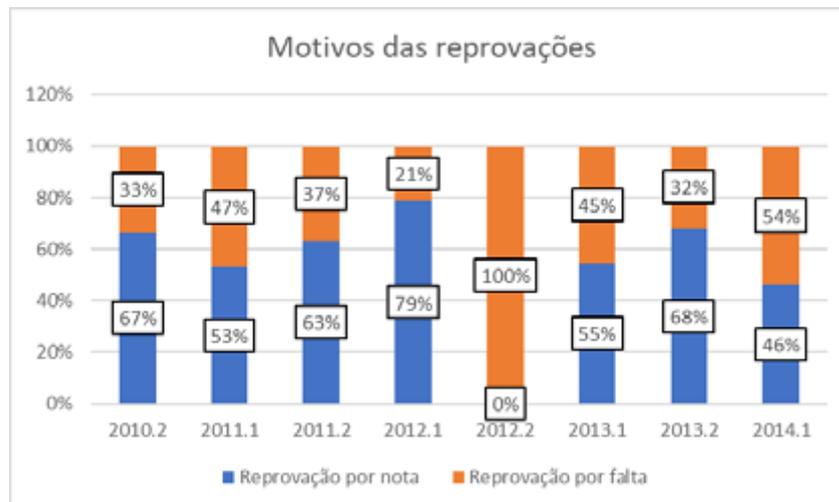
Figura 1 – Taxa de aprovação e reprovação em OAC entre os nos semestres de 2010.2 a 2014.1



Fonte: (FERNANDES; SILVA, 2017)

A Figura 2 apresenta os motivos das reprovações (por nota ou por falta). A média geral das reprovações por falta (46%), mesmo sendo menor que por nota (54%), é preocupante pois significa que quase metade dos reprovados na verdade desistiram da disciplina antes do seu término (FERNANDES; SILVA, 2017).

Figura 2 – Motivos das reprovações em OAC entre os nos semestres de 2010.2 a 2014.1



Fonte: (FERNANDES; SILVA, 2017)

Nesse sentido, é comum a adoção de abordagens complementares como a aplicação de metodologias que recorrem às tecnologias computacionais para facilitar o aprendizado. Um exemplo de tecnologia utilizada são os simuladores, que são *softwares* criados com o intuito de aumentar o nível de abstração dos conteúdos e assim facilitar a compreensão dos mesmos. Além disso, os simuladores permitem a redução dos custos em termos financeiros e de tempo para montar uma infraestrutura para atividades práticas (ESMERALDO; LISBOA, 2017). Outro exemplo de tecnologia é a Realidade Aumentada (RA) que, pela forma de utilização interativa e rica em elementos visuais, tem sido aplicada no campo da educação, como pode ser observado no trabalho de Akçayir e Akçayir (2017).

No contexto do ensino de OAC, os simuladores têm sido empregados tanto no mundo, como se observa no trabalho de Akram e Sawalha (2019), quanto na UFERSA, onde o simulador MARS (VOLLMAR; SANDERSON, 2005) é usado para auxiliar no ensino dos conteúdos da disciplina. Como, em geral, os cursos baseiam-se na aprendizagem de arquiteturas padronizadas tal como o MIPS (*Microprocessor without Interlocked Pipeline Stages*), para facilitar o aprendizado dos relacionamentos entre os componentes de um computador, é possível encontrar muitos simuladores que visam auxiliar o ensino de OAC. Dentre os quais pode-se citar: MARS, WEBMIPS (BRANOVIC; GIORGI; MATINELLI, 2004), VISIMIPS (KABIR; BARI; HAQUE, 2011) e DIMIPSS (FELIX; POUSA; CARVALHO, 2006). Embora esses simuladores proporcionem ao aluno uma noção desses relacionamentos, a não integração desses simuladores com um material de estudo básico da disciplina (livro didático) diminui a efetividade desses *softwares*. Como eles não foram

desenvolvidos para ser usados com um livro texto como uma ferramenta de apoio, quando os alunos os utilizam geralmente têm de lidar com diferenças na representação e apresentação de informações entre eles e o livro texto. Essas diferenças podem ser em relação ao caminho de dados, a nomenclatura e a ordem de apresentação das informações.

A tecnologia RA possibilita a "ligação" entre o simulador e o livro didático. A RA consiste na sobreposição em tempo real de elementos virtuais no mundo real por meio do uso de um dispositivo que permite a manipulação e visualização desses elementos (KIRNER; SISCOOTTO, 2007). Segundo Akçayir e Akçayir (2017), nos últimos anos, tem aumentado significativamente a quantidade de trabalhos que aplicam RA na educação, devido a muitas vantagens obtidas com a aplicação dessa tecnologia, a exemplo da melhora do aprendizado; facilitação do entendimento; aumento da motivação para aprender, do nível de engajamento, do interesse e da satisfação dos alunos; promoção do auto-aprendizado, do aprendizado multissensorial e do aprender fazendo; visualização de conceitos invisíveis, eventos e conceitos abstratos; e redução do custo do material de laboratório. No entanto, há poucos trabalhos que aplicam RA para o ensino da computação, em especial voltados para o ensino de arquitetura de computadores. Para identificar a aplicação da RA na educação foi conduzida uma revisão sistemática (Ver Capítulo 3), na qual não se encontrou nenhum trabalho que usa RA para ensinar sobre o processador MIPS. Conseqüentemente, até a presente data, não foram encontradas aplicações que utilizam RA no ensino do processador MIPS e que sejam integradas com materiais didáticos já consolidados.

Outro fator que tem se destacado nas práticas educativas são as intervenções de gamificação. Por meio do uso de pensamentos, abordagens e elementos de jogos aplicados no contexto educacional, a gamificação leva ao aumento da motivação dos alunos por engajamento nas atividades da disciplina, obtidos com o uso de intervenções individuais ou em grupo, acabando com o absenteísmo causador dos baixos índices de desempenho que disciplinas como OAC costumeiramente apresentam (FERNANDES; SILVA, 2017).

Diante do exposto, neste trabalho apresenta-se o desenvolvimento do simulador ARtEMIS para auxiliar no ensino/aprendizado de OAC que apresenta integração com material didático consolidado e integra RA e gamificação. O simulador é voltado para dispositivos móveis e, por meio da RA, reconhece algumas das figuras do livro “Organização e Projeto de Computadores” (PATTERSON; HENNESSY, 2005), cujos autores foram os vencedores do Prêmio Turing 2017¹ pelo MIPS/RISC e por serem pioneiros numa abordagem sistemática e

¹ <https://amturing.acm.org/byyear.cfm>

quantitativa para o design e avaliação de arquiteturas de computadores. Esse livro é uma das referências mais utilizadas no ensino de arquitetura de computadores – usado na USP (Universidade de São Paulo), UNICAMP (Universidade Estadual de Campinas), MIT (Massachusetts Institute of Technology), Universidade de Oxford, ETH Zurich, Universidade de Cambridge e Universidade Columbia – e apresenta, de forma didática e incremental, o caminho de dados do processador MIPS. O simulador permite que, à medida que estuda pelo livro, o aluno faça o reconhecimento das figuras e interaja com o modelo interativo (modelos 3D) presentes na RA para facilitar a assimilação dos conceitos apresentados tendo como possibilidades a execução de instruções e obtenção de informações sobre o funcionamento das unidades funcionais do processador MIPS. Além disso, elementos de jogos, como fases, pontos e conquistas foram usados para tornar o ARtEMIS um simulador gamificado e estimular ainda mais o seu uso.

Na validação realizada foi verificado que o ARtEMIS é capaz promover melhora no desempenho dos alunos. Os participantes da validação o consideraram muito útil para auxiliar no aprendizado de OAC e consideraram que houve aumento de aprendizado, com seu uso. A usabilidade também foi bem avaliada. Além disso, eles consideraram importante a inclusão da RA e da gamificação para estimular o aprendizado.

1.1 Objetivos

Este trabalho tem como objetivo geral desenvolver um simulador que integra RA e gamificação para ser usado em conjunto com o livro (PATTERSON; HENNESSY, 2005), para auxiliar no aprendizado da arquitetura do processador MIPS. Para alcançar o objetivo geral foram definidos os seguintes objetivos específicos:

- Analisar ferramentas e *frameworks* para desenvolvimento de soluções de RA.
- Desenvolver marcadores para reconhecimento de objetos pela RA.
- Desenvolver o simulador baseado em RA e gamificação.
- Desenvolver uma metodologia de uso do simulador.
- Analisar os benefícios da aplicação da RA no campo da educação.
- Analisar os benefícios da aplicação da gamificação no campo da educação.

1.2 Metodologia

Inicialmente realizou-se uma Revisão Sistemática da Literatura (RSL) nas bases ACM Digital Library e IEEE Xplore com o objetivo de conhecer a atual situação da aplicação da RA no campo da educação. Foram investigados os níveis de escolaridade e áreas de conhecimento em que a RA tem sido aplicada, as estratégias de avaliação do efeito do emprego da RA no aprendizado dos alunos e as estratégias que permitem que eles avaliem a efetividade da RA para melhorar o aprendizado. Também foi realizada uma revisão bibliográfica com o intuito de familiarizar-se com os conceitos necessários para a realização do trabalho, conhecer os trabalhos relacionados ao que é proposto aqui e identificar ferramentas para o desenvolvimento de aplicações RA para dispositivos móveis com Sistema Operacional (SO) Android.

O ARtEMIS foi implementado usando-se as ferramentas mais apropriadas dentre as identificadas na revisão bibliográfica. Ao término da implementação foram realizados testes que permitiram identificar e corrigir os erros levando a uma versão estável ARtEMIS.

Por fim, o ARtEMIS foi validado com alunos da UFERSA do campus Mossoró. A validação foi realizada visando verificar sua capacidade em promover melhora no aprendizado e obter *feedback* dos alunos sobre ele, em termos de usabilidade e utilidade para o aprendizado de OAC, bem como sobre as estratégias empregadas em seu desenvolvimento.

1.3 Organização do Trabalho

Esse trabalho encontra-se estruturado em capítulos. No Capítulo 1 foram apresentados a contextualização e o objetivo da dissertação. No Capítulo 2, apresentam-se os conceitos de organização e arquitetura de computadores, RA, gamificação e os trabalhos relacionados. No Capítulo 3, apresenta-se uma revisão sistemática sobre a aplicação da RA na educação que foi realizada entre Agosto/2017 e Dezembro/2017. No Capítulo 4 é apresentado o simulador desenvolvido. No capítulo 5, descreve-se como a validação foi realizada e apresentam-se os resultados obtidos. E, por fim, no Capítulo 6, apresentam-se as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta a teoria na qual o trabalho está embasado. A primeira seção apresenta os conceitos de organização e arquitetura de computadores, a segunda destaca o processador MIPS, a terceira apresenta o conceito de RA, a quarta trata da gamificação e a quinta exibe os trabalhos relacionados.

2.1 Organização e Arquitetura de Computadores

Organização e arquitetura de computadores são conceitos importantes na computação pois fazem a relação entre o *hardware* e o *software*. Arquitetura se refere aos atributos de um sistema que são visíveis ao programador, isto é, aqueles que impactam diretamente sobre a execução lógica de um programa e organização se preocupa com a forma com que os componentes de *hardware* estão interconectados para execução de instruções dos programas (STALLINGS, 2009). Por exemplo, é no projeto da arquitetura do computador que se decide se ele vai ou não ter uma instrução de multiplicação. Já no projeto de sua organização define-se como essa instrução será implementada (STALLINGS, 2009).

Na arquitetura são considerados aspectos como o conjunto de instruções (do inglês, *Instruction Set Architecture – ISA*), registradores, tamanho da palavra, tipos de dados (por exemplo, números, caracteres) e os modos de endereçamento. Já a organização foca na elaboração do controle e do caminho de dados, definindo como os elementos de processamento e armazenamento de dados são implementados.

2.2 O Processador MIPS

O MIPS (*Microprocessor without Interlocked Pipeline Stages – microprocessador sem estágios intertravados de pipeline*) é um processador criado por John Hennessy em 1981 na Universidade de Stanford. Ele e o RISC I CPU, foram os primeiros com arquitetura RISC (*Reduced Instruction Set Computing – computador com conjunto de instruções reduzido*). Ao contrário dos processadores CISC (*Complex Instruction Set Computer – computador com um conjunto complexo de instruções*), o MIPS e demais processadores RISC não são microprogramados o que significa que as instruções são executadas diretamente pelo

hardware em vez de interpretadas (TANENBAUM, 2007). A partir de 2006, mais de 2 bilhões de microprocessadores MIPS foram fabricados e são encontrados dispositivos que vão desde videogames e *palmtops* a impressoras a laser e *switches* de rede (PATTERSON; HENNESSY, 2013).

A arquitetura MIPS tem passado por inovação constante ao longo das 3 últimas décadas e é a arquitetura RISC mais eficiente do setor, oferecendo o melhor desempenho e menor consumo de energia em uma determinada área de silício. Compõe a família MIPS: nanoMIPS Architecture, MIPS32 Instruction Set Architecture (ISA), MIPS64 Architecture ISA, microMIPS ISA, MIPS Multi-Threading architecture module, MIPS Virtualization architecture module, MIPS SIMD architecture module, MIPS DSP architecture module, MIPS MCU architecture module e MIPS16e architecture module (MIPS, 2019).

A arquitetura do MIPS é baseada em registrador o que significa que os operandos necessários a execução de operações lógicas e aritméticas sempre devem estar armazenados em registradores quando essas operações estão sendo executadas. A versão MIPS32, que será considerada neste texto, possui um banco de registradores que contém 32 registradores de 32 *bits*. Esses registradores são agrupados em categorias de acordo com a finalidade que possuem conforme apresenta-se na Tabela 1. Outra característica do MIPS é o número limitado de formatos de instrução (3 no total), todas com o mesmo tamanho, também de 32 *bits*.

Tabela 1 – Conjunto de registradores do MIPS

Nome	Número	Uso
\$zero	0	O valor constante 0
\$at	1	Registrador reservado para o montador
\$v0	2	Armazenam o resultado de procedimentos
\$v1	3	
\$a0	4	Registradores para armazenar argumentos de procedimentos
\$a1	5	
\$a2	6	
\$a3	7	
\$t0	8	Registradores temporários que não precisam ser preservados entre chamadas de procedimentos
\$t1	9	
\$t2	10	
\$t3	11	
\$t4	12	
\$t5	13	
\$t6	14	
\$t7	15	
\$s0	16	Registradores temporários que precisam ser preservados entre chamadas de procedimentos
\$s1	17	
\$s2	18	
\$s3	19	
\$s4	20	
\$s5	21	
\$s6	22	
\$s7	23	
\$t8	24	Registradores temporários que não precisam ser preservados entre chamadas de procedimentos
\$t9	25	
\$k0	26	Registradores reservados para o <i>Kernel</i> do sistema operacional
\$k1	27	
\$gp	28	<i>Global pointer</i> – ponteiro global
\$sp	29	<i>Stack pointer</i> - aponta para endereço alocado mais recentemente na pilha
\$fp	30	<i>Frame pointer</i> - aponta para a primeira palavra do frame de pilha do procedimento
\$ra	31	Contém o endereço de retorno de uma chamada de procedimento

Fonte: Adaptada de (PATTERSON; HENNESSY, 2005)

2.2.1 Arquitetura do Conjunto de Instruções

Na arquitetura do conjunto de instruções do MIPS cada instrução é representada pelo mnemônico que a identifica seguido pelos parâmetros necessários à sua execução. Na Tabela 2 (“Formato”, “Op” e “Func” são explicados na seção 2.2.2), mostra-se parte dessas instruções agrupadas em categorias de acordo com os tipos de operações que executam (PATTERSON; HENNESSY, 2005). As instruções aritméticas são as que realizam as operações de adição, subtração, divisão e multiplicação no processador; as instruções de transferência de dados são responsáveis por levar dados da memória para o banco de registradores e vice-versa; as instruções lógicas são responsáveis por executar operações booleanas entre operandos e de deslocamento de bits, que movem os bits de um operando para a esquerda ou para a direita de sua posição original; as instruções de desvio podem mudar a sequência de execução do programa. Essas instruções podem alterar o ponteiro que indica a próxima instrução a ser executada para que ele aponte para a instrução indicada pela instrução de desvio, fazendo com que o programa passe a ser executado a partir dela. Nas de desvio condicional o desvio só ocorre se a condição for atendida. Já nas de desvio incondicional a mudança na sequência de execução sempre ocorre.

Tabela 2 – Conjunto de instruções básicas da arquitetura MIPS

Categoria	Nome	Sintaxe	Formato	Op	Func
Aritmética	add	add\$ \$rd, \$rs, \$rt	R	0	32
	subtract	sub \$rd, \$rs, \$rt	R	0	34
	add immediate	addi \$rt, \$rs, immediate	I	8	-
	add unsigned	addu \$rd, \$rs, \$rt	R	0	33
	subtract unsigned	subu \$rd, \$rs, \$rt	R	0	35
	add immediate unsigned	addiu \$rt, \$rs, immediate	I	9	-
Transferência de dados	load word	lw \$rt, offset(\$rs)	I	35	-
	store word	sw \$rt, offset(\$rs)	I	43	-
	load byte	lb \$rt, offset(\$rs)	I	32	-
	store byte	sb \$rt, offset(\$rs)	I	40	-
	load halfword	lh \$rt, offset(\$rs)	I	33	-
	store halfword	sh \$rt, offset(\$rs)	I	41	-
	load upper immediate	lui \$rt, immediate	I	15	-
Lógica	AND	and \$rd, \$rs, \$rt	R	0	36
	OR	or \$rd, \$rs, \$rt	R	0	37
	NOR	nor \$rd, \$rs, \$rt	R	0	38
	OR immediate	ori \$rt, \$rs, immediate	I	13	-
	AND immediate	andi \$rt, \$rs, immediate	I	12	-
	shift left logical	sll \$rd, \$rt, shamt	R	0	0
	shift right logical	srl \$rd, \$rt, shamt	R	0	2
Desvio condicional	branch if equal	beq \$rs, \$rt, address	I	4	-
	branch if not equal	bne \$rs, \$rt, address	I	5	-
	set less than	slt \$rd, \$rs, \$rt	R	0	42
	set less than immediate	slti \$rt, \$rs, immediate	I	10	0
	set less than unsigned	sltu \$rd, \$rs, \$rt	R	0	43
	set less than immediate unsigned	sltiu \$rt, \$rs, immediate	I	11	0
Desvio incondicional	jump	j address	J	2	-
	jump register	jr \$rs	R	0	8
	jump and link	jal address	J	3	-

Fonte: Adaptada de (PATTERSON; HENNESSY, 2005)

2.2.2 Formato de Instruções

Uma instrução de um processador é um conjunto de *bits* que é interpretado por ele para saber qual operação deve ser feita, sobre quais operandos, o que fazer com o resultado e

como encontrar a próxima instrução a ser executada em seguida. Esse formato binário das instruções também é chamado de linguagem de máquina, que são quase incompreensíveis para humanos. Para aumentar a abstração, os projetistas de processadores também oferecem uma linguagem simbólica ou mnemônica equivalente a linguagem de máquina. Desse modo, humanos conseguem entender e programar mais facilmente, podendo usar nomes para instruções, registradores e posições na memória ao invés de códigos e endereços. A linguagem simbólica também é chamada de linguagem de montagem (do inglês *assembly*), a qual serve de entrada para um *software* (montador ou *assembler*) que a converte para a linguagem de máquina a ser executada no processador alvo.

No MIPS o que determina em quais campos o conjunto de *bits* será dividido é o formato da instrução que pode ser R, I ou J (PATTERSON; HENNESSY, 2005). Como mostrado na Tabela 3, em todos esses formatos os primeiros 6 *bits* representam o *opcode* da instrução. Já os demais *bits* têm significados que variam de acordo com o formato.

Tabela 3 – Formatos de instruções da arquitetura MIPS.

Tamanho do campo	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
Formato R	op	rs	rt	rd	Shamt	func
Formato I	op	rs	rt	endereço/imediato		
Formato J	op	endereço de destino				

Fonte: Adaptada de (PATTERSON; HENNESSY, 2005)

No MIPS todas as instruções possuem 32 *bits* que se subdividem em campos com diferentes significados. Segundo Patterson e Hennessy (2005), esses campos são:

- op: este campo tem 6 *bits* de largura e armazena o código da operação básica da instrução, chamado de *opcode*. Esse campo é usado para definir a instrução que será executada e os sinais que o controle deve enviar para as demais unidades funcionais;
- rs: este campo tem 5 *bits* de largura e armazena o endereço do registrador do primeiro operando de origem da instrução;
- rt: este campo tem 5 *bits* de largura e armazena o endereço do registrador do segundo operando de origem, no caso de instruções do formato R (*add*, por exemplo), ou o endereço do registrador do operando de destino, em instruções do formato I (*load*, por exemplo).
- rd: este campo tem 5 *bits* de largura e armazena o endereço do registrador do operando de destino da instrução;

- *shamt* (*shift amount* – quantidade de deslocamento): este campo tem 5 *bits* de largura e indica a quantidade em que os *bits* contidos no registrador *rt* serão deslocados para a direita ou para a esquerda de sua posição original.
- *func*: este campo tem 6 *bits* de largura e seleciona a variante específica da operação no campo *op*.

As instruções do formato R (*Register*), possuem três campos de 5 *bits* para endereçar os três registradores usados em seu processamento. Os dois primeiros registradores contêm os dados que serão processados e o terceiro recebe o resultado do processamento da operação. Além desses campos, esse formato conta com o campo *shamt*, que recebe um valor de 0 a 31 que indica a quantidade de bits que o valor do registrador *rt* será deslocado, e o campo *func* que é usado pelo controle da Unidade Lógica e Aritmética (ULA) para coordená-la. Alguns exemplos de instruções do formato R são *add*, *sub*, *and*, *or* e *nor*.

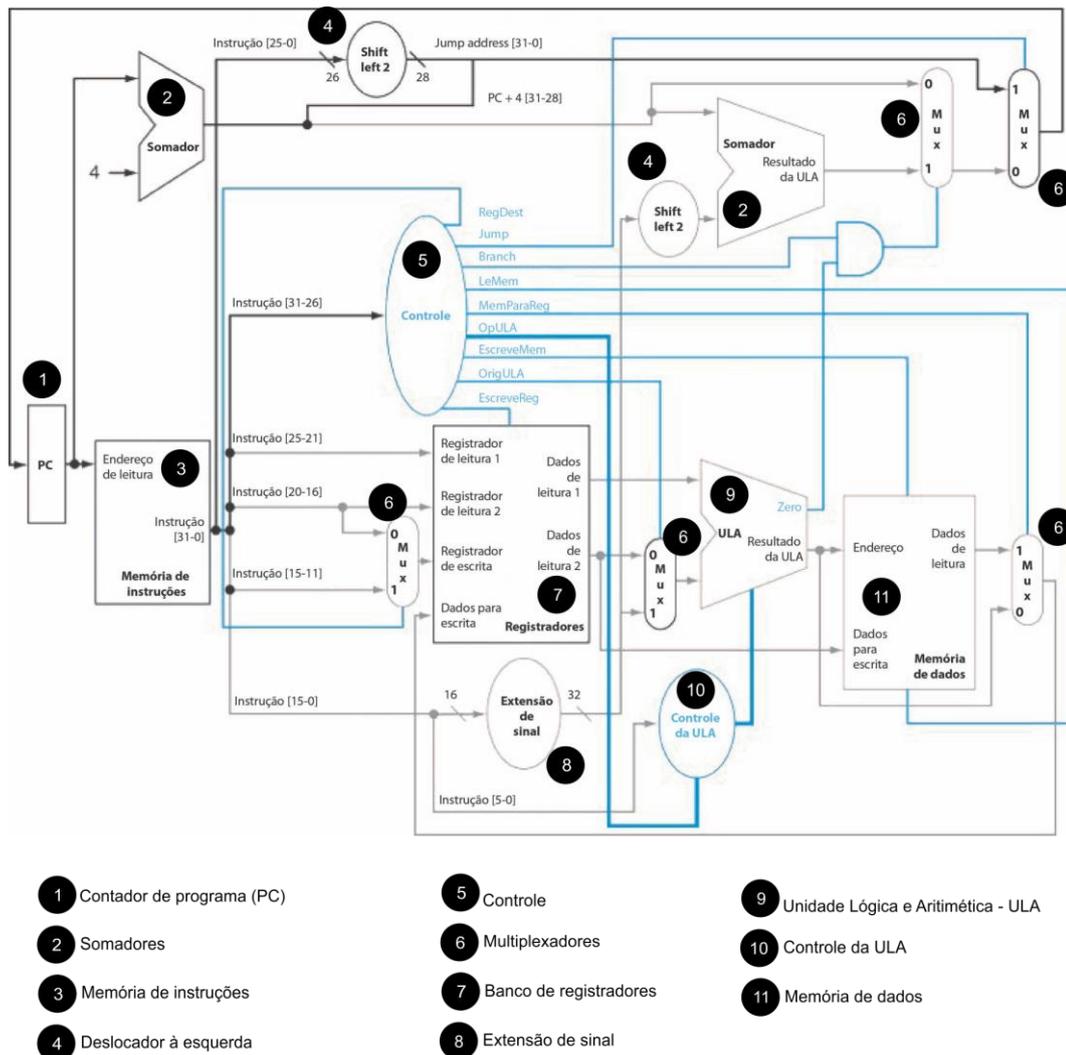
As instruções do formato I (*Immediate*), são chamadas assim porque trabalham com leitura de informações diretamente no código da instrução. Essas informações imediatas estão contidas nos últimos 16 *bits* da instrução. Esses 16 *bits* representam ou uma constante ou de um endereço de memória, o que é determinado pelo tipo da instrução. Há também dois campos de 5 *bits* responsáveis por armazenar os endereços de dois registradores. Nas instruções de desvio *beq* e *bne* esses dois registradores contêm os operandos de origem; na instrução *lui* o *rt* armazena o resultado do processamento e o registrador *rs* não é utilizado; nas instruções que transferem dados da memória para o banco de registradores o registrador *rs* contém parte do endereço de memória acessado e o registrador *rt* recebe o dado trazido da memória; nas instruções que transferem dados do banco de registradores para a memória de dados o registrador *rs* contém parte do endereço de memória acessado e o registrador *rt* contém o dado que é levado para memória; nas demais instruções desse formato o registrador *rs* contém o operando de origem e o registrador *rt* armazena o resultado do processamento. Outros exemplos de instruções desse formato são *addi*, *slli*, *and*, *lw* e *sw*.

As instruções do formato J (*Jump*), realizam desvio incondicional (sem a necessidade de uma condição seja satisfeita) no fluxo normal de execução do programa passando a executá-lo a partir da instrução contida em um endereço especificado. Nesse formato, todos os bits, exceto os *bits* de *opcode*, são utilizados para especificar o endereço da instrução para onde irá ocorrer o desvio. As instruções desse formato são *j* e *jal*.

2.2.3 Caminho de dados MIPS

O caminho de dados do MIPS é formado por um conjunto de unidades funcionais interligadas por meio de barramentos por onde transitam os sinais de dados e os sinais de controle. De acordo com (PATTERSON; HENNESSY, 2005), as unidades funcionais consistem em dois tipos diferentes de elementos lógicos: os elementos que operam sobre os valores dos dados e os elementos que possuem estado. Os elementos que operam sobre valores são combinacionais, o que significa que os valores de saída dependem unicamente dos valores de entrada, como por exemplo a ULA, os somadores e os multiplexadores. Os elementos que contêm estado são aqueles que possuem armazenamento interno, sendo responsáveis por armazenar as informações que serão utilizadas pelas unidades combinacionais. As memórias de dados, memória de instruções e banco de registradores são exemplos desse tipo de elemento.

Figura 3 – Caminho de dados do processador MIPS. Em azul estão destacadas as partes do caminho por onde transitam os sinais de controle com destaque para os barramentos ligados ao controle da ULA



Fonte: Adaptada de (PATTERSON; HENNESSY, 2005)

As unidades funcionais presentes na arquitetura MIPS são apresentadas na Figura 3 e brevemente descritas a seguir:

- Contador de Programa (PC): é um registrador que contém o endereço da próxima instrução a ser executada. Ele repassa esse endereço para a memória de instruções, local em que são armazenadas todas as instruções do programa. Ao final do ciclo da instrução o valor do PC é atualizado seja recebendo seu valor atual incrementado, seja recebendo o endereço indicado pelo desvio para o caso de execução de uma instrução de desvio. No caso do MIPS, o incremento é feito em 4 unidades, uma vez que a memória é endereçada *byte a byte* e cada instrução tem o tamanho de 4 *bytes* (32 *bits*).
- Somadores: somadores são ULAs que possuem apenas a função de soma de sinais. Os somadores recebem como entrada dois sinais de dados de 32 *bits* e retornam como saída a soma desses sinais.
- Memória de instruções: é uma memória que tem como finalidade específica o armazenamento de instruções. Ela recebe como entrada o endereço de instrução contido no PC e tem como saída os 32 *bits* da instrução contida neste endereço. Na seção de decodificação, esses bits são divididos em campos de acordo com o formato da instrução e esses campos são enviados para as demais unidades funcionais para que a operação especificada na instrução seja executada.
- Deslocador à esquerda (*Shift left*): essa unidade tem como função executar o deslocamento dos *bits* do sinal de dados que recebe como entrada para a esquerda preenchendo a posição dos *bits* que se tornaram vazios com zeros. Na prática, isso é o mesmo que multiplicar o valor contido no sinal de entrada por 2^i onde i é a quantidade de *bits* deslocados.
- Controle: a unidade de controle tem como função enviar às demais unidades funcionais os sinais que definem como elas devem operar. O sinal de entrada da unidade de controle é o *opcode* da instrução. A saída da unidade de controle são os sinais que controlam as demais unidades.
- Multiplexadores: os multiplexadores são seletores que têm como função selecionar, dentre diferentes sinais, qual será utilizado pelo sistema. Eles recebem como entrada os sinais de dados e um sinal de controle que permite determinar qual dos sinais de dados recebido será transmitido pela saída.
- Banco de registradores: contém 32 registradores com 32 *bits* cada. As entradas do banco de registradores são três entradas de endereço de 5 *bits* cada, uma de sinal de

dados e uma de sinal de controle. Das entradas de endereços, duas delas recebem os endereços dos registradores (registradores de leitura) cujos valores são lidos em paralelo e a outra recebe o endereço do registrador (registrador de escrita) cujo valor é modificado na execução da instrução. A entrada de sinal de dados recebe 32 *bits* que são armazenados no registrador de escrita. O sinal de controle determina se a operação de escrita deve ser realizada sobre o registrador escrita. O banco apresenta também duas saídas de sinal de dados de 32 *bits* por onde saem os valores contidos nos registradores de leitura.

- Extensão de sinal: essa unidade transforma o sinal de dados de 16 *bits* de largura que recebe como entrada em um sinal de dados de 32 *bits* de largura. Ela faz isso adicionando cópias do valor do *bit* mais significativo do sinal de entrada à esquerda desse *bit* até preencher os 16 *bits* disponíveis.
- Unidade Lógica e Aritmética (ULA): realiza as operações de soma, subtração, multiplicação, divisão e as comparações lógicas. As entradas da ULA são duas entradas de sinal de dados de 32 *bits* por onde recebe os dados dos registradores de leitura (uma dessas entradas pode receber os dados diretamente da memória de instruções no caso de instruções do formato I) que serão processados e uma entrada de sinal de controle de 4 *bits* por onde recebe o sinal que definem a operação a ser realizada. As saídas são uma de sinal de dados de 32 *bits* por onde sai o resultado do processamento e uma de sinal de controle de 1 *bit* que indica se esse resultado foi zero.
- Controle da ULA: esta unidade é responsável por enviar para a ULA o sinal de controle de 4 *bits* que define como ela deve operar. Ela define esse sinal com base em um sinal de controle de 2 *bits* recebido da unidade de controle que, caso tenha valor 2, faz-se necessário também a análise dos *bits* do campo *func* da instrução para definir o sinal para a operação específica.
- Memória de dados: é a memória que armazena os dados que devem ser persistidos ao término da execução da instrução. As entradas da memória de dados são duas de sinal de controle que controlam as operações de leitura e escrita de dados, uma de endereço que indica uma posição da memória que será acessada e uma de sinal dados de 32 *bits* por onde recebe dados para serem armazenados. Como saída ela possui uma saída de sinal de dados de 32 *bits* por onde sai o dado contido na posição da memória indicada pelo endereço fornecido.

2.3 Realidade Aumentada

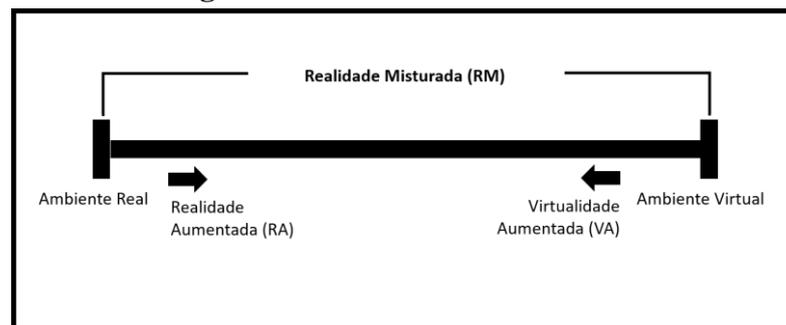
A seguir, são apresentados, alguns conceitos relativos a RA. Mais detalhes sobre o uso da RA para desenvolvimento do simulador proposto encontram-se descritos no Capítulo 4.

RA é uma tecnologia que sobrepõe imagens do mundo real capturadas por uma câmera em tempo real. Para Kirner e Kirner (2011), a RA pode ser definida como o enriquecimento do mundo real por meio de informações virtuais (imagens dinâmicas, sons espaciais, sensações hápticas) que são geradas por computador em tempo real e devidamente posicionadas no espaço 3D, percebidas por meio de dispositivos tecnológicos.

Azuma (1997) define RA como um sistema que apresenta três características: combina o real com o virtual; é interativo em tempo real; e ajusta os objetos virtuais no ambiente 3D.

Para Milgram e Kishino (1994), a RA juntamente com a Virtualidade Aumentada (VA) forma o que eles chamam de Realidade Misturada (RM). Segundo esses autores a RM está localizada entre o ambiente real e o virtual (Figura 4) e consiste em ambientes onde elementos virtuais e reais coexistem. Próximo ao extremo onde se localiza o ambiente virtual está a Virtualidade Aumentada (VA), na qual prevalece o mundo virtual. Próximo ao extremo onde se localiza o ambiente real está a Realidade Aumentada (RA), na qual os elementos reais predominam no ambiente. O que caracteriza a RA é o enriquecimento do mundo real por informações virtuais. Já na VA o mundo virtual é enriquecido por informações reais.

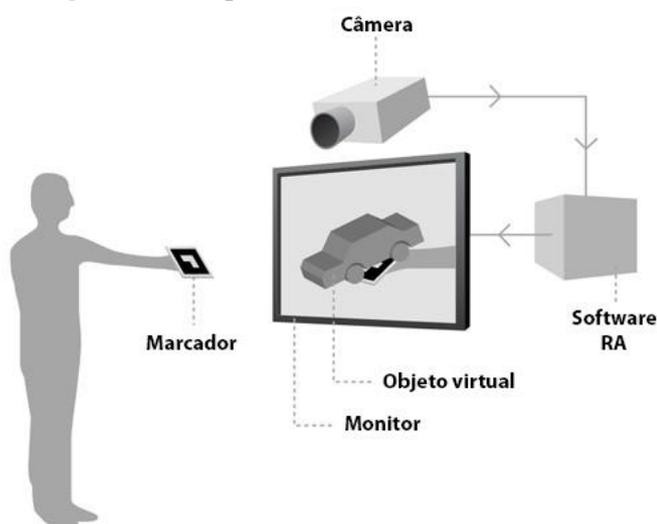
Figura 4 – Contínuo de virtualidade



Fonte: Adaptada de (MILGRAM; KISHINO, 1994)

O funcionamento da RA envolve alguns elementos básicos, os quais consistem em monitor, câmera, *software* de RA e alvos (símbolos ou marcadores, objetos 3D, geolocalização). Alguns desses elementos são mostrados na Figura 5.

Figura 5 – Esquema de funcionamento da RA



Fonte: Adaptada de (BIANCHINI; SILVA, 2014)

Como se pode ver na Figura 5, o marcador é o elemento no qual os objetos virtuais são sobrepostos. Para isso, a câmera captura a imagem do marcador e a envia para um *software* que realiza o processamento necessário e, em seguida, envia para o monitor a imagem com os objetos virtuais sobrepostos ao marcador.

2.4 Gamificação

Gamificação (do original em inglês *gamification*) é um conceito introduzido em 2002 por Nick Pelling e se popularizou em 2010. Tal conceito se refere ao emprego de elementos de jogo em ambientes não jogáveis. Seu principal objetivo é aumentar a participação e gerar engajamento e comprometimento entre um público específico (VIANNA et al., 2013).

Seaborn e Fels (2015) listam os seguintes elementos como mecanismos de jogos normalmente ligados à gamificação: pontos, crachás, tabelas de liderança, progressão, status, níveis, recompensas e papéis. Esses mecanismos são brevemente apresentados na Tabela 4.

A gamificação tem sido empregada em vários campos como: educação e interação homem-máquina (SEABORN; FELLS, 2015), *marketing* e saúde (LEE; HAMMER, 2011), entretenimento e negócios (HERVAS et al., 2017), entre outros. Para Signori e Guimarães (2016) o emprego da gamificação no campo educacional leva a modernização das metodologias de ensino para adaptá-las ao perfil das novas gerações, que mesmo pertencendo à era digital e estando imersas em um mar de inovações tecnológicas, das quais fazem um uso intensivo, são expostas a modelos de didática defasados.

Tabela 4 – Elementos de jogos ligados à gamificação

Mecanismo	Definição
Pontos	Indicadores numéricos que indicam progresso.
Crachás	Elementos visuais que simbolizam conquistas.
Tabelas de liderança	Exposição da classificação para comparação.
Progressão	Marcos que indicam progresso.
<i>Status</i>	Alcunhas que indicam progresso.
Níveis	Ambientes com dificuldade crescente.
Recompensas	Itens tangíveis e desejáveis.
Papéis	Personagens de jogo.

Fonte: Autoria própria

Subhash e Cudney (2018) conduziram uma revisão sistemática para identificar o estado da aplicação da gamificação no nível superior. Os achados revelam que houve um crescimento da aplicação nos últimos anos e também que a maioria dos estudos foi realizada na área da computação. Além disso, foram identificados como benefícios mais significativos o aumento do envolvimento e da motivação dos alunos e a melhora da atitude (sob a forma de maior esforço, participação, confiança e interesse) e do desempenho deles.

2.5 Trabalhos relacionados

Nesta seção apresentam-se alguns simuladores que, assim como o proposto neste trabalho, trabalham o caminho de dados do MIPS32, provendo uma visualização das sequências de interações entre suas unidades funcionais. Também é mostrada uma abordagem RA para o ensino de OAC e outra para o ensino de circuitos. Outros simuladores de OAC podem ser encontrados no trabalho de Akram e Sawalha (2019).

2.5.1 MARS

O MARS (*Mips Assembly and Runtime Simulator*) é um simulador desenvolvido na linguagem de programação Java (DEITEL; DEITEL, 2009) que simula o conjunto de instruções do MIPS32. O MARS disponibiliza um editor de texto e um montador² MIPS o que permite ao usuário fornecer um código *assembly*, que tem sua sintaxe verificada pelo

² Montador é um *software* que converte um programa escrito em linguagem de montagem (*assembly*) em um código que pode ser entendido por um computador - código de máquina.

montador e, caso esteja correto, é possível fazer sua simulação. Durante a simulação, o usuário pode controlá-la, tendo as opções de executar todas as instruções de uma vez, uma por uma, ou ainda, escolher a quantidade de instruções executadas por segundo. Além disso, existem as opções de pausar, retroceder e parar a execução. Os segmentos de memória (código e dados) e os registradores são visíveis ao usuário em tempo de execução e as informações neles atualizadas são destacadas de forma colorida para percepção do usuário. Os conteúdos da memória de dados dos registradores podem ser editados pelo usuário durante a simulação a qualquer momento, quando as instruções são executadas uma por uma, ou quando a execução é pausada, se o usuário escolher executar todas as instruções de uma vez.

Uma extensão (ARAÚJO; PÁDUA; CORRÊA JUNIOR, 2014) disponível para as versões a partir da 4.5, possibilita ainda, a visualização do caminho de dados mostrando, por meio da alteração das cores dos barramentos, a sequência em que os dados são transferidos entre as unidades funcionais. Contudo, essa extensão não exibe os valores desses dados.

2.5.2 WEBMIPS

O WEBMIPS é um ambiente Web³ escrito em ASP⁴ que tem como objetivo a simulação da arquitetura MIPS com *pipeline*⁵ para parte do conjunto de instruções do MIPS. Ele disponibiliza um editor para que o usuário insira o código que deseja simular e um montador que traduz esse código em código de máquina. O usuário pode controlar a simulação escolhendo entre executar o código etapa por etapa, em que pode seguir o avanço das instruções em cada estágio do *pipeline*, ou todas as etapas de uma vez. Na simulação, além do diagrama esquemático do caminho de dados, são visíveis ao usuário a memória de instruções, a memória de dados e o banco de registradores. Os conteúdos dos registradores e da memória de dados são atualizados em tempo de execução, e na memória de instruções as instruções em execução recebem rótulos coloridos com o nome do estágio do *pipeline* em que se encontra. Há ainda a opção de clicar sobre qualquer componente do caminho de dados para visualizar seus valores de entrada e saída. Ao término da execução, o número total de ciclos

³ Ambiente de sistemas computacionais que normalmente utilizam páginas HTML como interface com usuário. Estas páginas são visualizadas através do *browser* - software para visualização de páginas HTML/Web.

⁴ Linguagem de script que permite criar sites dinâmicos. Acrônimo de *Active Server Pages* (Páginas Ativas de Servidor).

⁵ *Pipeline* (técnica de segmentação de instruções em múltiplos ciclos).

de *clock*⁶ é exibido. Diferente do MARS, em que o usuário pode alterar os conteúdos da memória de dados e dos registradores via interface amigável de usuário, no WEBMIPS só é possível alterá-los por meio de código *assembly*.

2.5.3 VISIMIPS

O VISIMIPS é um simulador desenvolvido em Java voltado para a simulação dos cinco estágios do funcionamento do *pipeline* do MIPS32. Ele suporta parte do conjunto de instruções MIPS e tem um editor que permite ao usuário inserir código *assembly* e um montador que traduz esse código em linguagem de máquina. Na simulação é possível avançar/retroceder a execução de uma instrução por vez e visualizar em qual posição no *pipeline* as instruções em execução se encontram. O caminho de dados é visível em tempo de execução e o usuário pode visualizar os valores contidos nos barramentos passando o *mouse* sobre eles. Diferente dos simuladores já citados, não é possível visualizar as informações da memória de dados e dos registradores.

2.5.4 DIMIPSS

O DIMIPSS (*Didact Interactive MIPS Simulator*) é um *software* desenvolvido em Java para simulação da execução de parte das instruções do MIPS monociclo, ou seja, organização do MIPS para que suas instruções sejam executadas em um único cliço de *clock*. Ele tem um editor que recebe um programa em *assembly* e um montador que converte esse programa para a linguagem de máquina. Após a conversão é possível simular a execução das instruções do código uma por uma. Na simulação é possível visualizar o caminho de dados, os segmentos de memória (dados e instruções) e o banco de registradores e acompanhar as atualizações de seus conteúdos que são destacadas em cinza. Para facilitar o entendimento do usuário, os barramentos que possuem sinais usados durante a execução da instrução também são destacados para cada instrução através das cores azul (para o caminho de dados) e vermelho (para o caminho de controle). Semelhante ao WEBMIPS, é possível visualizar os valores de entrada e de saída das unidades funcionais e os estados dos sinais da unidade de controle bastando, para tanto, passar o *mouse* sobre eles.

⁶ Sinal usado para coordenar as ações de dois ou mais circuitos eletrônicos. Métrica que contabiliza os intervalos básicos de tempo nos quais são executadas as operações elementares de uma instrução.

2.5.5 DRMIPS

O DRMIPS (NOVA; ARAÚJO; FERREIRA, 2013) é um simulador que permite tanto a simulação do MIPS monociclo quanto *pipeline*. Ele suporta parte do conjunto de instruções MIPS e contém um editor de código e um montador que traduz as instruções do MIPS em código de máquina. Após a montagem o usuário pode realizar a simulação executando todas as instruções de uma vez ou uma por uma. Na simulação é possível visualizar o caminho de dados, os segmentos de memória (dados e instruções) e o banco de registradores. Os conteúdos dos registradores e da memória são atualizados em tempo de execução destacando de forma colorida para percepção do usuário. Os conteúdos da memória de dados e dos registradores podem ser editados pelo usuário via interface amigável de usuário durante a simulação. No caminho de dados é possível visualizar os valores contidos nos barramentos. Além da versão multiplataforma para PCs ele apresenta uma versão para dispositivos móveis com SO Android. Contudo, os recursos de visualização e usabilidade (por exemplo, nem todos os botões/ícones/menus/conteúdo de uma tela estão acessíveis o tempo todo sendo necessário alternar entre eles) tornam o uso desse simulador desmotivante. A falta de integração com um material de apoio e de recursos para avaliação dos alunos, deixam-no limitado enquanto ferramenta didática.

2.5.6 MIPSFPGA

O MIPSFPGA (PENHA; FONTES; FERREIRA, 2016) é baseado no simulador Hades (HENDRICH, 2000) como camada básica e é um simulador com interface gráfica para visualização do caminho de dados das implementações do MIPS com e sem *pipeline*. Ele segue a sequência didática do livro de Patterson & Hennessy (PATTERSON; HENNESSY, 2005), na qual os autores propõem o ensino MIPS em partes e de forma incremental. Ao carregar o projeto no MIPSFPGA, o estudante tem acesso a uma figura interativa do livro. A cada passo da simulação (ciclo de *clock*), os fios e barramentos podem ter seus valores monitorados. Assim como nos outros simuladores apresentados é possível ver o conteúdo dos registradores, memória e avançar ciclo a ciclo. O MIPSFPGA permite ao estudante editar graficamente o projeto e derivar uma nova implementação com mais instruções. Além disso, permite a resolução de exercícios propostos pelo livro – alguns exercícios do livro foram remodelados para adquirirem o formato de múltipla escolha e estão disponibilizados com o

MIPSFPGA – e de exercícios propostos pelos autores e exportar o projeto para implementação física em FPGA.

2.5.7 RaVi

O RaVi (MARWEDEL; SIROCIC, 2003) é composto por um conjunto de projetos e componentes que executam no simulador Hades (HENDRICH, 2000). O RaVi possibilita visualizar o funcionamento do MIPS multiciclo e *pipeline*, do protocolo MESI, do algoritmo de Tomasulo e do algoritmo Scoreboarding. Com relação a simulação do MIPS, há suporte para parte do conjunto de instruções MIPS e é disponibilizado um editor que permite ao usuário inserir código *assembly* e um montador que traduz esse código em linguagem de máquina. O usuário pode controlar a simulação executando as instruções etapa por etapa. O caminho de dados é visível em tempo de execução e o usuário pode visualizar os valores contidos nos barramentos passando o *mouse* sobre eles. Também é possível visualizar as informações da memória de dados e dos registradores.

2.5.8 UCOMIPSIM 2.0

O UCOMIPSIM 2.0 (GERSNOVIEZ et al., 2018) é um simulador em JAVA para a arquitetura do MIPS32. Com ele é possível visualizar graficamente o caminho de dados de várias instruções do MIPS passando pelos diferentes estágios do *pipeline*. Para fornecer instruções o usuário pode carregar um arquivo contendo um programa. O usuário também pode analisar e modificar a memória de dados e o banco de registradores. São dadas as opções de inicializar, pausar e reiniciar a simulação e executá-la passo a passo. Há também uma barra para regular a velocidade de execução da simulação. Quando a execução de um programa é simulada, as memórias de instrução e de dados e o banco de registradores podem ser manipulados diretamente da interface do simulador. Cada instrução tem uma cor associada e o caminho de dados correspondente para cada uma é destacado na mesma cor. É possível visualizar os valores de entrada e de saída das unidades funcionais clicando sobre elas. Feito isso é aberta uma nova janela, mostrando uma imagem mais detalhada do componente contendo esses valores.

2.5.9 RA para ensinar sobre a arquitetura do processador Neander

Brum, Pinho e Camargo (2017) recorreram a RA para criar um objeto de aprendizagem (OA) para o ensino da arquitetura do processador Neander – Ele foi o primeiro computador hipotético criado com o intuito de facilitar a aprendizagem de OAC. Suas características incluem: dados e endereços de 8 bits; dados representados em complemento de dois; Acumulador (AC) de 8 bits e PC de 8 bits (BRUM; PINHO; CAMARGO; 2017) – A ideia foi permitir ao aluno visualizar em diferentes níveis (nível de blocos, registradores e portas lógicas) de abstração o funcionamento e as interações entre os elementos básicos da organização do processador. No entanto, os autores não mencionam se o OA criado o usuário possibilita que o usuário insira código para visualizar sua execução.

2.5.10 VirtualComponent

Kim et al. (2019) desenvolveram o VirtualComponent, um sistema interativo de realidade mista que permite aos usuários construir um circuito *breadboard* combinando componentes físicos e virtuais. Usando uma interface de RA, os usuários podem colocar componentes virtuais em uma *breadboard* e modificar seus valores e propriedades no *software*. Essas ações são refletidas imediatamente no nível elétrico na *breadboard* física.

2.6 Considerações Finais

Este capítulo foi dedicado ao referencial teórico, abordando o conteúdo teórico que embasa o desenvolvimento deste trabalho. Foi apresentada uma revisão sobre OAC e sobre o processador MIPS. Também foram apresentados os conceitos de RA e gamificação. Além disso, foram apresentados alguns trabalhos presentes na literatura que são relacionados à proposta deste trabalho. Na Tabela 5, esses trabalhos são comparados em relação as características consideradas relevantes. Como se pode observar o único que apresenta todas as características consideradas é o ARtEMIS que é detalhado no Capítulo 4.

Tabela 5 – Comparação dos trabalhos relacionados com o simulador ARtEMIS

	MARS	WEBMIPS	VISIMIPS	DIMIPSS	DRMIPS	MIPSFPGA	RaVI	UCOMIPSIM 2.0	OA	VirtualComponent	ARtEMIS
Simula o processador MIPS?	X	X	X	X	X	X	X	X			X
Permite visualizar o caminho de dados?	X	X	X	X	X	X	X	X	X		X
Usa RA?									X	X	X
Usa gamificação?											X
É integrado com material didático já consolidado?											X
É voltado para dispositivos móveis?					X						X

Fonte: Autoria própria

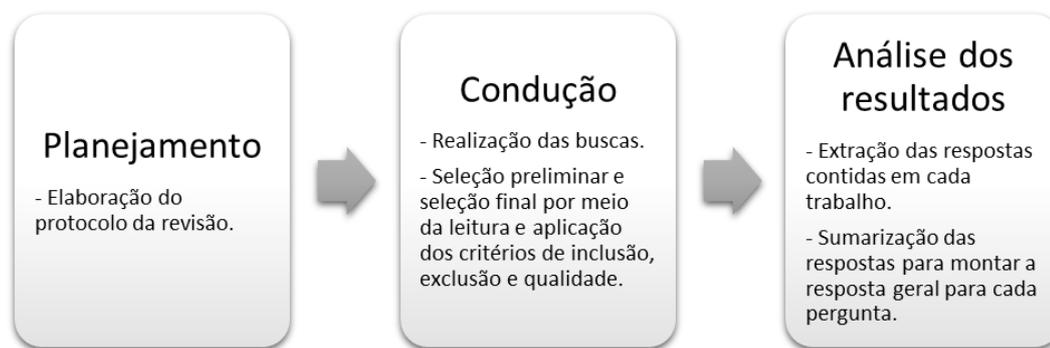
3 USO DE REALIDADE AUMENTADA NA EDUCAÇÃO

Este capítulo apresenta e descreve uma revisão da literatura realizada para identificar a atual situação da abordagem RA no ambiente educacional. Ela foi conduzida com os seguintes objetivos: verificar a distribuição dos trabalhos que aplicam RA na educação por áreas de conhecimento e níveis de escolaridade; conhecer as estratégias utilizadas para verificar o impacto da aplicação da RA no desempenho dos alunos, bem como as estratégias para avaliação das RAs pelos alunos e sua efetividade como ferramenta facilitadora do aprendizado.

3.1 Metodologia

A metodologia utilizada foi a Revisão Sistemática da Literatura (RSL), que de acordo com Kitchenham (2004), é uma forma de identificar, avaliar e interpretar toda a pesquisa disponível relevante para uma questão de pesquisa específica, área temática ou fenômeno de interesse. Para a condução da RSL foi seguido o modelo de revisão apresentado em (BIOLCHINI et al., 2007) composto pelas etapas de planejamento da revisão, condução da revisão e análise dos resultados conforme mostrado na Figura 6 e detalhado a seguir.

Figura 6 – Etapas seguidas na condução da revisão



Fonte: Autoria própria

3.1.1 Planejamento da revisão

O planejamento da revisão consiste na definição de seu protocolo que especifica os métodos que serão usados para realizá-la. Para tanto, o modelo de protocolo proposto por

Biolchini et al. (2007) foi tomando como base. Nele devem ser definidos os objetivos da revisão, questões de pesquisa, estratégia de busca, critérios de inclusão, exclusão e qualidade e o procedimento de seleção dos estudos primários.

Os objetivos da RSL foram os já citados. A partir deles foram definidas as seguintes questões de pesquisa:

- i) Em quais áreas de conhecimento e níveis de escolaridade a RA tem sido mais aplicada?
- ii) Quais estratégias de avaliação têm sido utilizadas para verificar o impacto da aplicação da RA no desempenho dos alunos?
- iii) Quais estratégias têm sido utilizadas para permitir aos alunos avaliar a efetividade da RA como ferramenta facilitadora do aprendizado?

Na estratégia de busca foi definido que as buscas seriam feitas nas bases de dados indexadas ACM Digital Library⁷ e IEEE Xplore⁸ aplicando-se *strings* de busca formadas pelas palavras-chave *augmented reality* e *education*, bem como seus sinônimos apresentados na Tabela 6. Também foi definido que os idiomas dos estudos primários seriam português e inglês. O inglês por ser uma linguagem universal para a escrita de trabalhos acadêmicos de maior impacto; o português para valorização dos trabalhos desenvolvidos por pesquisadores brasileiros e possivelmente o contexto de aplicabilidade mais próximos das nossas pesquisas.

Tabela 6 – Palavras-chave e seus sinônimos

<i>Palavra-chave</i>	<i>Sinônimos</i>
<i>augmented reality</i>	<i>augmenting reality, mixed reality</i> , realidade aumentada e realidade misturada.
<i>Education</i>	<i>teaching, learning</i> , educação, ensino, aprendizado e aprendizagem

Fonte: Autoria própria

Crítérios de inclusão e exclusão devem ser definidos para refinar as buscas para respostas mais precisas às questões de pesquisa. Isso é necessário porque as palavras-chave podem ter diferentes significados ou serem usadas em estudos que não tratam do tema de pesquisa de revisão sistemática. Artigos que se enquadram em, pelo menos, um critério de inclusão são aceitos. Dessa forma, foram definidos os seguintes critérios de inclusão:

- i) Trabalhos que desenvolvem *softwares* RA voltados para o ensino/aprendizado.
- ii) Trabalhos que descrevem experimentos com RA no ensino/aprendizado em ambiente escolar.

⁷ <https://dl.acm.org/>

⁸ <https://ieeexplore.ieee.org/Xplore/home.jsp>

- iii) Trabalhos que avaliam o impacto da aplicação da RA no desempenho dos alunos.
- iv) Trabalhos que permitem aos alunos avaliar a efetividade da RA como ferramenta facilitadora do aprendizado.

Os trabalhos que não se encaixam nos critérios de inclusão, devem se encaixar em algum critério de exclusão para justificar sua retirada da revisão. Os critérios de exclusão definidos são:

- i) Trabalhos que não desenvolvem *softwares* RA.
- ii) Trabalhos que desenvolvem *softwares* RA, mas não voltados para o ensino/aprendizado.
- iii) Trabalhos que descrevem experimentos com RA, mas não no ensino/aprendizado.
- iv) Trabalhos que descrevem experimentos com RA no ensino/aprendizado, mas não em ambiente escolar.
- v) Trabalhos duplicados.

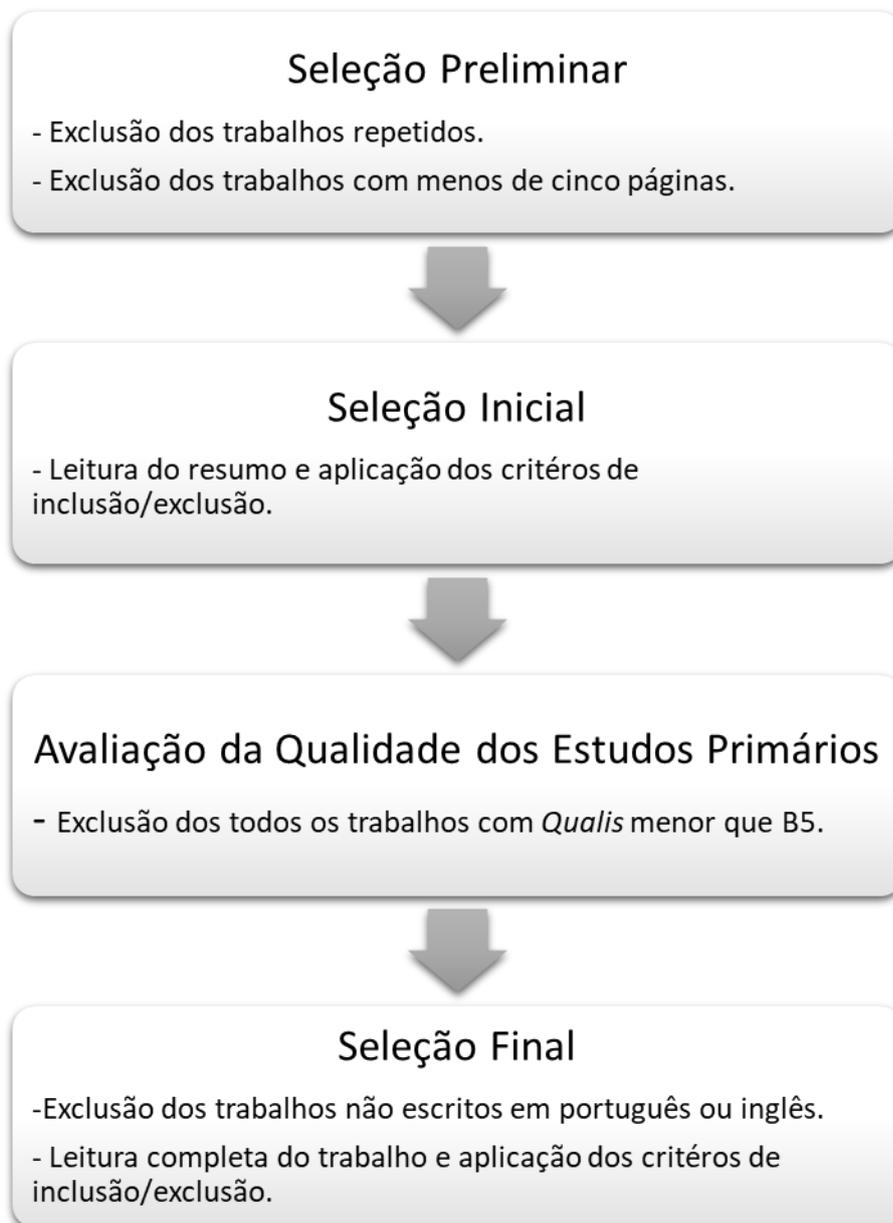
Crítérios de qualidade também são importantes pois ajudam a garantir que o resultado dos estudos primários selecionados seja relevante cientificamente. Para a RSL foram definidos quatro critérios de qualidade:

- i) Trabalhos escritos em português ou inglês.
- ii) Trabalhos com no mínimo cinco páginas.
- iii) Trabalhos publicados a partir de 2007.
- iv) Trabalhos com Qualis, pelos menos, B5 na área de Computação e de Educação.

Após a definição dos critérios de seleção dos estudos é importante definir o processo. No processo de seleção dos estudos primários são especificadas as etapas que serão seguidas para selecionar os trabalhos que respondem as questões de pesquisa. Como se pode observar na Figura 7, a primeira delas é a seleção preliminar, na qual serão excluídos os artigos repetidos e aqueles com menos de cinco páginas. Na segunda etapa, chamada de seleção inicial, é feita a leitura do resumo de cada artigo remanescente da etapa anterior para verificar quais critérios de inclusão eles atendem, para, com base nisso, definir quais passarão para a próxima etapa. Na etapa seguinte, avaliação da qualidade dos estudos primários, é feita a exclusão dos trabalhos com Qualis menor que B5. A última etapa é a seleção final, na qual são eliminados os artigos escritos em outros idiomas que não sejam português ou inglês e é realizada a leitura completa de cada artigo restante para identificar os que, de fato, respondem

a pelo menos uma das questões de pesquisa. Os critérios de inclusão e exclusão são aplicados novamente, de forma semelhante a etapa anterior.

Figura 7 – Etapas do processo de seleção



Fonte: Autoria própria

Para análise dos trabalhos selecionados foi elaborado um formulário de extração de dados com base nas questões de pesquisa dessa RSL, no qual foram anotadas as respostas contidas em cada trabalho. As respostas foram reunidas procurando-se agrupar respostas semelhantes de forma a permitir identificar padrões que formariam respostas gerais para cada uma das questões.

3.1.2 Condução da Revisão

A revisão foi conduzida durante um período de 5 meses (Agosto/2017 a Dezembro/2017), de acordo com o planejamento apresentado na seção anterior. O primeiro passo foi a construção e aplicação das *strings* de busca nas bases de dados.

A busca na IEEE Xplore foi realizada com a aplicação das *strings* de busca para inglês e português, conforme mostrado na Tabela 7. Para a realização das buscas foi utilizada a opção de busca avançada da IEEE que está disponível no endereço eletrônico <http://ieeexplore.ieee.org/search/advsearch.jsp> e nela selecionou-se a opção "Command Search" onde as *strings* foram inseridas.

Tabela 7 – Strings aplicadas na busca realizada na IEEE Xplore

String em inglês	(.QT.augmented reality.QT. OR .QT.augmenting reality.QT. OR .QT.mixed reality.QT) AND (education OR teaching OR learning)
String em português	(.QT.realidade aumentada.QT. OR .QT.realidade misturada.QT.) AND (educação OR ensino OR aprendizado OR aprendizagem)

Fonte: Autoria própria

Para ambas *strings* a busca foi realizada em 27/09/2017. A aplicação da *string* em inglês retornou 1108 trabalhos. No entanto, como o interesse é em trabalhos publicados a partir de 2007, foi aplicado o filtro do ano de publicação restando, com isso, 995 trabalhos. Com a *string* em português encontrou-se um trabalho que foi incluído por ter sido publicado em 2014.

As *strings* elaboradas para a busca realizada na ACM Digital Library são mostradas na Tabela 8. Elas foram aplicadas no modo de busca avançada, disponível no endereço eletrônico <https://dl.acm.org/advsearch.cfm?coll=DL&dl=ACM&CFID=840229478&CFTOKEN=30137173>.

Tabela 8 – *Strings* aplicadas na busca realizada na ACM Digital Library

<i>String</i> em inglês	acmdlTitle:(+("augmented reality" "augmenting reality" "mixed reality") +(education teaching learning)) OR recordAbstract:(+("augmented reality" "augmenting reality" "mixed reality") +(education teaching learning)) OR keywords.author.keyword:(+("augmented reality" "augmenting reality" "mixed reality") +(education teaching learning))
<i>String</i> em português	acmdlTitle:(+("realidade aumentada" "realidade misturada") +(educação ensino aprendizado aprendizagem)) OR recordAbstract:(+("realidade aumentada" "realidade misturada") +(educação ensino aprendizado aprendizagem)) OR keywords.author.keyword:(+("realidade aumentada" "realidade misturada") +(educação ensino aprendizado aprendizagem))

Fonte: Autoria própria

As buscas na ACM também foram realizadas em 27/09/2017. Com a aplicação da *string* em inglês foram obtidos 391 trabalhos. No entanto, com a aplicação do filtro do ano de publicação, esse número caiu para 333. Já com a *string* em português nenhum trabalho foi encontrado.

Ao todo, 1329 trabalhos (996 da IEEE e 333 da ACM) foram recuperados e submetidos às etapas de seleção preliminar, seleção inicial, seleção final e análise dos resultados.

Na seleção preliminar, os artigos repetidos e aqueles com menos de cinco páginas foram excluídos. Desse modo, 12 trabalhos foram excluídos por se repetirem, e 174 (43 da ACM e 131 da IEEE) por ter menos de cinco páginas. Em seguida, no processo de seleção inicial, foram lidos os resumos de todos os artigos selecionados. Nessa etapa, foram excluídos 910 artigos com base nos critérios de inclusão/exclusão, sendo 671 da IEEE e os outros 239 da ACM.

Na avaliação de qualidade verificou-se que apenas 104 (73 da IEEE e 31 da ACM) dos 233 trabalhos tinham *Qualis* de pelo menos B5, sendo estes submetidos a etapa de seleção final.

Na última etapa, seleção final, os 104 artigos foram lidos na íntegra e verificou-se que deles, apenas 70 (52 da IEEE e 18 da ACM) colaboravam com as respostas das questões de pesquisa sendo estes os únicos selecionados para a extração das informações.

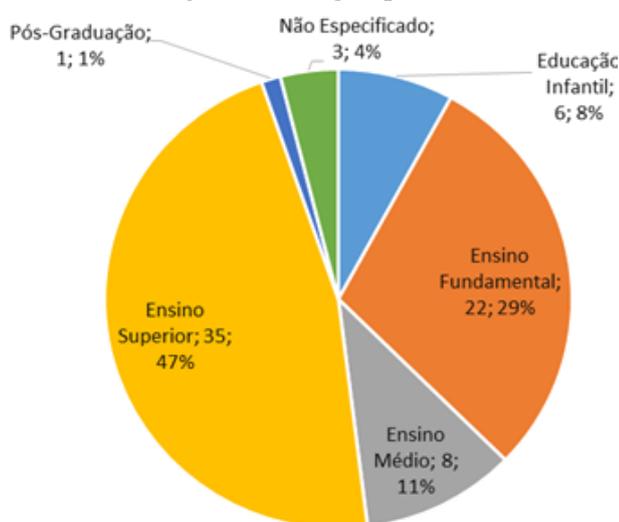
3.2 Resultados da RSL

A seguir são apresentados os resultados obtidos e as respostas encontradas para cada questão de pesquisa.

3.2.1 Questão de Pesquisa 1: Em quais níveis de escolaridade e áreas de conhecimento a RA tem sido mais aplicada?

Essa questão trata de dois pontos: distribuição por níveis de escolaridade e distribuição por áreas de conhecimento. Os resultados obtidos para o primeiro ponto são mostrados na Figura 8. Pode-se ver que o foco está principalmente no ensino superior, alvo de 47% dos trabalhos, e no ensino fundamental, alvo de 29% trabalhos. Para o ensino médio, a educação infantil e a pós-graduação os percentuais são bem menores, 11%, 6%, e 1%, respectivamente. Além disso, 4% dos trabalhos não teve público-alvo específico. Essa distribuição mostra que a RA é aplicável a qualquer nível de escolaridade.

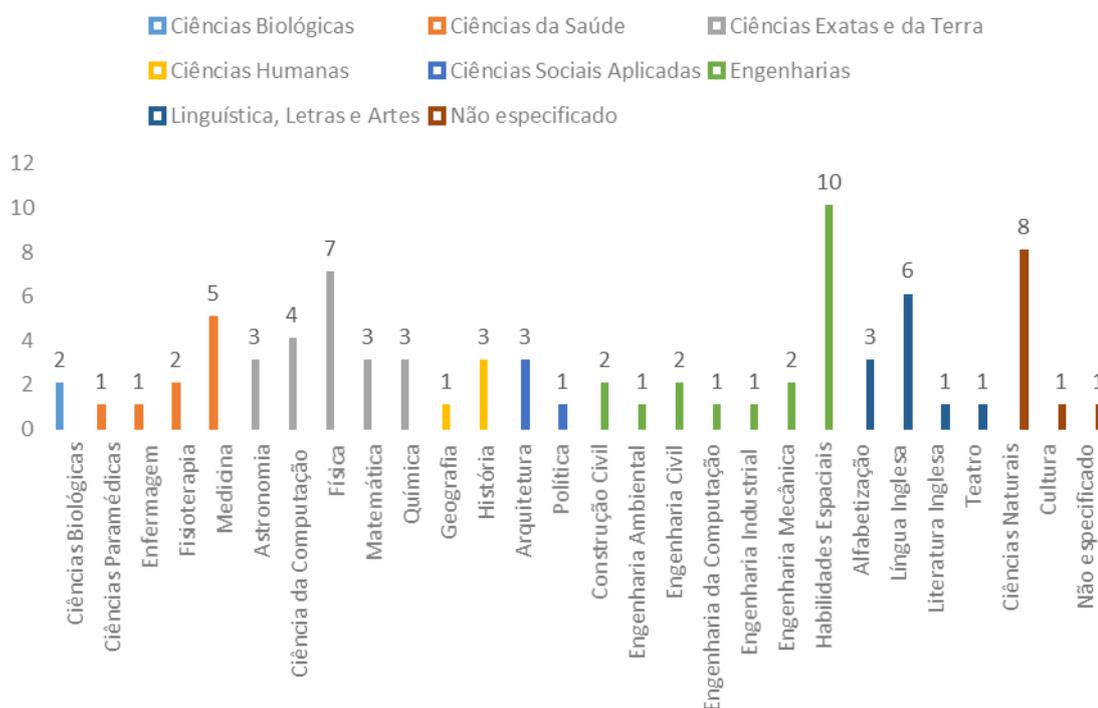
Figura 8 – Distribuição dos artigos por nível de escolaridade



Fonte: Autoria própria

Com relação ao segundo ponto, distribuição por área de conhecimento, os resultados são mostrados na Figura 9. Como pode se ver, a área mais enfatizada foi Habilidades Espaciais - definida em (MARTÍN-GUTIÉRREZ; NAVARRO; GONZÁLEZ, 2011) como a “capacidade de representar formas tridimensionais no olho da mente” - sendo foco de 10 trabalhos. Em segundo lugar, como áreas mais enfatizadas foram, as Ciências Naturais com 8 trabalhos, Física com 7, Língua Inglesa com 6 e Medicina com 5 trabalhos.

Figura 9 – Distribuição dos artigos por área e grande área de conhecimento



Fonte: Autoria própria

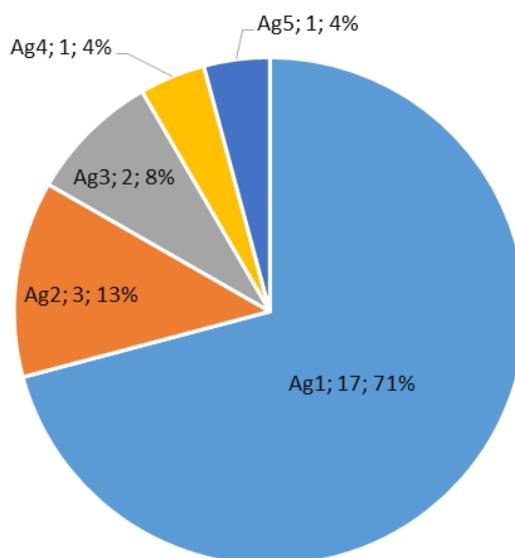
Considerando-se o agrupamento dos trabalhos por grandes áreas (CAPES, 2017), vê-se um domínio das Ciências Exatas e da Terra, com 20 trabalhos, seguida pelas Engenharias com 19, Linguística, Letras e Artes com 11, Ciências da saúde com 9, Ciências Humanas e Ciências Sociais Aplicadas com 4 e Ciências Biológicas com 2. Os trabalhos relacionados a Cultura e Ciências Naturais não foram agrupados. Essas distribuições por área e grande área mostram a aplicabilidade da RA nos mais diversos temas e conteúdos a serem ensinados.

3.2.2 Questão de Pesquisa 2: Quais estratégias de avaliação têm sido utilizadas para verificar o impacto da aplicação da RA no desempenho dos alunos?

Um total de 23 dos 70 trabalhos respondem a essa questão. Na elaboração da resposta foram considerados os seguintes aspectos: forma como os participantes da pesquisa são agrupados, forma como o procedimento de avaliação do desempenho foi realizado e o formato dos testes. Os resultados obtidos para cada um desses aspectos são apresentados a seguir.

Para as formas de agrupamento temos os resultados da Figura 10. Das cinco formas identificadas, Ag1 (grupo de teste e grupo de controle) foi a mais utilizada, presente em 71% dos trabalhos. Em segundo lugar está Ag2 (único grupo realiza tarefas com apenas um tipo de RA), presente em 13% dos trabalhos e em terceiro Ag3 (único grupo realiza tarefas com e sem RA), presente em 8% dos trabalhos. Por último temos Ag4 (2 ou mais grupos, cada um realiza tarefas um tipo de RA) e Ag5 (2 grupos que realizam tarefas tanto com RA quanto sem RA, mas em ordens diferentes), cada uma presente em 4% dos trabalhos.

Figura 10 – Formas de agrupamento identificadas. Ag1 – grupo de teste e grupo de controle; Ag2 – único grupo realiza tarefas com apenas um tipo de RA; Ag3 – único grupo realiza tarefas com e sem RA; Ag4 – 2 ou mais grupos, cada um realiza tarefas um tipo de RA; e Ag5 – 2 grupos que realizam tarefas tanto com RA quanto sem RA, mas em ordens diferentes



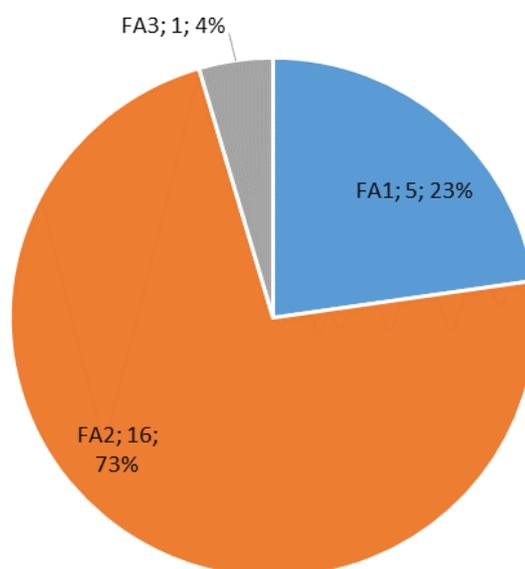
Fonte: Autoria própria

A análise da distribuição das formas de agrupamento em relação aos níveis de escolaridade revela o seguinte: Ag1 foi empregada em todos os níveis; Ag2 foi empregada somente no ensino fundamental; Ag3 na educação infantil e em um trabalho que não tem um nível definido; Ag4 no ensino médio; Ag5 foi empregada juntamente com Ag1 no ensino superior.

Por ser a mais utilizada e ser aplicada em todos os níveis de escolaridade, Ag1 parece ser a forma mais apropriada de agrupamento. Ela é interessante por permitir comparar o desempenho de um grupo que usa RA com um grupo que não usa e verificar o quanto a RA pode influenciar no desempenho de um grupo em relação ao outro. Outras formas interessantes são Ag3 e Ag5, que permitem avaliar o desempenho que o mesmo grupo de alunos obtêm ao realizar tarefas com e sem o uso da RA, sendo que em adicional a Ag5 verifica se a ordem (se RA é utilizada primeiro ou depois) influencia no desempenho. Ag2 é a mais simples já que há apenas um grupo que realiza tarefas somente com RA e parece apropriada para crianças para evitar a sobrecarga de tarefas. Ademais, Ag4 é útil quando se quer testar mais de um tipo de RA para verificar com qual os alunos obtêm melhor resultado.

Os resultados para o segundo aspecto considerado, forma como o procedimento de avaliação do desempenho foi realizado, são exibidos na Figura 11. Dos 23 trabalhos, apenas um não aplicou testes para verificar o efeito de emprego da RA. Eles foram distribuídos nas seguintes formas de aplicação: Apenas Pós-teste (FA1), Pré-teste e Pós-teste (FA2), Pré-teste, Pós-teste e teste de retenção (FA3). FA2 foi empregada em 73% dos trabalhos sendo a mais utilizada. Em seguida vem FA1 empregada em 23% trabalhos e FA3 empregada em 4%.

Figura 11 – Formas de aplicação identificadas



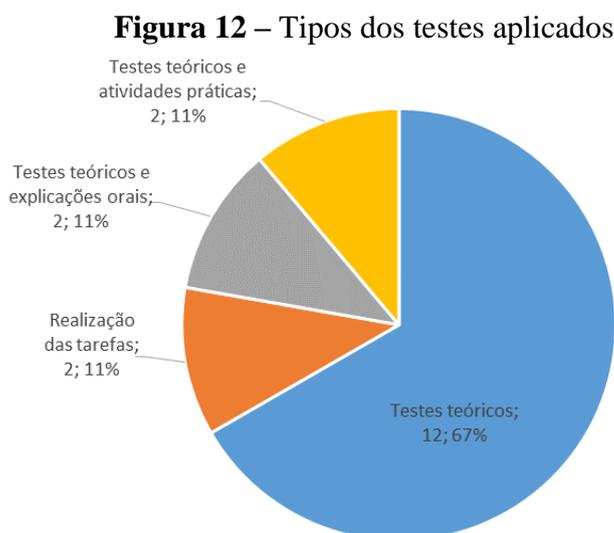
Fonte: Autoria própria

FA2 permitir conhecer o desempenho dos alunos antes e depois de participar do experimento, o que pode justificar o fato dela ser a mais utilizada, pois com isso pode-se verificar se de fato a RA teve efeitos positivos. A sua variação (FA3) também é interessante, pois permite verificar o efeito da RA em termos de fixação de conhecimento. FA1 é menos

interessante já que não permite medir o conhecimento dos alunos antes do uso da RA e assim não é possível saber se de fato houve alguma melhora em termos de aprendizado. Com ela só é possível verificar se um grupo que usou RA obteve um desempenho melhor que outro grupo (de controle ou que também usou RA) ou se um grupo obteve desempenho melhor usando RA ou outra condição (outra tecnologia ou método de ensino tradicional) se um pós-teste for aplicado após cada condição experimentada.

No trabalho que não foi aplicado teste, para verificar a influência da RA os autores compararam os resultados obtidos pelos alunos nas unidades da disciplina em que eles estavam usando RA com os que eles obtiveram nas unidades que foram submetidos ao método tradicional de ensino, ou seja, sem RA.

Para o terceiro aspecto considerado foram extraídas informações de 18 trabalhos. Desses, em 67% os autores aplicaram apenas testes teóricos sendo 1 constituído por questões associativas e os demais por questões de múltipla escolha e/ou discursivas. Em 11%, como os autores usaram RA para treinar os alunos na execução de tarefas, os testes consistiram na realização dessas tarefas. Em outros 11%, além dos testes teóricos os autores analisaram as explicações orais dadas pelos alunos para o fenômeno estudado e nos outros 11% o teste foi composto por testes teóricos e atividades práticas. A sumarização desses resultados pode ser vista na Figura 12.

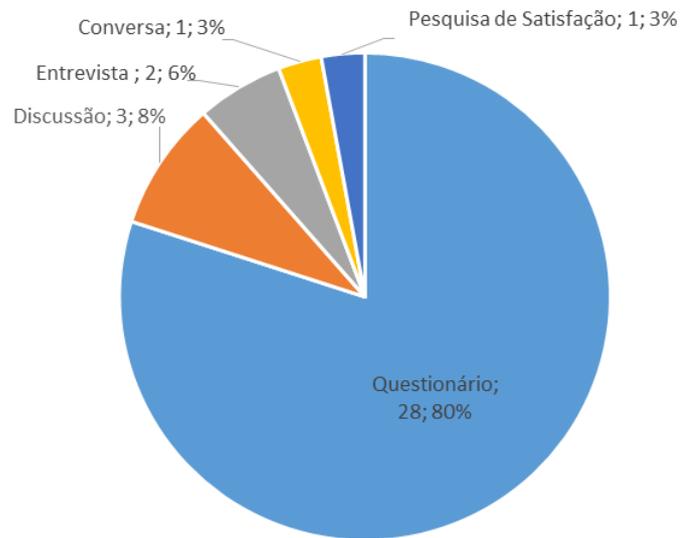


Fonte: Autoria própria

3.2.3 Questão de Pesquisa 3: Quais estratégias têm sido utilizadas para permitir aos alunos avaliar a efetividade da RA como ferramenta facilitadora do aprendizado?

Um total de 33 dos 70 trabalhos respondem a essa questão. Eles foram divididos em duas categorias: os que aplicaram questionários e os que procuraram ouvir os alunos. É importante salientar que alguns trabalhos se enquadram nas duas categorias. Como se pode observar na Figura 13, o questionário foi a estratégia mais usada, presente em 80% dos trabalhos. Desses questionários, apenas 4 não são formados por itens em escala Likert e em 3 deles também haviam questões discursivas. Os outros 20% procuraram ouvir os alunos por meio de discussões, entrevistas, pesquisa de satisfação e conversas.

Figura 13 – Estratégias de Avaliação da RA identificadas



Fonte: Autoria própria

3.3 Considerações finais

Neste capítulo foram apresentados os resultados obtidos da condução de uma RSL sobre a aplicação da RA na educação. As respostas às questões levantadas nessa RSL permitem tirar as seguintes conclusões:

- A RA tem sido aplicada em todos os níveis de escolaridade e há um foco maior no nível superior.
- A RA pode ser usada para ensinar os mais diversos temas e conteúdos, mas apesar disso, a Ciência da Computação, área com vários conteúdos que podem ser trabalhados por meio da RA, foi abordada em apenas 4 trabalhos (3 tratavam do ensino de programação e 1 do ensino de redes).

- A divisão dos alunos em grupo de teste e de controle, aplicação de pré-teste e pós-teste e teste escrito constituem a estratégia mais utilizada para avaliação do efeito da aplicação da RA no aprendizado.
- A estratégia mais empregada para permitir aos alunos avaliar as aplicações RA é o questionário, que em geral, é constituído de itens de escala Likert.

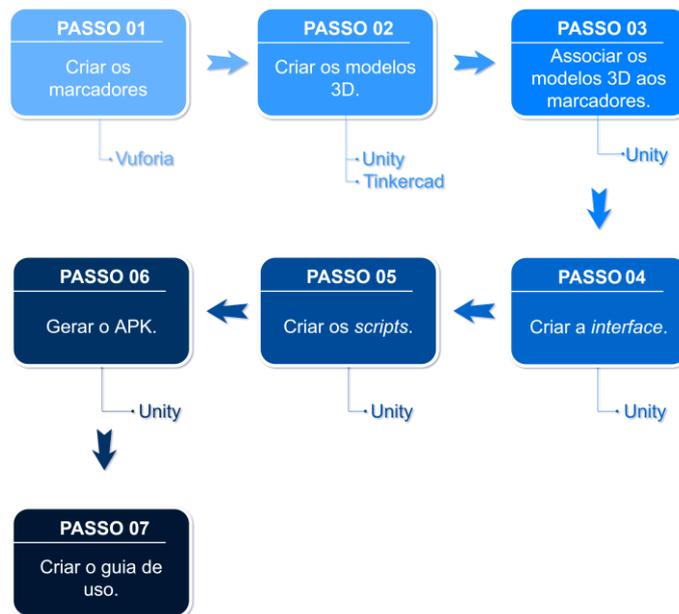
Essas conclusões mostram que esse trabalho está no caminho certo no sentido de propor um simulador que é voltado para um nível de escolaridade no qual a RA tem sido aplicada e que preenche uma lacuna aplicando RA para o ensino de um conteúdo da área de computação que ainda não foi trabalhado por meio dessa abordagem. Elas também revelam as estratégias que podem ser usadas no processo de avaliação do mesmo.

4 O SIMULADOR ARTEMIS

Neste capítulo, trata-se do simulador desenvolvido. Na primeira seção apresenta-se uma visão geral do mesmo. Na segunda, descreve-se como foi feita a implementação, mostrando as ferramentas de *software* e como elas foram utilizadas. Na terceira explica-se como a gamificação foi empregada para inserir elementos de jogos no simulador. E, por fim, na última seção detalha-se o funcionamento do simulador. O *download* do simulador pode ser feito acessando o repositório virtual do GESyCA (*Group of Embedded Systems and Computer Architecture*) – UFERSA.

O desenvolvimento do ARtEMIS seguiu o fluxo apresentado na Figura 14.

Figura 14 – Fluxo de desenvolvimento do ARtEMIS



Fonte: Autoria própria

Um dos passos da implementação é transformar as figuras do livro em marcadores (passo 01). A transformação consiste em convertê-las em *Image Target*, um dos tipos de *Target* (alvo) que pode ser criado no Vuforia e que possibilita transformar uma imagem qualquer em um marcador. Ao transformar uma imagem o Vuforia faz uma análise para determinar o quão reconhecível ela é. Isso é indicado pela quantidade de estrelas (entre 0 e 5) que o Vuforia atribui a tal imagem. As *Image Targets* são armazenadas em um banco de dados que deve ser baixado e importado para o projeto criado no Unity.

No passo 02 são criados os modelos 3D dos componentes do processador. Embora no próprio Unity possam ser criados os modelos 3D de algumas formas geométricas como cubo,

esferas, cilindros e cápsulas, ele não oferece meios para transformá-los em modelos mais elaborados (por exemplo, o modelo 3D de um Somador). Assim, o Tinkercad foi usado para modelar todos os componentes que não podem ser representados pelas formas disponíveis no Unity.

Com os componentes modelados e as figuras transformadas em marcadores, o próximo passo é criar o modelo 3D de cada figura associando os modelos dos componentes aos marcadores (passo 03). O passo 04 é inserir os elementos de GUI (*Graphical User Interface*) e, por fim, são criados *scripts* na linguagem C# (passo 05) para definir o comportamento dos componentes 3D e dos elementos de GUI. Finalizados os passos anteriores, o passo seguinte (passo 06) é gerar o arquivo APK (*Android Application Pack*). Para isso é necessário indicar as cenas que farão parte do aplicativo, selecionar dentre as plataformas disponíveis a plataforma Android, fazer as configurações necessárias (nome do aplicativo, empresa/instituição do aplicativo, ícone do aplicativo, etc) e clicar na opção de gerar o arquivo APK. Feito isso o arquivo APK é gerado e está pronto para ser instalado.

Por fim foi criado o Guia de Uso (passo 07), um documento que contém orientações pedagógicas para a utilização do ARtEMIS no ambiente educacional. Esse guia apresenta informações sobre os conteúdos abordados pelo simulador, bem como descrição de atividades e avaliações que podem ser realizadas, trazendo detalhes sobre preparação, requisitos técnicos, tempo previsto e material necessário para execução das atividades com o ARtEMIS. Em síntese, o Guia de Uso traz sugestões, para professor e aluno, de ações que podem ser realizadas antes, durante e depois do uso da ferramenta.

4.1 Visão Geral

O simulador proposto foi inicialmente denominado ARMS (*Augmented Reality MIPS Simulator*) (SILVA; OLIVEIRA; FERNANDES, 2018), uma vez que foi idealizado para ser usado integrado ao livro (PATTERSON; HENNESSY, 2005) usando RA e gamificação. No entanto, devido a semelhança sonora com a arquitetura ARM decidiu-se “rebatizá-lo” com o nome ARtEMIS (*Aumengetd REality MIPS Simulator*). Na mitologia grega Ártemis é uma deusa ligada à vida selvagem e à caça e também associada à lua e à magia⁹, o que de certa forma pode ser associada a RA.

⁹ <https://pt.wikipedia.org/wiki/%C3%81rtemis>

Como o livro aborda um subconjunto de instruções do processador MIPS e apresenta seu caminho de dados de maneira incremental, por meio de figuras, a ideia é que o aluno possa fazer uso do simulador para visualizar a funcionalidade dos componentes, a execução das instruções e o respectivo controle suportado pelo caminho de dados representado em cada figura do livro, enquanto está lendo-o. O ARtEMIS suporta o MIPS monociclo – Embora não exista MIPS monociclo no mercado, ele é uma opção didática para facilitar a compreensão do MIPS multiciclo e do MIPS *pipeline*. Tal simulador corresponde a um aplicativo voltado para dispositivos móveis com o SO Android. O aplicativo faz uso da câmera do dispositivo para reconhecer as figuras do livro e cria uma RA que usa modelos interativos (modelos 3D) dos caminhos de dados com os quais é possível interagir.

4.2 Implementação

A implementação do ARtEMIS foi realizada usando as ferramentas Unity 2017.2.0¹⁰, Vuforia SDK (*Software Development Kit*) para Unity¹¹, Tinkercad¹², Android SDK¹³ e a linguagem C# (NAKOV; KOLEC, 2013). Para a escolha dessas ferramentas foi realizado um levantamento sobre as características de cada uma e suas contribuições para o desenvolvimento do ARtEMIS.

Foram encontradas várias ferramentas que permitem criar aplicações RA para o SO Android como: Vuforia, Wikitude¹⁴, EasyAR¹⁵, ARKit¹⁶, ARCore¹⁷ e ARToolKitX¹⁸. Para determinar qual delas seria usada, levou-se em conta aspectos como o tipo de licença (gratuita, comercial), suporte à plataforma Android, capacidade de transformar as ilustrações do livro em marcadores, variedade de dispositivos móveis com que tem compatibilidade (ARCore por exemplo, tem compatibilidade apenas com alguns dispositivos de algumas das principais fabricantes do mercado) e documentação disponibilizada. A que atendeu as restrições e necessidades de maneira mais satisfatória foi o Vuforia, e por isso foi escolhida. Vuforia é um SDK para desenvolvimento de aplicações RA que permite desenvolver para as

¹⁰ <https://unity3d.com/pt/>

¹¹ <https://www.vuforia.com/>

¹² <https://www.tinkercad.com/>

¹³ <https://developer.android.com/>

¹⁴ <https://www.wikitude.com/>

¹⁵ <https://www.easyar.com/>

¹⁶ <https://developer.apple.com/arkit/>

¹⁷ <https://developers.google.com/ar/>

¹⁸ <http://www.artoolkitx.org/docs/about/>

plataformas Android, iOS, UWP e óculos digitais. Ele pode ser integrado ao Android Studio, Xcode, Visual Studio e Unity.

Ao se desenvolver para a plataforma Android, o Vuforia dá as opções de trabalhar com o Android Studio ou com o Unity. Inicialmente a opção escolhida foi o Android Studio devido a autora deste trabalho já ter experiência. No entanto, foram enfrentados problemas na configuração do ambiente de desenvolvimento para os quais não foram encontradas soluções mesmo consultando a documentação disponível. Então, optou-se por trabalhar com o Unity, cujos problemas enfrentados puderam ser solucionados ao consultar documentação disponível. O Unity é um *game engine* para o desenvolvimento de jogos e aplicativos de visualização 3D que possibilita fácil exportação para PC (Windows, Mac, Linux), Android, IOS, UWP, consoles e outros.

Para a modelagem 3D também há muitas ferramentas como: 3ds Max¹⁹, Blender²⁰ e Maya²¹. Contudo, essas ferramentas são pagas e/ou tem um processo de modelagem 3D que requer elevado conhecimento técnico. Então, optou-se por usar o Tinkercad que é gratuito e tem um processo de modelagem 3D simples. O Tinkercad é um ambiente Web para *design* de modelos 3D e também de simulação de circuitos elétricos analógicos e digitais. Ele tem uma abordagem fácil e intuitiva para criar os modelos e permite exportá-los nos formatos .obj e .stl, os quais são usados pelo Unity.

O Android SDK é o kit de desenvolvimento de aplicações Android e é requisitado pelo Unity para a criação de aplicações para Android.

C# é uma linguagem orientada à objetos semelhante a C++ e Java. Ela foi criada e desenvolvida pela Microsoft em conjunto com a plataforma .NET. C# é uma das linguagens que podem ser usadas para criar *scripts* no Unity e foi a escolhida para a implementação do ARtEMIS.

4.3 Proposta de gamificação com ARtEMIS

O ARtEMIS reconhece 4 figuras do livro “Organização e Projeto de Computadores” de Patterson e Hennessy, 3ª edição em português (PATTERSON; HENNESSY, 2005). Essa edição foi escolhida por ser a edição disponível na biblioteca da UFERSA (Universidade Federal Rural do Semi-Árido) do campus Mossoró, cidade em que funciona o Programa de

¹⁹ <https://www.autodesk.com.br/products/3ds-max/overview>

²⁰ <https://www.blender.org/>

²¹ <https://www.autodesk.com.br/products/maya/overview>

Pós-graduação em Ciência da Computação (PPgCC). O simulador não funciona para todas as edições porque há ligeiras modificações nos textos e nos formatos das unidades funcionais das figuras, de uma edição para outra, que impossibilitam o reconhecimento dessas.

O ARtEMIS foi originalmente pensado para ser usado como uma ferramenta auxiliar no ensino/aprendizado do MIPS permitindo ao usuário interagir com qualquer uma das quatro ilustrações em qualquer ordem (SILVA; OLIVEIRA; FERNANDES, 2018). Entretanto, aproveitando a apresentação didática das ilustrações no livro e a conceito de gamificação, cada vez mais presente no âmbito educacional, foram inseridos elementos de jogos no ARtEMIS modificando a forma como o usuário tem acesso ao reconhecimento dessas ilustrações. Os diagramas de casos de uso e de atividades podem ser visualizados no APÊNDICE A – DIAGRAMAS UML.

Na 3ª edição do livro as figuras são respectivamente numeradas como 5.1, 5.2, 5.17 e 5.24, as quais são apresentadas na Figura 15. As figuras do livro serão chamadas nesse texto de “ilustração” para não confundir com as próprias figuras desse trabalho. As ilustrações, são apresentadas em ordem incremental de detalhamento do caminho de dados, inclusive aumentando o suporte de mais instruções.

Figura 15 – Ilustrações 5.1, 5.2, 5.17 e 5.24 do livro texto (PATTERSON; HENNESSY, 2005) utilizadas como marcadores pela RA do ARtEMIS. Em cada ilustração, os componentes acrescentados em relação a ilustração anterior estão destacados em vermelho.

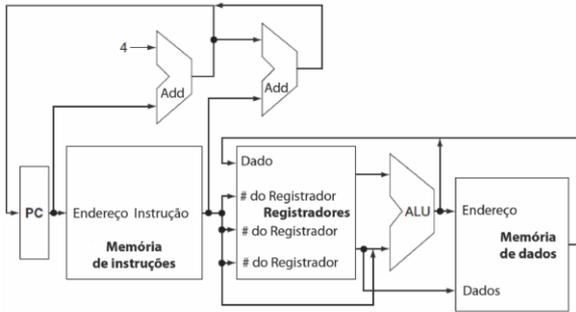


Ilustração 5.1

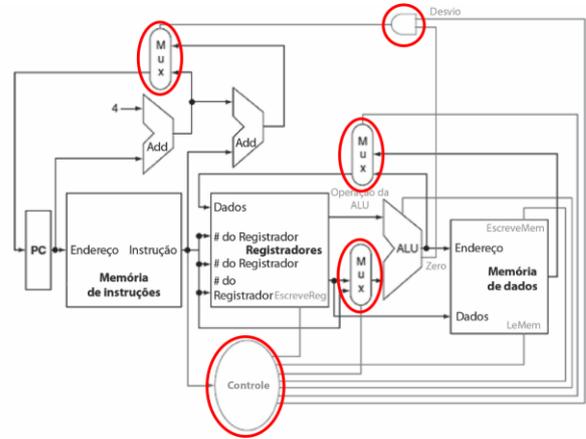


Ilustração 5.2

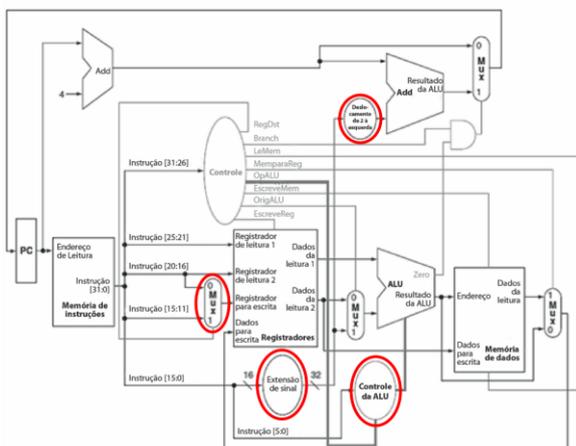


Ilustração 5.17

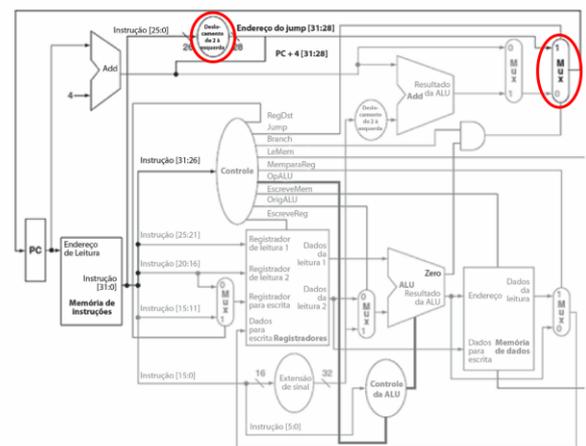
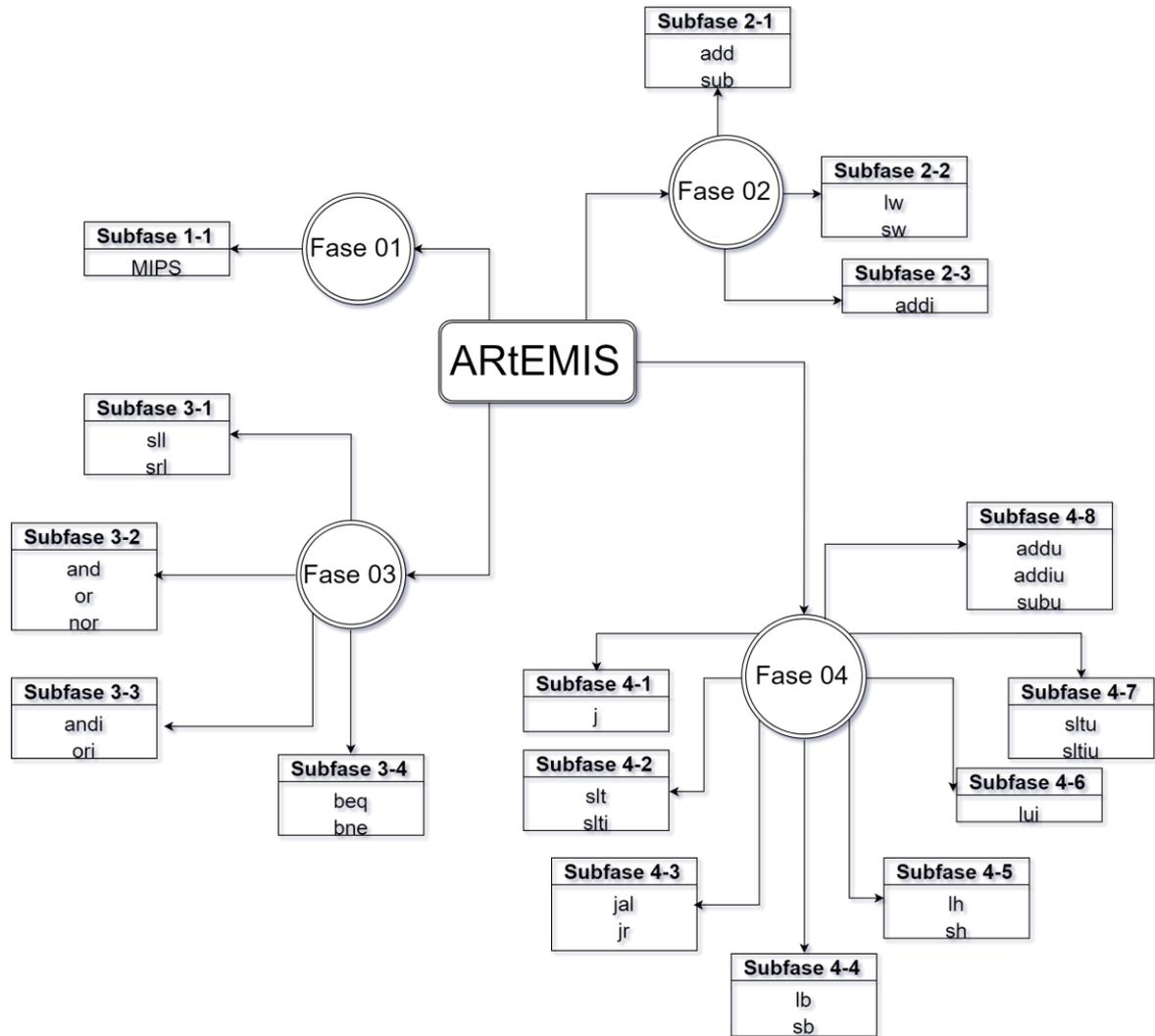


Ilustração 5.24

Fonte: Autoria própria

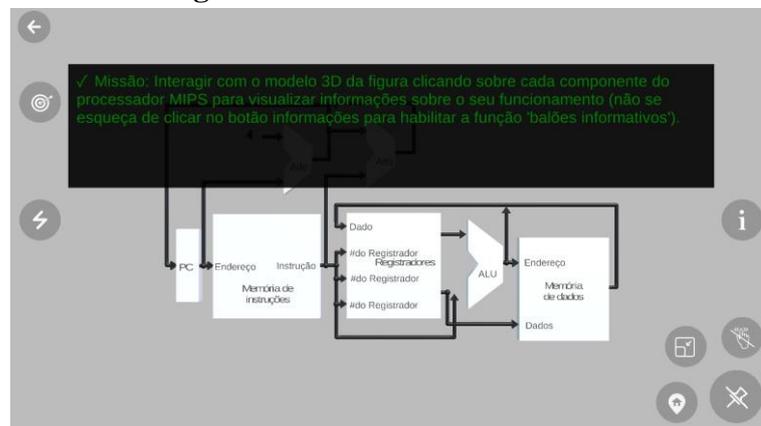
Cada uma das quatro ilustrações tornou-se uma fase do ARtEMIS contendo uma ou mais subfases. Na Figura 16 são apresentadas as fases, subfases e instruções abordadas no ARtEMIS. Cada instrução ou conjunto de instruções constitui uma subfase. Em cada subfase há uma missão (Figura 17) em que o usuário precisa cumpri-la para liberar o desafio relacionado à subfase. Os desafios são questões relacionadas a missão ou as instruções/componentes apresentados na subfase. As questões podem ser questões de verdadeiro ou falso, múltipla escolha ou para configurar os sinais do caminho de controle (Ver Figura 24 e Figura 25). Quando cumpre o desafio, acertando todas as questões, o usuário tem acesso a próxima subfase e quando o usuário conclui todas as subfases de uma fase a próxima fase é liberada.

Figura 16 – Subfases que constituem as fases com as instruções que abordam



Fonte: Autoria própria

Figura 17 – Missão da subfase 1-1



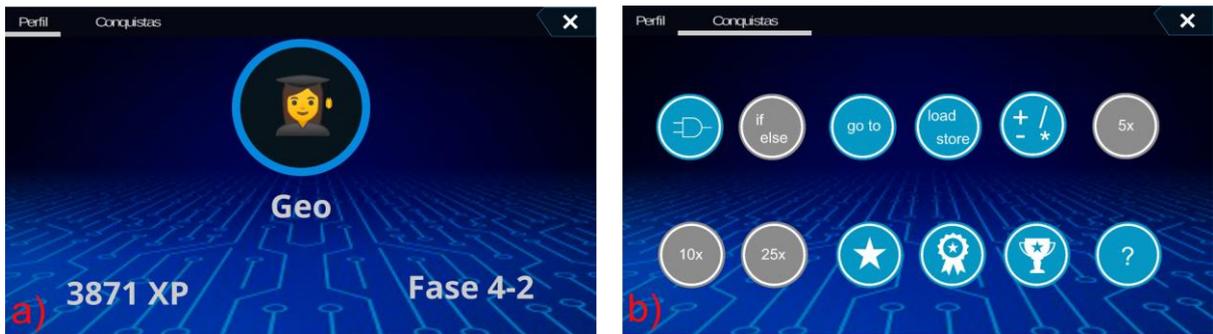
Fonte: Autoria própria

Além das fases, o ARtEMIS conta com os elementos de jogos: pontos de experiência (XP) e conquistas. O usuário conquistará os pontos sempre que: executar uma instrução (1 ponto), passar de subfase (100 pontos), passar de fase (1000 pontos) e resolver os desafios (2 pontos para cada questão correta mais 10 pontos se todas as questões estiverem corretas). As conquistas são liberadas pelo usuário ao longo do uso do ARtEMIS. No total são 12 conquistas apresentadas abaixo:

- Lógicas: liberada quando todas as instruções lógicas disponíveis (*sll, srl, and, or, nor, andi, ori*) são executadas.
- Desvio condicional: liberada quando todas as instruções de desvio condicional disponíveis (*beq, bne, slt, slti, sltu, sltiu*) são executadas.
- Desvio Incondicional: liberada quando todas as instruções de desvio incondicional disponíveis (*j, jal, jr*) são executadas.
- Transferência de dados: liberada quando todas as instruções de transferências de dados disponíveis (*lw, sw, lb, sb, lh, sh, lui*) são executadas.
- Aritméticas: liberada quando todas as instruções aritméticas disponíveis (*add, sub, addi, addu, addiu, subu*) são executadas.
- 5X: liberada quando cada tipo de instrução (*add, sub, etc*) é executado 5 vezes independente da fase/subfase.
- 10X: liberada quando cada tipo de instrução é executado 10 vezes independente da fase/subfase.
- 25X: liberada quando cada tipo de instrução é executado 25 vezes independente da fase/subfase.
- 50%: liberada quando o desafio da última subfase da Fase 2 é cumprido.
- 80%: liberada quando o desafio da última subfase da Fase 3 é cumprido.
- Campeão: liberada quando o desafio da última subfase da Fase 4 é cumprido.
- Média: liberada quando o desafio da última subfase da fase 4 é cumprido. Permite ao usuário fornecer a média obtida na disciplina de Arquitetura de Computadores e exibe uma mensagem e um emblema de acordo com a média informada. Se a média for maior ou igual a sete a mensagem parabeniza o aluno e o emblema é de um capelo simbolizando a aprovação, caso contrário, a mensagem aconselha o aluno a usar o ARtEMIS para obter mais conhecimento na disciplina e o emblema é de um *halter* simbolizando reforço.

Na Figura 18 (a), mostra-se a tela de perfil na qual o usuário pode consultar a fase na qual se encontra e sua pontuação. Na Figura 18 (b), mostra-se a tela de conquistas com as conquistas liberadas em azul e as bloqueadas em cinza.

Figura 18 – Telas com informações sobre o progresso do usuário no jogo. a) Informações sobre a pontuação adquirida e fase alcançada pelo usuário. b) Conquistas disponíveis para o usuário. As destacadas em azul estão liberadas e as em cinza estão pendentes



Fonte: Autoria própria

4.4 Interface do ARtEMIS

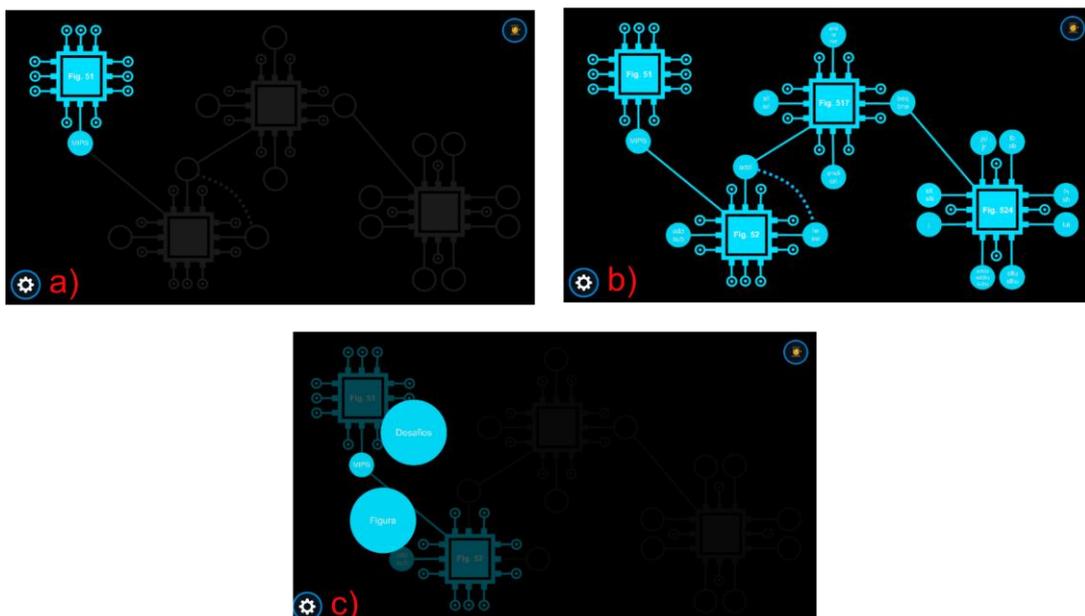
Da primeira vez que o simulador é executado é exibida a tela de cadastro (Figura 19) na qual o usuário deve informar o nome, o e-mail e o sexo. Depois dessa tela é exibido um tutorial que descreve o simulador e o funcionamento dos botões da tela inicial (Figura 20). Há tutoriais também na primeira subfase de cada fase e nos desafios. O tutorial da tela inicial e das fases pode ser ativado/desativado pelo usuário nas configurações do simulador. Os tutoriais dos desafios aparecem para o usuário enquanto não forem visualizados por completo. Além dos tutoriais, o usuário pode ativar/desativar a descrição de cada fase/subfase. Essa descrição contém informações sobre a ilustração que deve ser reconhecida, subfases que compõem a fase, instruções abordadas e missão. Um exemplo de descrição de fase/subfase pode ser visto na Figura 22.

Figura 19 – Tela de cadastro do usuário

Fonte: Autoria própria

A tela inicial do ARtEMIS é exibida na Figura 20 (a). Nessa tela há um botão (canto inferior esquerdo) para alterar as configurações do simulador (ativar/desativar o tutorial inicial e os tutoriais das fases e as descrições das fases), um botão (canto superior direito) para o usuário acessar informações sobre seu progresso no jogo (conquistas e pontos) e os botões que permitem acessar as subfases do jogo. Inicialmente, apenas a subfase 1-1 está liberada e aparece destacada em azul. A medida que o usuário interagir com o simulador e liberar as próximas fases/subfases essas vão sendo destacadas em azul (Figura 20 (b)). Todas as fases/subfases liberadas e desafios cumpridos ficam disponíveis para o usuário acessar a qualquer momento.

Figura 20 – Tela inicial do ARtEMIS. a) Apenas a subfase 1-1 está disponível. b) Todas as subfases estão liberadas. c) Opções de ir para o desafio da subfase ou reconhecer a ilustração do livro

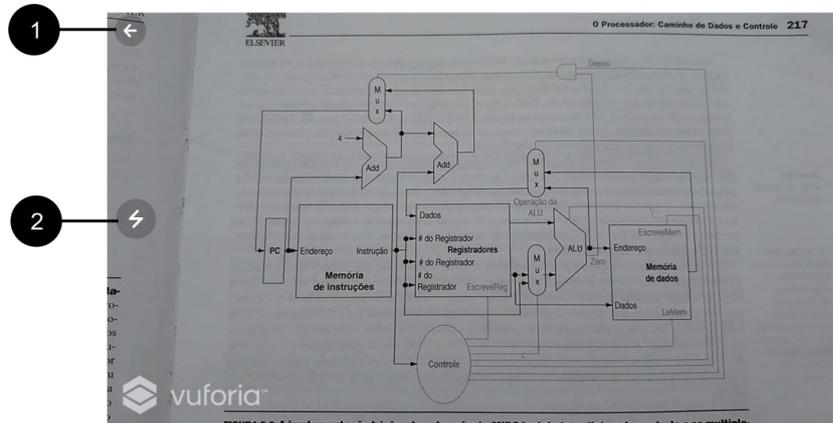


Fonte: Autoria própria

Em qualquer uma das subfases, até o desafio ser cumprido, quando o usuário seleciona a subfase é direcionado para a tela de reconhecimento da ilustração. Após o desafio ser cumprido, são dadas as opções (Figura 20 (c)) de ir para o desafio ou para a tela de reconhecimento.

A tela de reconhecimento de ilustração inicialmente exibida é mostrada na Figura 21 - se a opção de “descrição da subfase” estiver ativa é exibida a descrição da subfase (Figura 22), e só após o usuário clicar no botão “OK” da tela de descrição é que exibe-se a tela de reconhecimento. Nessa tela há um botão para voltar para a tela inicial e um botão para ativar o *flash* do dispositivo e evitar problemas de reconhecimento devido a iluminação.

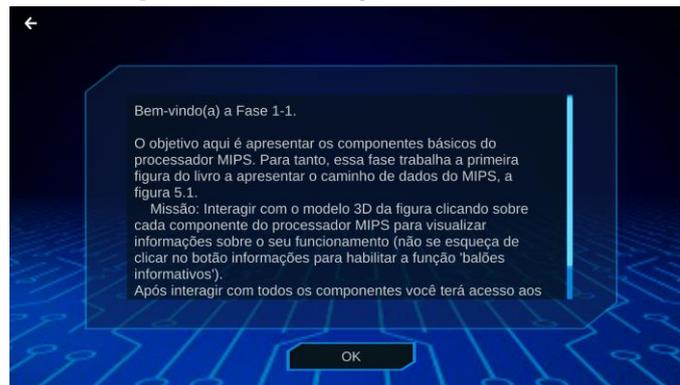
Figura 21 – Tela de reconhecimento antes da ilustração ser reconhecida



- 1 Voltar para a tela inicial. 2 Ativar/Desativar o *flash*.

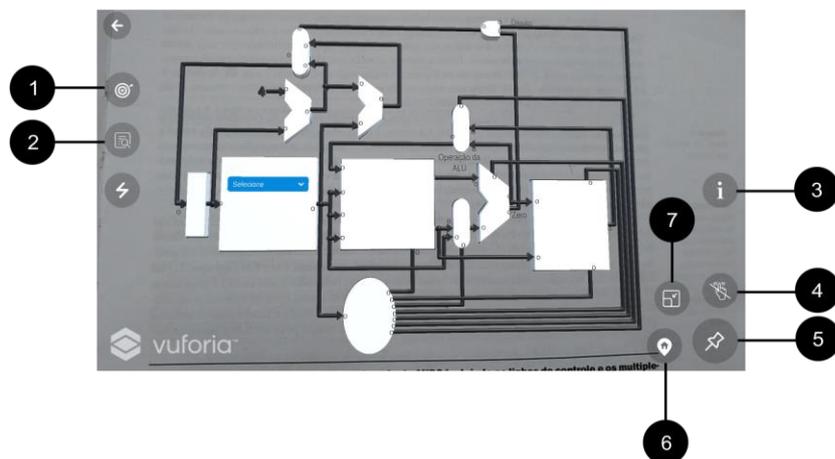
Fonte: Autoria própria

Quando a ilustração é reconhecida o modelo da arquitetura é gerado sobre ela juntamente com os elementos GUI que permitem configurar e interagir com ele, bem como a visualização das informações relacionadas a ilustração. Caso a ilustração não seja reconhecida, devido problemas na qualidade da impressão do livro por exemplo, o simulador permite o carregamento automático do modelo interativo após 10 segundos tentando fazer o reconhecimento.

Figura 22 – Descrição da subfase 1-1

Fonte: Autoria própria

Com a ilustração reconhecida (ou carregada automaticamente) novos botões são adicionados na tela de reconhecimento, como mostra a Figura 23. Esses botões são para: acessar a missão; acessar informações sobre as instruções (ausente na Fase 1, pois não são trabalhadas instruções); fixar o modelo interativo da ilustração reconhecida para facilitar a realização de atividades; travar/destravar a movimentação do modelo interativo da ilustração para que o usuário possa interagir com a tela sem movimentá-lo; restaurar o tamanho inicial do modelo interativo; retornar o modelo interativo para a posição inicial e ativar/desativar a opção de “Balões informativos” (explicados mais adiante).

Figura 23 – Tela de reconhecimento após a ilustração ser reconhecida. Destaque para os botões de funcionalidades

- | | | | |
|---|---|---|--|
| 1 | Visualizar a missão da fase. | 5 | Travar/Destravar o modelo 3D. |
| 2 | Visualizar informações sobre instruções. | 6 | Retornar o modelo 3D para a posição inicial. |
| 3 | Ativar/Desativar os balões informativos. | 7 | Restaura tamanho do modelo 3D. |
| 4 | Ativar/Desativar a movimentação do modelo 3D. | | |

Fonte: Autoria própria

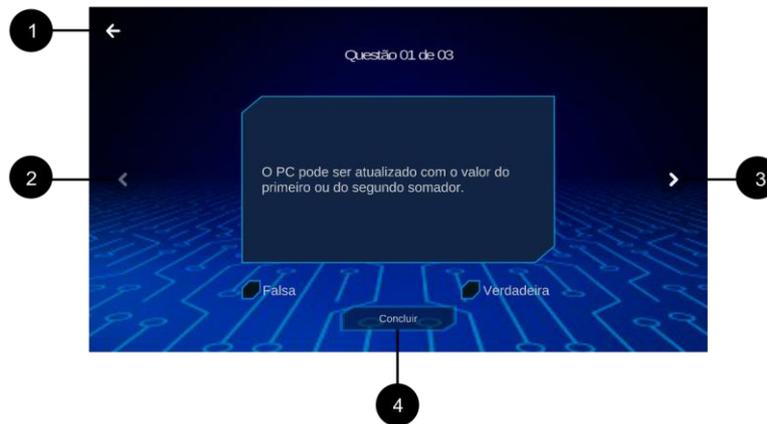
Os desafios das subfases são compostos por 1 (uma) até 3(três) questões que podem ser dos tipos: verdadeiro ou falso (Figura 24(a)), múltipla escolha (Figura 24 (b) e Figura 25(a)) ou para configurar os sinais do caminho de controle (Figura 25(b)).

Nas telas dos desafios de verdadeiro ou falso e de múltipla escolha há botões para voltar para a tela inicial, ir para a próxima questão, ir para a questão anterior e concluir o desafio. Quando o desafio é concluído, é exibida a tela de resultado (Figura 26) na qual o usuário visualiza o quantitativo de acertos, bem como se conseguiu liberar a próxima subfase/fase. Nessa tela o usuário também tem acesso às opções de visualizar seu gabarito - as questões que acertou ficam destacadas em verde e as que errou destacadas em vermelho; refazer o desafio, voltar para a subfase para praticar mais, ir para a próxima subfase/fase e voltar para a tela inicial são as demais opções apresentadas.

Na tela do último tipo de desafio (Figura 25 (b)) há um botão para voltar para a tela inicial e o botão “Verificar” que exibe uma janela com os valores de sinais que o usuário forneceu para a questão e os valores esperados para esses sinais. Nessa janela há o botão “OK” que permite ir para a próxima questão (caso haja) ou para a tela de resultado. A tela de resultado é semelhante à dos tipos de desafios anteriores, exceto que não possui o botão visualizar o gabarito.

Figura 24 – Tipos de desafios disponíveis. a) Questões de verdadeiro ou falso. b) Questões de múltipla escolha com textos

a)



1 Voltar para a tela inicial.

3 Ir para próxima questão.

2 Ir para questão anterior.

4 Concluir exercício.

b)



1 Voltar para a tela inicial.

3 Ir para próxima questão.

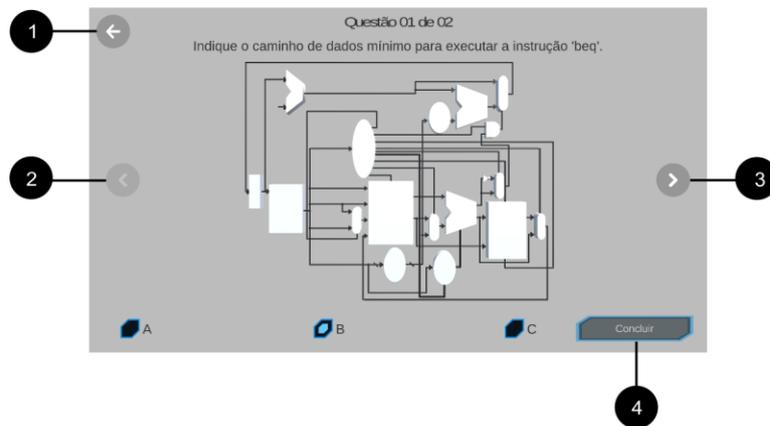
2 Ir para questão anterior.

4 Concluir exercício.

Fonte: Autoria própria

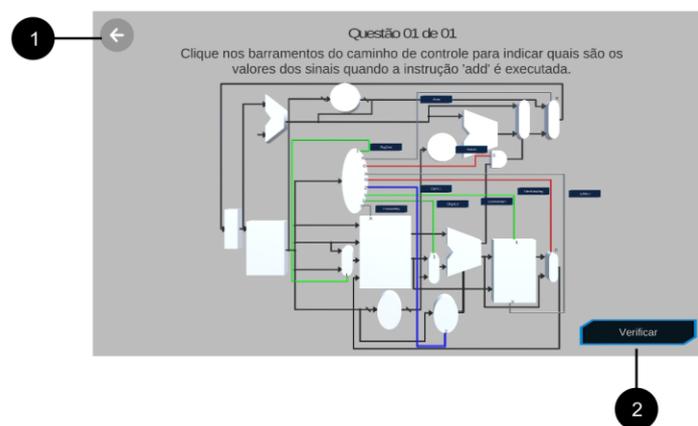
Figura 25 – Tipos de desafios disponíveis. a) Questões de múltipla escolha com caminhos de dados. b) Questões para configurar os sinais do controle

a)



- 1 Voltar para a tela inicial.
- 2 Ir para questão anterior.
- 3 Ir para próxima questão.
- 4 Concluir exercício.

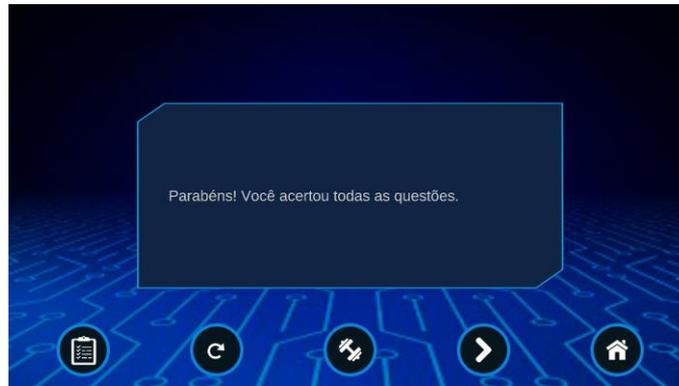
b)



- 1 Voltar para a tela inicial.
- 2 Verificar resposta.

Fonte: Autoria própria

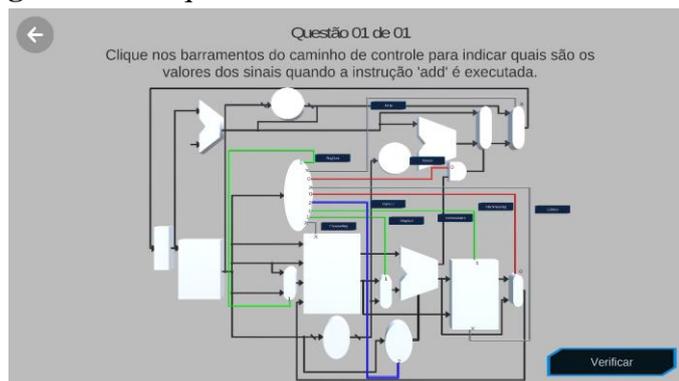
Figura 26 – Tela de resultado. Exemplo de mensagem para quando o usuário acerta todas as questões do desafio



Fonte: Autoria própria

Dos tipos de desafios, o mais diferenciado é o para configurar os sinais de controle. Nesse desafio o usuário deve clicar nos barramentos ligados ao controle para alterar os valores dos sinais que passam por eles, e assim, ajustar os valores necessários à execução de um determinado tipo de instrução. Inicialmente, todos os barramentos estão destacados em cinza e todos os sinais estão em *don't care* (X). Clicando 1 vez sobre o barramento o usuário altera o valor do sinal para zero (vermelho), clicado 2 vezes para 1 (verde) e clicando 3 vezes para *don't care* (cinza) novamente. No caso do barramento que sai do controle para o controle da ALU, clicando 3 vezes o valor é alterado para 2 (azul) e clicando 4 vezes é alterado para *don't care* (cinza) novamente. Esse esquema de cores e valores pode ser visto na Figura 27.

Figura 27 – Esquema de cores e valores dos barramentos

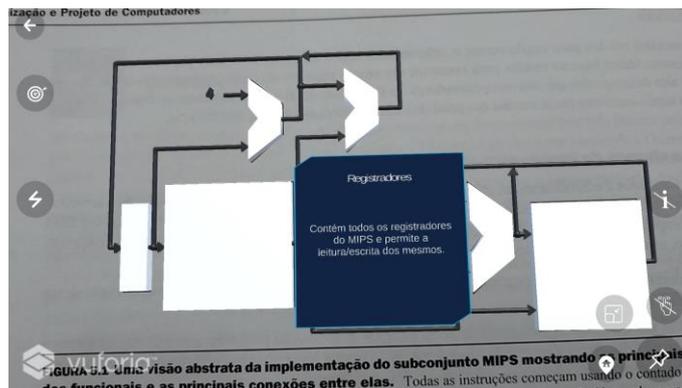


Fonte: Autoria própria

Na Fase 1 é reconhecida a ilustração 5.1, cujo propósito é apresentar as unidades funcionais básicas (PC, memória de instruções, banco de registradores, somadores, ULA e memória de dados) do processador MIPS. Ela é composta pela subfase 1-1. Nessa subfase o usuário tem como missão obter informações sobre as unidades funcionais básicas do

processador MIPS clicando sobre o modelo 3D gerado de cada uma delas quando a ilustração 5.1 é reconhecida. Ao ser clicada, a unidade é destacada e informações textuais sobre sua função são exibidas em um balão informativo, como mostrado na Figura 28. Para que o balão informativo desapareça basta clicar novamente na unidade ou clicar em uma outra unidade. Após interagir com todas as unidades funcionais o usuário libera o acesso ao desafio.

Figura 28 – Reconhecimento da ilustração 5.1 pelo ARtEMIS. Detalhe para o balão informativo com informações sobre a unidade banco de registradores



Fonte: Autoria própria

Na Fase 2 é reconhecida a ilustração 5.2. Essa ilustração estende o caminho de dados da ilustração 5.1 adicionando a unidade de controle. Esse caminho um pouco mais completo facilita o entendimento de como as instruções são executadas. Dessa forma, nessa fase começa-se a trabalhar o conjunto de instruções do MIPS apresentando algumas das instruções (*add*, *addi*, *lw*, *sub* e *sw*) suportadas pelo caminho representado na ilustração, seguindo a ordem em que elas são trabalhadas no livro. Inicialmente apenas *add* e *sub* estão disponíveis. Essas instruções são trabalhadas na subfase 2-1. As demais instruções são trabalhadas na subfase 2-2.

O reconhecimento da ilustração permite que o usuário interaja com as unidades funcionais do MIPS para configurar instruções para serem executadas. O usuário pode configurar uma instrução por vez e, em seguida, visualizar sua execução. A execução é dividida em etapas que representam os atrasos dos circuitos combinacionais e tem intuito didático, para facilitar o entendimento do funcionamento do caminho de dados.

Desse modo, quando a ilustração é reconhecida, o usuário deve selecionar na memória de instruções um tipo de instrução para visualizar a execução (*add*, *addi*, *lw*, *sub*, *sw*). Quando o tipo é selecionado é exibida uma tela sobre as unidades funcionais (Figura 29 (a)) que permite selecionar os registradores envolvidos na execução dela, definir seus respectivos valores e os valores da memória de dados e outros parâmetros (*offset* e *immediate*) quando

necessários. Nessa tela há também botões para acessar informações sobre os campos que compõem o tipo de instrução selecionado - Nas fases 3 e 4 esses botões estão presentes na aba “Código” mostrada na Figura 32. A inserção dos valores é opcional, e no caso do usuário não informar, eles serão setados com os valores padrões: zero. Para salvar valores na memória de dados ou registradores o usuário deve selecionar a posição da memória ou registrador e informar o valor. Esse valor é verificado e, caso seja válido, será salvo. Após informar os valores o usuário deve clicar no botão “Gerar Instrução” (Figura 29 (b)). Com isso, é verificado se todos os registradores necessários foram selecionados e, em caso afirmativo, a tela para definir valores e o botão “Gerar Instrução” são ocultados e os botões “Play”, “Anterior” e “Próximo” tornam-se visíveis (Figura 29(c-f)). Também passa a ser exibida a instrução gerada.

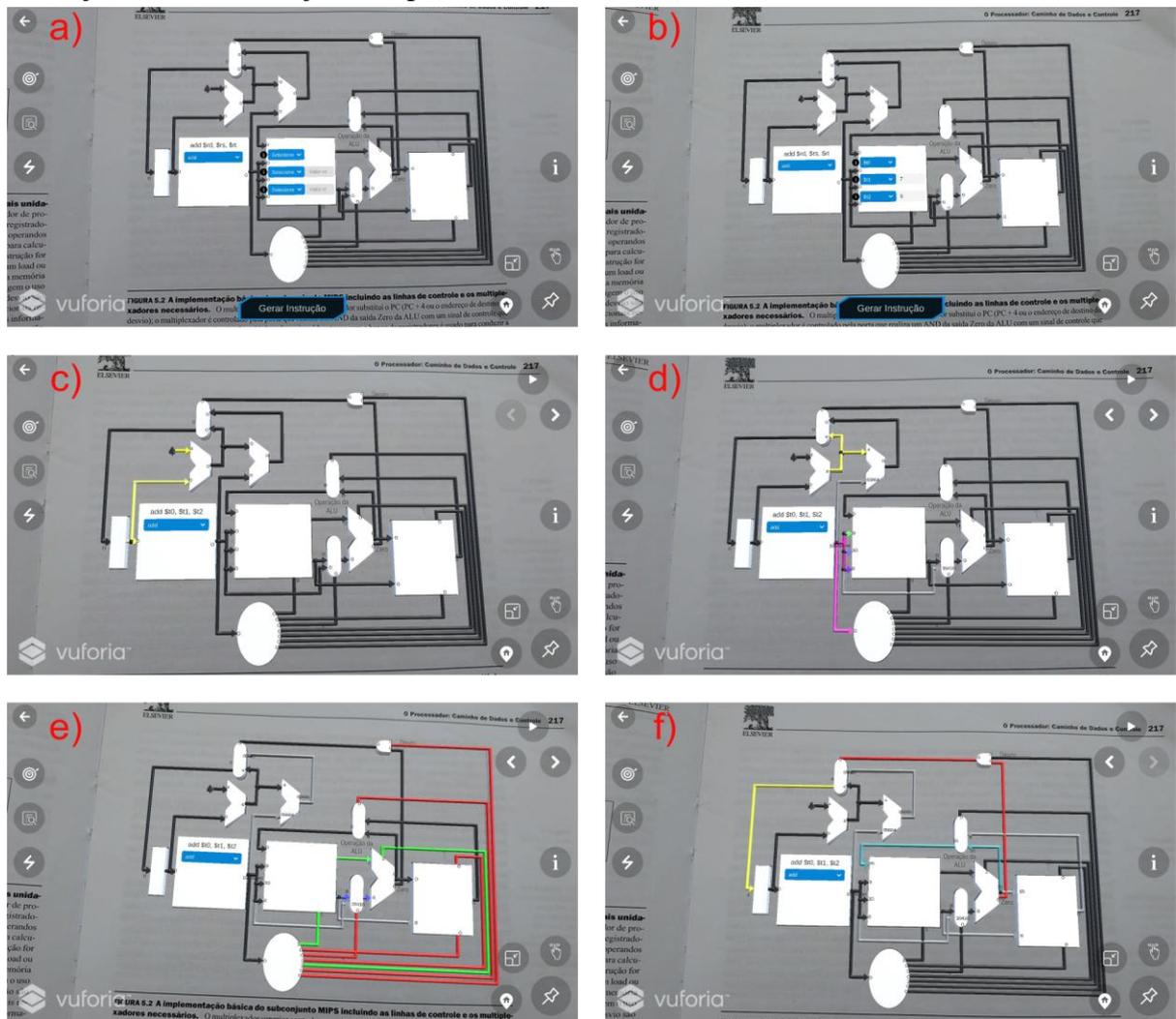
Com a instrução gerada, é possível visualizar as etapas da sua execução no modo automático ou no passo a passo. No modo automático, ativado pelo botão “Play”, uma etapa da execução da instrução é exibida a cada 2 segundos até que a execução termine. Quando o botão “Play” é clicado também é exibido um controle que possibilita alterar a velocidade de execução, selecionando um valor de 1 (mais lento) a 5 (mais rápido). No modo passo a passo os botões “Anterior” e “Próximo” permitem navegar entre as etapas da execução. Vale salientar que os passos ou etapas da execução não representam ciclos, uma vez que essa organização é monociclo.

Em cada etapa da execução é possível visualizar a atualização dos valores que saem das unidades ativas na etapa. Além disso, os barramentos ativos nos caminhos de dados e de controle são destacados de forma colorida. Os barramentos do caminho de dados que não são necessários para a instrução em execução são coloridos com cinza. Os demais são coloridos com cores diversificadas, com os barramentos que tem os mesmos dados (por exemplo, os que saem do PC para o primeiro somador e para a memória de instruções) são coloridos com a mesma cor e os que apresentam dados diferentes são coloridos com cores diferentes. No controle, os barramentos ativos (valor lógico 1) são coloridos em verde e os inativos (valor lógico 0) em vermelho. Como forma de destacar os barramentos ativos em cada etapa, todos os barramentos que foram coloridos na etapa anterior, exceto os coloridos em cinza, voltam a ser preto na etapa atual.

Na Figura 29, exemplifica-se o funcionamento do reconhecimento da ilustração 5.2 mostrando a simulação da execução de uma instrução do tipo *add*. Na Figura 29 (a) mostra-se a tela que surge sobre os componentes quando o tipo *add* é selecionado. Na Figura 29 (b), mostra-se a configuração da instrução na qual o registrador *t1* foi selecionado como o *rs* e

recebeu o valor 7, $t2$ foi selecionado como rt e recebeu o valor 8 e $t0$ foi selecionado como rd . Na Figura 29 (c) é mostrada a primeira etapa da execução da instrução na qual o PC repassa o endereço da instrução a ser executada para a Memória de Instruções e para o Somador. Na Figura 29 (d) o Somador soma 4 ao endereço vindo do PC e a instrução é decodificada. Na Figura 29 (e) os valores contidos em $t1$ e $t2$ são somados pela ULA. E por fim, na Figura 29 (f) o resultado da ULA é escrito em $t0$ e o PC é incrementado em 4.

Figura 29 – Funcionamento do reconhecimento da ilustração 5.2 mostrando a simulação da execução de uma instrução do tipo *add*



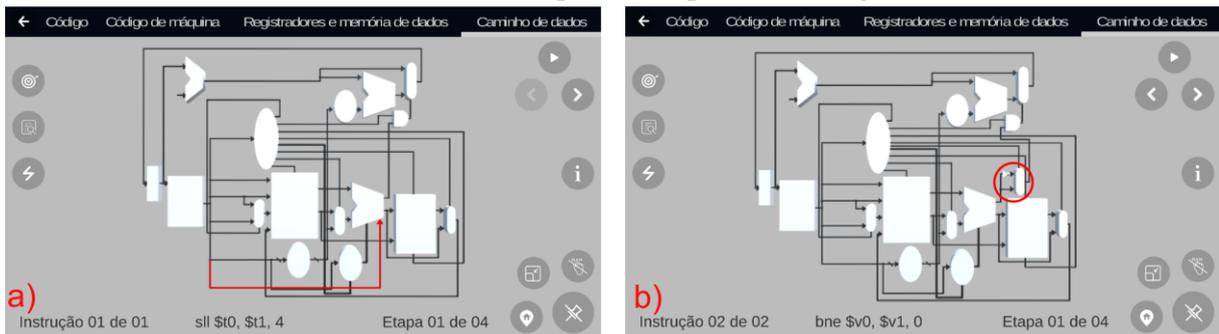
Fonte: Autoria própria

Nas Fases 3 (ilustração 5.17) e 4 (ilustração 5.24) o propósito é o mesmo da Fase 2, ou seja, permitir ao usuário visualizar a execução de instruções, contudo acrescentando instruções não suportadas nas fases anteriores. A Fase 3 é composta por 4 subfases (Figura 16). Na subfase 3-1 são trabalhadas as instruções *sll* e *srl*, na 3-2 as instruções *and*, *or* e *nor*, na 3-3 as instruções *andi* e *ori* e na 3-4 as instruções *beq* e *bne*. A Fase 4 é composta por 8

subfases (Figura 16). Na subfase 4-1 é trabalhada a instrução *j*, na 4-2 as instruções *slt* e *slti*, na 4-3 as instruções *jal* e *jr*, na 4-4 as instruções *lb* e *sb*, na 4-5 as instruções *lh* e *sh*, na 4-6 a instrução *lui*, na 4-7 as instruções *sltu* e *sltiu* e na 4-8 as instruções *addu* e *subu*.

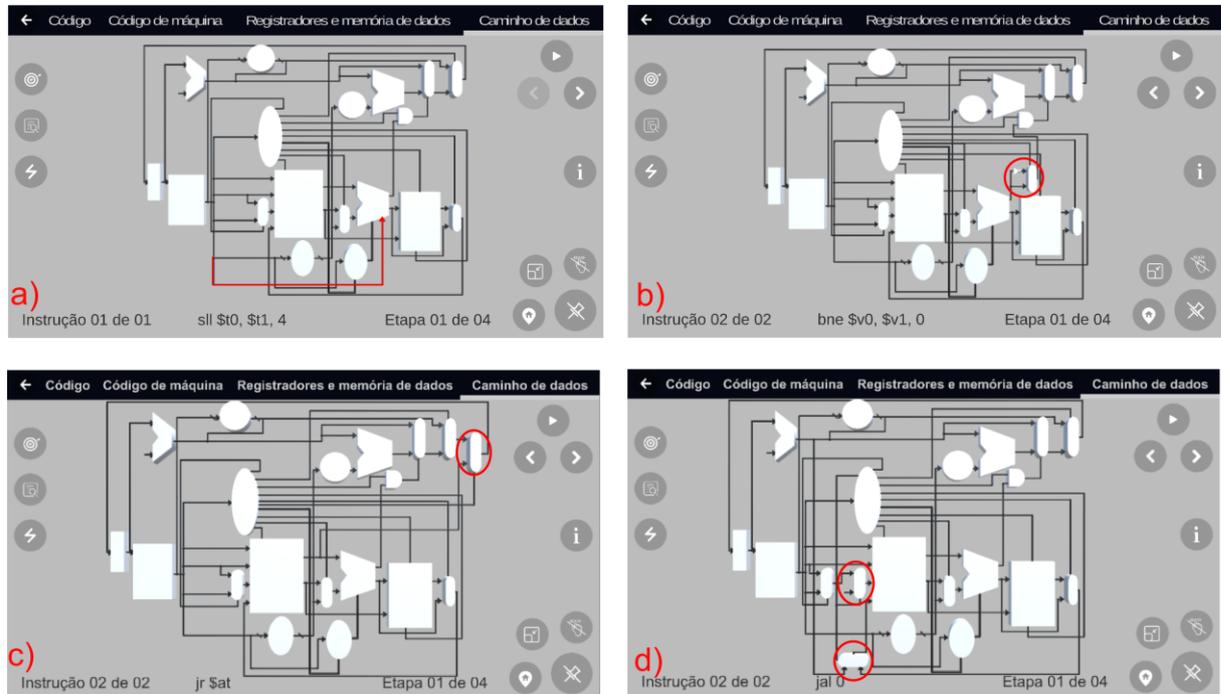
Embora as instruções *sll*, *srl*, *bne*, *jr* e *jal* sejam apresentadas no livro, em nenhuma das ilustrações dele o caminho de dados representado dá suporte a execução delas. Como a proposta do simulador é ser uma ferramenta integrada ao livro para facilitar o aprendizado dos conteúdos nele abordados, foram inseridos componentes nos caminhos originais das ilustrações 5.17 e 5.24 para permitir a execução dessas instruções tornando o ARtEMIS um recurso adicional ao livro. Na Figura 30 são mostrados os caminhos de dados com as modificações necessárias para dar suporte as instruções *sll*, *srl* e *bne* na Fase 3. Na Figura 31 mostra-se os caminhos de dados com as modificações necessárias para dar suporte as instruções *sll*, *srl*, *bne*, *jr* e *jal* na Fase 4.

Figura 30 – Caminho de dados da ilustração 5.17 modificado. Os componentes adicionados estão destacados em vermelho. a) Extensão do caminho de dados para dá suporte as instruções *sll* e *srl*. b) Extensão do caminho de dados para dá suporte a instrução *bne*.



Fonte: Autoria própria

Figura 31 – Caminho de dados da ilustração 5.24 modificado. Os componentes adicionados estão destacados em vermelho. a) Extensão do caminho de dados para dá suporte as instruções *sll* e *srl*. b) Extensão do caminho de dados para dá suporte a instrução *bne*. c) Extensão do caminho de dados para dá suporte a instrução *jr*. d) Extensão do caminho de dados para dá suporte a instrução *jal*



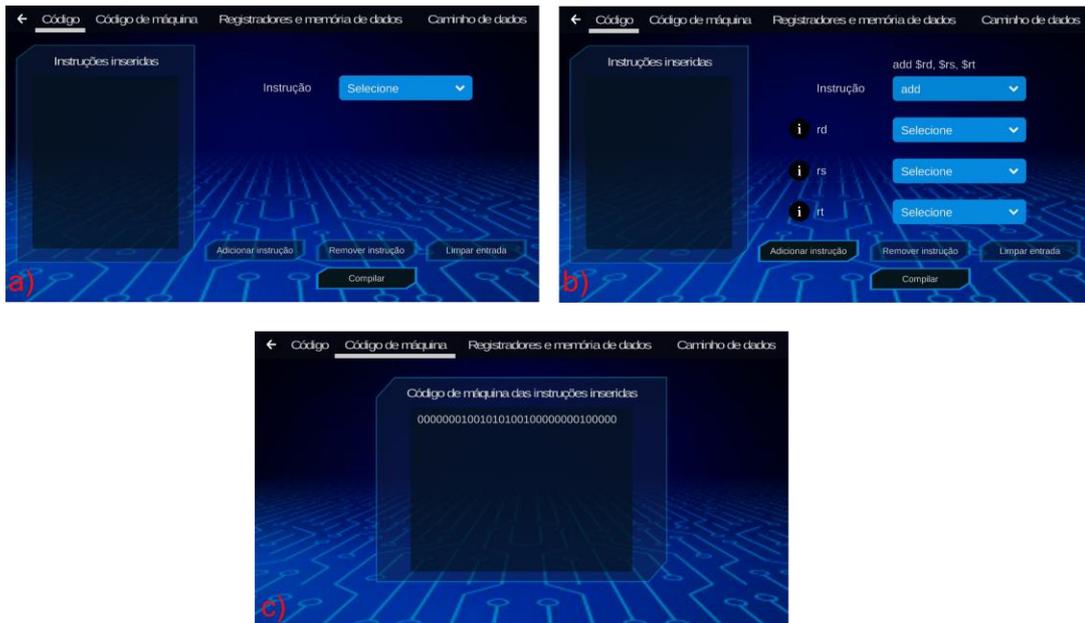
Fonte: Autoria própria

Diferentemente da Fase 2, no qual o usuário deve interagir com os componentes do processador para gerar as instruções, uma por vez, as Fase 3 e 4 permitem que o usuário forneça o código com várias instruções de uma vez para executar e visualizar a execução delas em sequência (a ideia de um programa mesmo).

A partir da Fase 3, a aba “Código” torna-se disponível, ver Figura 32 (a), na qual o usuário pode inserir o programa que deseja executar. O código deve ser inserido em linguagem de montagem *assembly*. Essa aba possui os botões “Adicionar instrução”, “Remover instrução”, “Limpar entrada” e “Compilar” e uma caixa de seleção para o usuário selecionar o tipo de instrução que deseja inserir. Ao selecionar o tipo de instrução que deseja inserir o usuário tem acesso a campos para informar os registradores e outros parâmetros (*offset* e *immediate*) envolvidos na execução da instrução (Figura 32 (b)). Após preencher os campos o usuário deve clicar no botão “Adicionar instrução”. O botão “Remover instrução” remove a última instrução inserida e o botão “Limpar entrada” remove todas as instruções. O botão “Compilar” submete o código para compilação. Quando o código é compilado, a tela de inserir o código é ocultada e os botões “Play”, “Anterior” e “Próximo” tornam-se visíveis na

RA de reconhecimento das ilustrações. Há também a aba “Código de máquina” (Figura 32 (c)) que permite ao usuário visualizar as instruções em linguagem de máquina binária gerada a partir das instruções do código *assembly* inserido.

Figura 32 – Abas Código e Código de máquina. a) Aba código inicialmente exibida ao usuário. b) Aba código após o usuário selecionar um tipo de instrução para inserir. c) Aba código de máquina exibindo o binário das instruções inseridas



Fonte: Autoria própria

A aba “Registadores e memória de dados” (Figura 33), permite ao usuário visualizar/alterar os valores dos registradores e da memória de dados. Ela possui dois botões “Reset” para restaurar os valores da memória de dados e dos registradores, separadamente. A última aba é “Caminho de dados” para visualização da execução das instruções no caminho de dados.

Figura 33 – Tela para visualizar/alterar os valores da memória de dados e dos registradores



Fonte: Autoria própria

Para acessar uma determinada aba o usuário deve clicar sobre o nome dela. Os botões “Play”, “Anterior” e “Próximo” também estão disponíveis para o controle da simulação. Os esquemas de cores e a atualização de valores também funcionam do mesmo modo que na Fase 2. Também é exibido o total de instruções inseridas, a instrução atual em execução (posição e *assembly*) e em qual etapa da execução ela se encontra.

4.5 Considerações finais

Esse capítulo apresentou o simulador desenvolvido. Foi dada uma visão geral do ARtEMIS na qual foi explicado o papel da RA e em seguida foi explicada como ele foi implementado. Em seguida foi explicada como a proposta de gamificação do simulador e, por fim, foi explicado o funcionamento do mesmo.

5 VALIDAÇÃO DO ARTEMIS

Neste capítulo é descrita a validação do simulador. A metodologia usada para validação, a apresentação dos dados e a discussão dos resultados é apresentado ao longo deste capítulo.

5.1 Metodologia

A validação do ARtEMIS ocorreu durante a realização do V WTCC (Workshop Técnico-Científico de Computação). O WTCC é um evento que tem como objetivo despertar e estimular o interesse de estudantes de graduação e pós-graduação em Computação em extensão e pesquisa científica em temas relacionados à ciência da computação (V WTCC, 2019). O V WTCC ocorreu no período de 23 à 26 de abril de 2019. A validação consistiu em 3 etapas realizadas nos 3 primeiros dias do evento. A seguir são descritas as etapas e apresentados os resultados obtidos em cada uma delas.

5.1.1 Etapa 01: Pré-teste

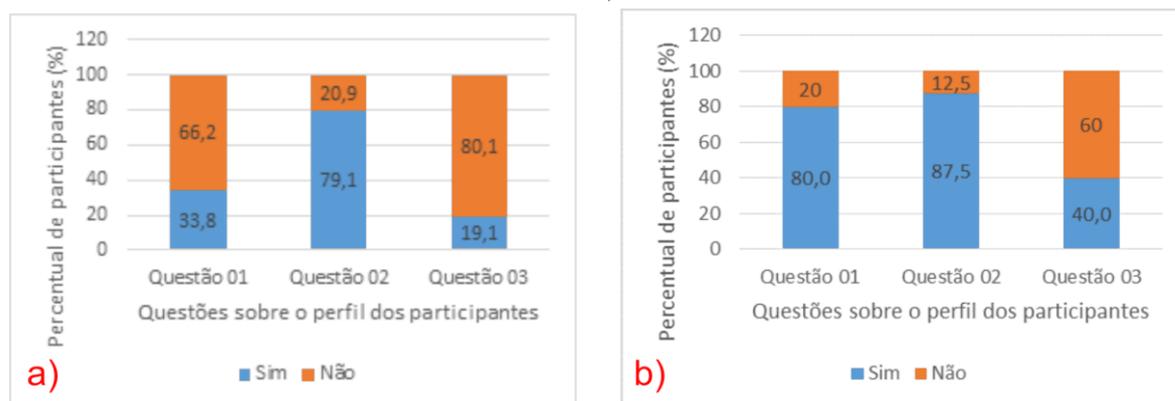
Na primeira etapa buscou-se conhecer o perfil dos participantes e mensurar quanto eles sabiam sobre o processador MIPS. Foi aplicado um questionário pré-teste (APÊNDICE B – QUESTIONÁRIO PRÉ-TESTE) contendo 3 questões sobre o participante e 10 questões sobre o processador MIPS.

Um total de 71 pessoas responderam o questionário. No entanto, apenas 10 delas concluíram a etapa 02 e somente dessas foi possível coletar dados para fazer a comparação do desempenho antes e depois de usar o ARtEMIS. Dessa forma, para a análise dos resultados do questionário pré-teste, as respostas serão apresentadas em dois grupos: um contendo as respostas de todos os participantes chamado de GC (Grupo Completo) e outro contendo as respostas apenas dos 10 participantes que concluíram a etapa 02, chamado de GP (Grupo Parcial).

5.1.1.1 Perfil do participante

A análise do perfil dos participantes foi feita por meio de 3 questões. Os resultados obtidos para elas são exibidos na Figura 34.

Figura 34 – Resultado obtidos para as questões relacionadas ao perfil dos participantes. a) Resultado do GC. b) Resultado do GP



Fonte: Autoria própria

A primeira questão sobre o perfil do participante perguntava se ele já havia cursado a disciplina Organização e Arquitetura de Computadores (OAC). Como se pode ver na Figura 34 (a), no GC 33,8% dos participantes (24 participantes) responderam que sim. Na Figura 34 (b), com as respostas apenas do GP, 80% deles (8 participantes) responderam que sim.

A segunda questão era voltada para os participantes que responderam sim à primeira questão. Nela se queria saber se a disciplina OAC foi cursada na UFERSA. No GC 79,1% (19 participantes) responderam que sim (Figura 34 (a)). No GP, 87,5% (7 participantes) cursou na UFERSA (Figura 34 (b)).

A terceira questão perguntava se o participante já havia estudado o processador MIPS, na disciplina OAC ou estudando por conta própria. No GC 19,1% (14 participantes) já haviam estudado (Figura 34 (a)) e desses, todos já cursaram a disciplina OAC sendo que todos cursaram-na na UFERSA. No GP, 40% dos participantes (4 participantes) já haviam estudado o processador MIPS (Figura 34 (b)), e como no GC, todos eles já cursaram a disciplina e todos cursaram-na na UFERSA.

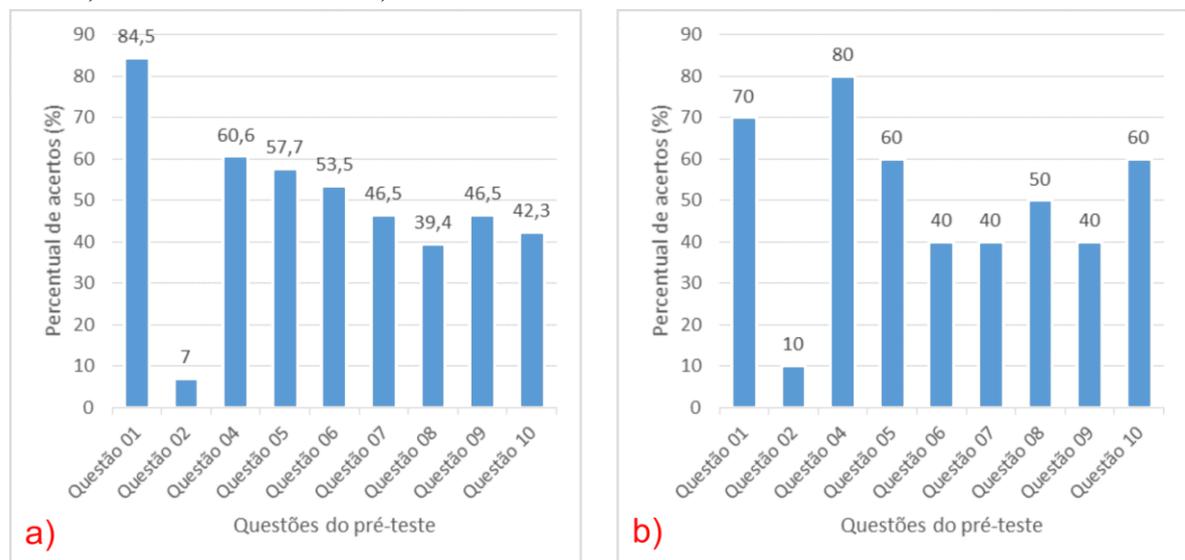
Os resultados da análise do perfil mostram que, de 33,8% participantes do GC (24 participantes) que cursaram a disciplina OAC, 14 deles foram alunos da UFERSA e estudaram o processador MIPS. Dos 10 restantes, metade cursaram OAC na UFERSA, mas não estudaram o processador MIPS e a outra metade não cursou OAC na UFERSA e não

estudou o processador MIPS. Considerando o GP, dos 80% dos participantes (8 participantes) que cursaram a disciplina OAC, metade deles foram alunos da UFERSA e estudaram o processador MIPS. Da outra metade que cursou OAC, 3 participantes foram alunos da UFERSA que não estudaram o processador MIPS e 1 participante não foi aluno da UFERSA e não estudou o processador MIPS.

5.1.1.2 Conhecimentos sobre o MIPS

O questionário pré-teste continha 10 questões sobre o processador MIPS, no entanto, foi identificada uma inconsistência na questão 3 e essa foi anulada sendo consideradas nos cálculos as outras nove questões. Na Figura 35 (a), mostra-se o percentual de participantes que acertaram cada questão para o GC e na Figura 35 (b), mostra-se o percentual para o GP.

Figura 35 – Percentuais de participantes que acertaram cada questão do questionário pré-teste. a) Percentuais do GC. b) Percentuais do GP

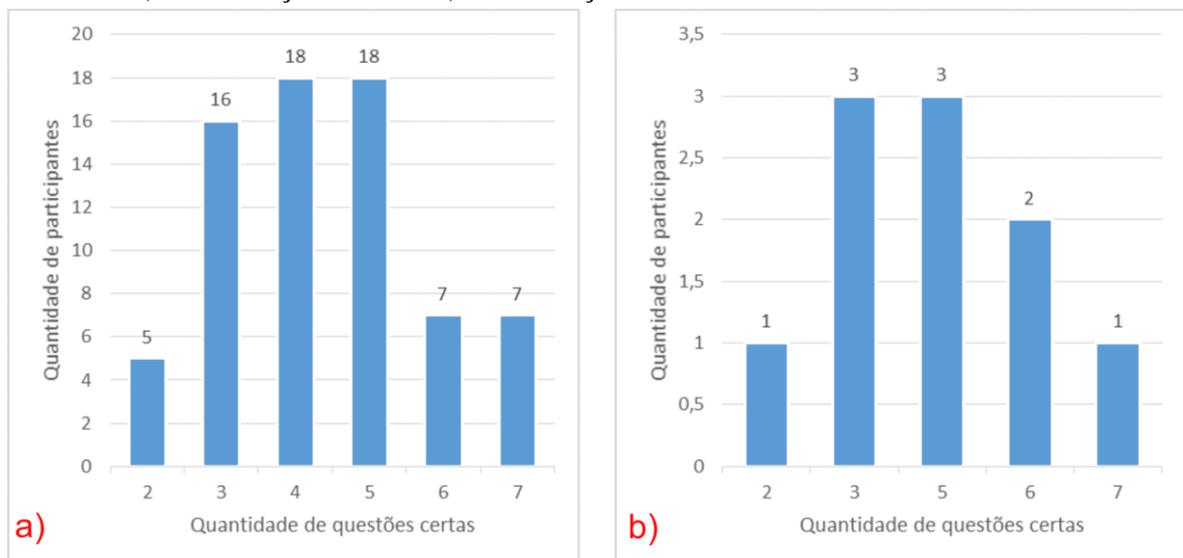


Fonte: Autoria própria

O nível de dificuldade das questões foi equivalente já que, de modo geral, os percentuais de acerto delas foram próximos (entre 40% e 60%). A questão com percentual mais diferenciado foi a Questão 02. O que pode justificar o baixo percentual de acerto é que nela os participantes precisavam digitar a resposta correta enquanto as demais eram questões de múltipla escolha ou de Verdadeiro ou Falso nas quais as possíveis respostas estavam disponíveis e eles só precisavam escolher uma dentre elas.

Na Figura 36 (a), mostra-se a distribuição dos participantes do GC em relação a quantidade de questões que acertaram. Na Figura 36 (b), mostra-se essa distribuição para os participantes do GP. No GC 55 % dos participantes obtiveram um percentual de acerto menor que 50% das questões. Já no GP, 60 % dos participantes acertaram 50% das questões ou mais. O que pode justificar esses percentuais é o fato de que no GC apenas 33,8% dos participantes haviam cursado OAC e desses, apenas 58,3 % estudaram o processador MIPS. Já no GP, 80% dos participantes já havia cursado OAC e desses, 50% estudaram o processador MIPS.

Figura 36 – Distribuição dos participantes em relação a quantidade de questões que acertaram. a) Distribuição do GC. b) Distribuição do GP



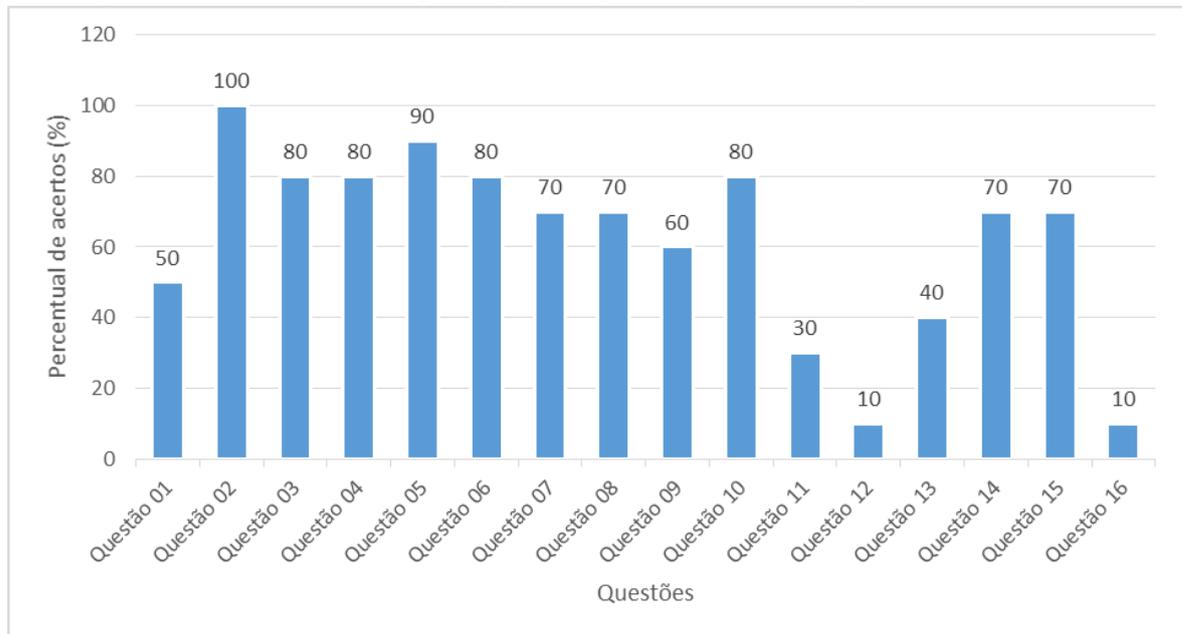
Fonte: Autoria própria

5.1.2 Etapa 02: uso do ARtEMIS e Pós-teste

Na segunda etapa os participantes usaram o ARtEMIS e ao mesmo tempo responderam o questionário pós-teste (APÊNDICE C – QUESTIONÁRIO PÓS-TESTE). Durante o uso do ARtEMIS os participantes responderam questões referentes ao MIPS sob forma de desafios presentes no próprio aplicativo. Essas questões constituem o pós-teste e tratam dos mesmos assuntos abordados no pré-teste aplicado na etapa 01. As respostas fornecidas para elas foram coletadas para comparar com o desempenho obtido no pré-teste. Ressalta-se que apenas 10 dos 71 participantes concluíram essa etapa, de modo que apenas as

respostas desses foram coletadas. Nessa etapa haviam 16 questões sob forma de desafios. Os percentuais de participantes que acertaram cada questão são mostrados na Figura 37.

Figura 37– Percentuais de participantes que acertaram cada questão do ARtEMIS



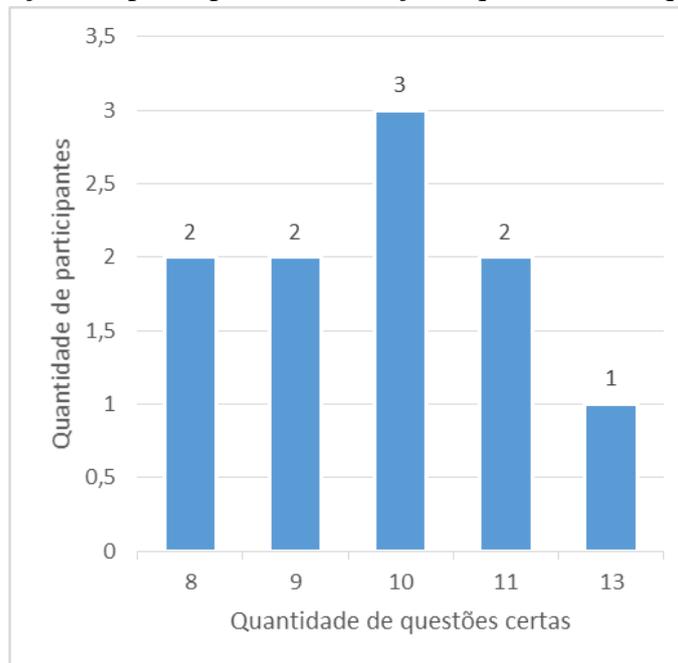
Fonte: Autoria própria

As questões 01, 11, 12, 13 e 16 foram as que tiveram menor percentual de acerto. A Questão 01 perguntava se o valor do PC pode ser atualizado com o valor dos dois somadores presentes no caminho de dados. Apesar de as próprias ilustrações do livro mostrarem que isso é verdade, como na ilustração do livro trabalhada na fase 1 os multiplexadores ainda estão presentes, talvez isso não tenha ficado claro para uma parte dos participantes (50%) que jogou a afirmação como falsa. A Questão 11 era relacionada às instruções *and*, *or* e *nor* e os participantes tinham que trabalhar com números binários, talvez por isso eles tenham tido maior dificuldade nela. As questões 12 e 13 eram para indicar o caminho de dados correto para executar as instruções *beq* e *bne*, respectivamente. Embora na missão da subfase desses desafios (questões) houvesse um alerta para observar as modificações no caminho de dados, os participantes podem não ter observado e por isso tiveram dificuldades para responder essas questões. A questão 16 era do tipo de desafio para configurar os sinais de controle. Como no total haviam 9 sinais para configurar e a questão só era considerada correta se todos os sinais estivessem corretos, talvez isso explique o fato de apenas um participante ter acertado essa questão.

As questões 02, 05 foram as que tiveram maior percentual de acerto. A Questão 02 teve 100% de acertos e perguntava se a função do primeiro Somador do caminho de dados é somar 4 ao valor do PC. Como nas próprias ilustrações do livro uma das entradas do primeiro Somador é um 4 e a outra é o valor do PC, isso facilitou julgar a afirmação como verdadeira ou falsa. A Questão 05 foi a segunda com maior percentual de acertos e perguntava do resultado obtido na execução da última instrução da missão da fase 2-1. Como a questão trata do resultado de uma missão, os participantes podem ter refeito a missão para verificar qual a resposta correta da questão. As demais questões tiveram percentuais de acerto próximos, variando de 60% a 80%.

A distribuição dos participantes em relação a quantidade de questões que acertaram é mostrada na Figura 38. Como se pode observar, todos os participantes acertaram pelo menos 50% das questões. Ao comparar essa distribuição com a distribuição da Figura 36 (b) percebe-se um aumento de 40 % no percentual de participantes que acertaram pelo menos 50% das questões.

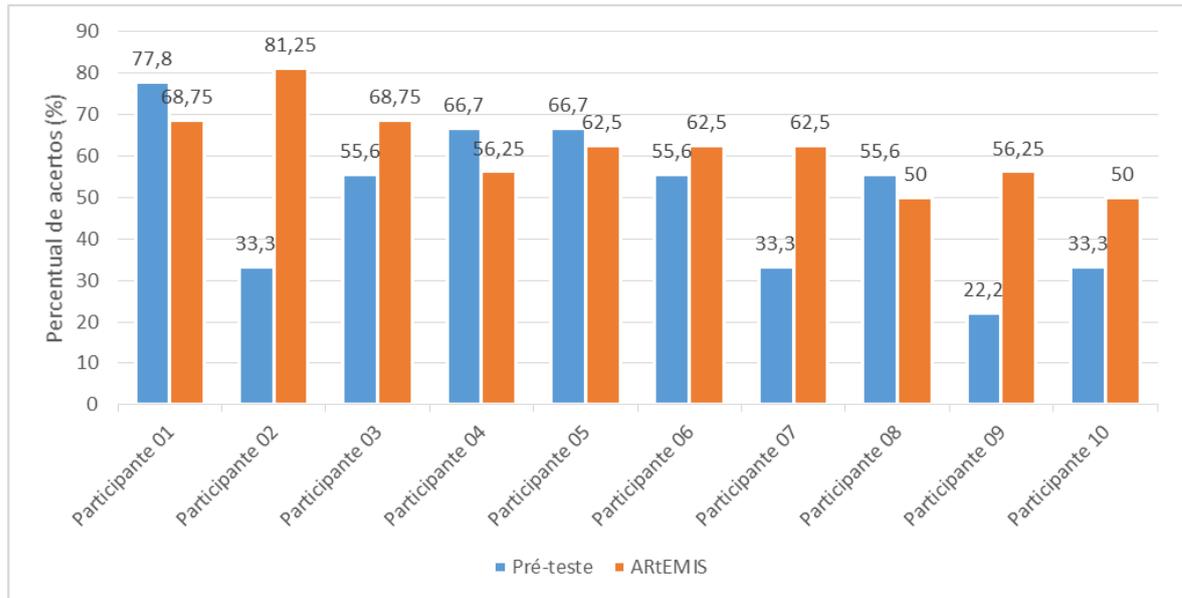
Figura 38 – Distribuição dos participantes em relação a quantidade de questões que acertaram



Fonte: Autoria própria

A comparação dos percentuais de acertos dos participantes obtidos no pré-teste e usando o ARtEMIS é feita na Figura 39. Como pode-se ver, 60% dos participantes melhoram o desempenho.

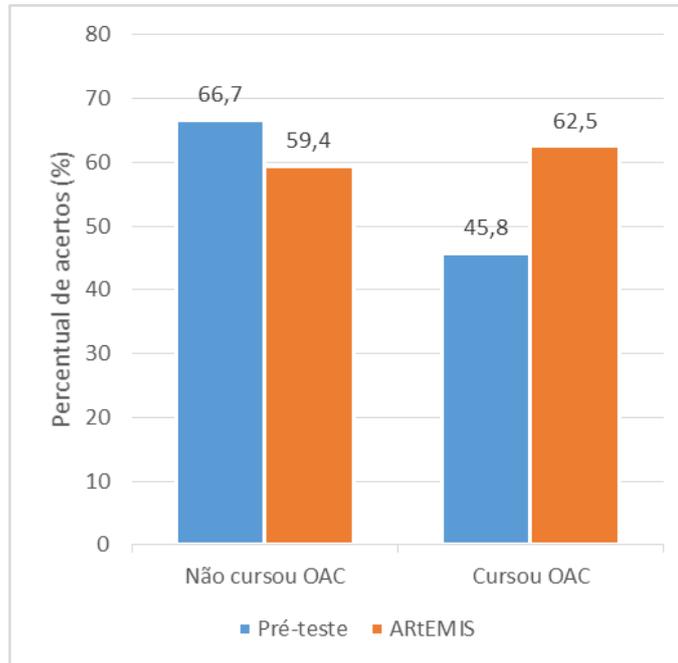
Figura 39 – Comparação dos percentuais de acertos dos participantes obtidos no pré-teste e usando o ARtEMIS



Fonte: Autoria própria

Buscando identificar a relação dos resultados com os diferentes perfis de participantes algumas análises foram feitas. A primeira delas foi em relação ao impacto do participante ter ou não cursado OAC. O resultado dessa análise é exibido na Figura 40. Como se pode observar, o desempenho médio dos participantes que não cursaram OAC (Participantes 4 e 5) caiu 7,3%. Já o dos participantes que cursaram aumentou em 16,7%. Para os participantes que já cursaram OAC, o ARtEMIS ajudou a revisar os conteúdos, levando a melhora do desempenho. Já para os participantes que não cursaram, a queda no desempenho surpreende uma vez que as questões do ARtEMIS foram consideradas mais fáceis do que as do pré-teste (Ver seção 5.1.3).

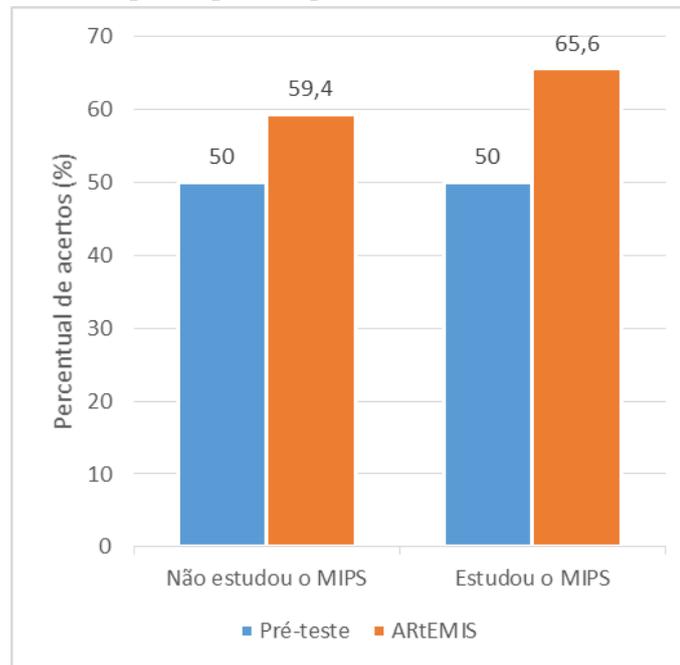
Figura 40 – Comparação do desempenho médio dos participantes que cursaram OAC com o dos participantes que não cursaram



Fonte: Autoria própria

O fato de o participante ter ou não estudado o processador MIPS é outro fator impactante. O resultado dessa análise pode ser visto na Figura 41. Como se pode observar, tanto os participantes que não estudaram o processador MIPS (Participantes 3, 4, 5, 7, 8, e 9) quanto os que estudaram melhoraram seu desempenho médio, em 9,4 % e 15,6 % respectivamente. Nesse caso, novamente a hipótese é que o ARtEMIS ajudou a revisar os conhecimentos sobre o MIPS, e por isso os participantes que estudaram o processador tiveram uma melhora no desempenho superior aos que não estudaram. Além disso, todos que estudaram o processador MIPS já cursaram OAC enquanto nem todos que não estudaram o processador MIPS cursaram a disciplina.

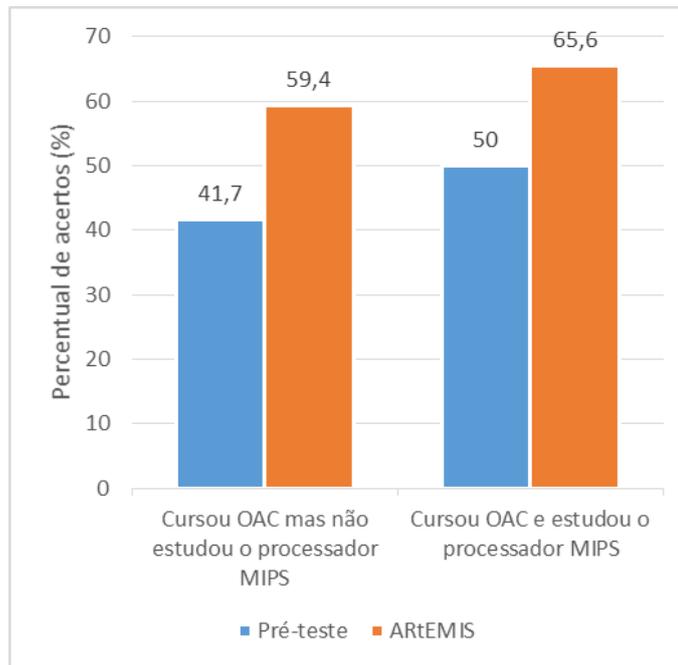
Figura 41 – Comparação do desempenho médio dos participantes que não estudaram o processador MIPS com o dos participantes que estudaram



Fonte: Autoria própria

Também foi feita uma análise considerando apenas os participantes que já cursaram OAC. Para tanto, foi feita uma comparação do desempenho médio dos participantes que cursaram OAC e não estudaram o processador MIPS com o dos que cursaram OAC e estudaram o processador MIPS obtido no pré-teste e usando o ARtEMIS. O resultado dessa análise pode ser visto na Figura 42. Observa-se que nos dois casos os participantes melhoraram seu desempenho médio. No primeiro grupo os participantes melhoraram o desempenho em 17,7%. No segundo grupo os participantes melhoraram o desempenho em 15,6%. Esse resultado surpreendeu, pois, o esperado era que os participantes do primeiro grupo tivessem um aumento no desempenho superior aos participantes do segundo, uma vez que com o ARtEMIS eles revisam os conhecimentos sobre o processador. Mas é um resultado positivo pois em ambos os casos houve melhora no desempenho.

Figura 42 – Comparação do desempenho médio dos participantes que cursaram OAC mas não estudaram o MIPS com o dos participantes que cursaram OAC e estudaram o MIPS



Fonte: Autoria própria

5.1.3 Etapa 03: avaliação do processo

Na terceira etapa os participantes avaliaram o ARtEMIS e as atividades propostas por meio de um questionário (APÊNDICE D – QUESTIONÁRIO DE AVALIAÇÃO DO ARTEMIS E DAS ATIVIDADES PROPOSTAS). Embora apenas 10 participantes tenham concluído a etapa 02, também se permitiu que os demais participantes que usaram o ARtEMIS o avaliassem. Com isso, nessa etapa foram coletadas respostas de 21 participantes.

O questionário é composto por 6 questões relacionadas a usabilidade do ARtEMIS, 5 relacionadas às estratégias empregadas no desenvolvimento do ARtEMIS e 4 relacionadas às atividades propostas.

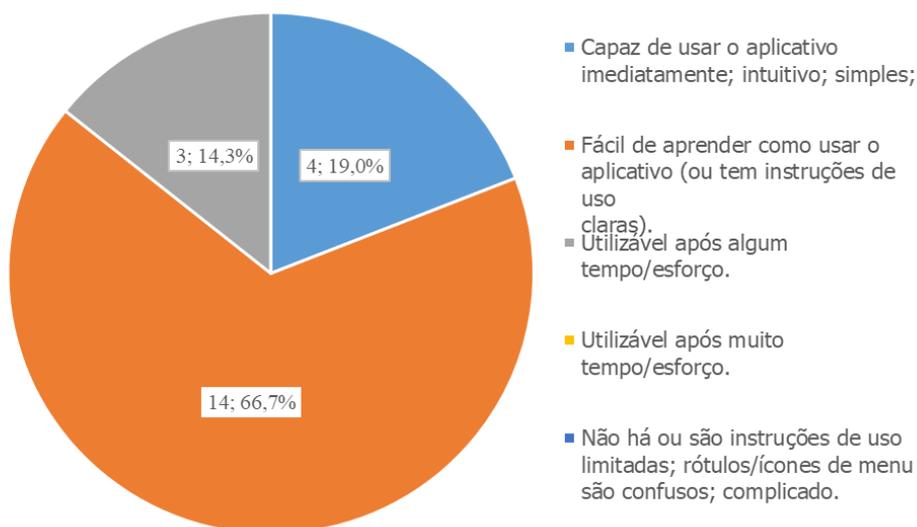
5.1.3.1 Avaliação da usabilidade do ARtEMIS

As questões relacionadas a usabilidade foram baseadas do questionário *Mobile App Rating Scale* (STOYANOV et al., 2015). Em todas essas questões haviam 5 opções e em

todas elas os usuários selecionaram uma das 3 primeiras opções, que são as que representam a avaliação mais positiva. Os resultados para cada uma delas são apresentados a seguir.

Na primeira questão foi avaliada a facilidade de uso, isto é, o quão fácil é aprender a usar o ARtEMIS e quão claras são as etiquetas/ícones e as instruções de uso. Como se pode observar na Figura 43, 85,7% (19% + 66,7%) dos participantes achou o ARtEMIS fácil de aprender a usar e intuitivo e simples.

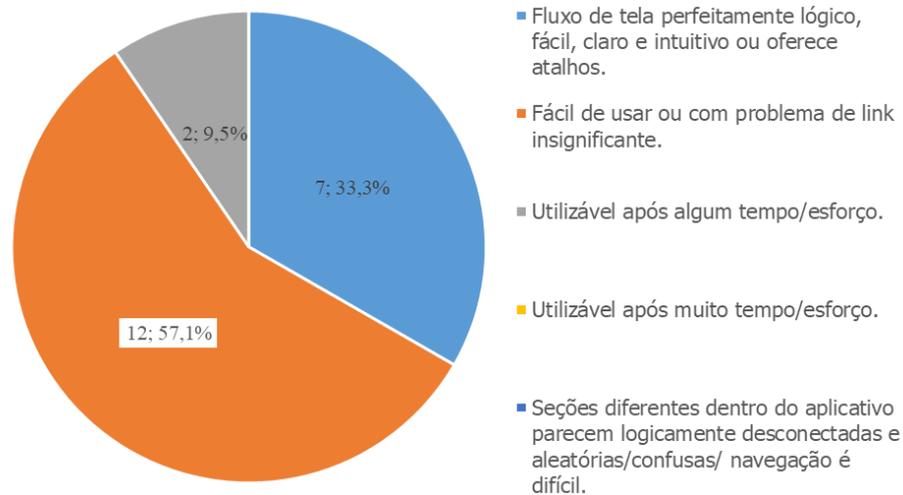
Figura 43 – Questão 01: Quão fácil é aprender a usar o aplicativo; quão clara são as etiquetas/ícones do menu e instruções de uso?



Fonte: Autoria própria

A segunda questão tratou da navegação no ARtEMIS, ou seja, a movimentação entre as telas e os links entre elas. Como exibido na Figura 44, cerca de 90,4% (33,3% + 57,1%) dos participantes acharam o fluxo de tela lógico, fácil, claro e intuitivo ou com problema de *link* insignificante.

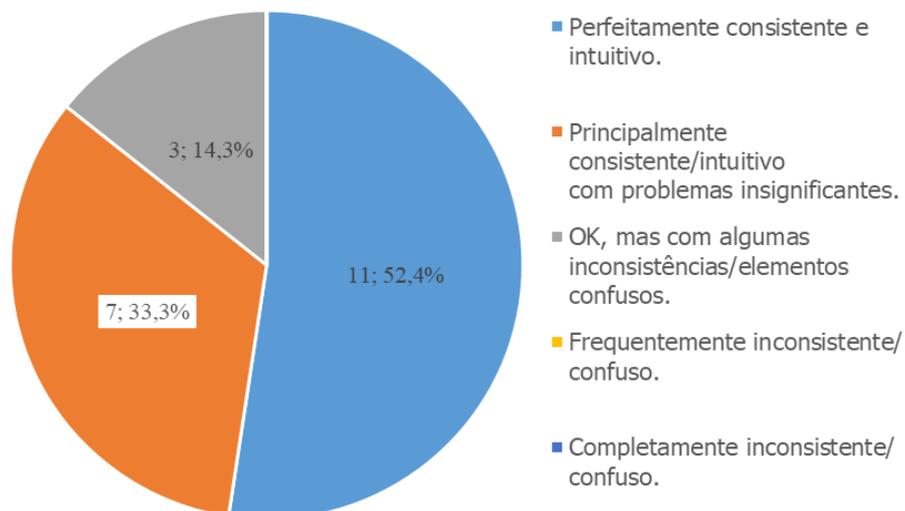
Figura 44 – Questão 02: A movimentação entre as telas é lógica/precisa/apropriada/ininterrupta; todos os links de tela necessários estão presentes?



Fonte: Autoria própria

A terceira questão avaliou o projeto gestual do ARtEMIS, isto é, toques, *swipes*, *pinches* e *scrolls*. Como mostra a Figura 45, cerca de 85,7% (52,4% + 33,3%) dos participantes achou consistente e intuitivo ou com problemas insignificantes.

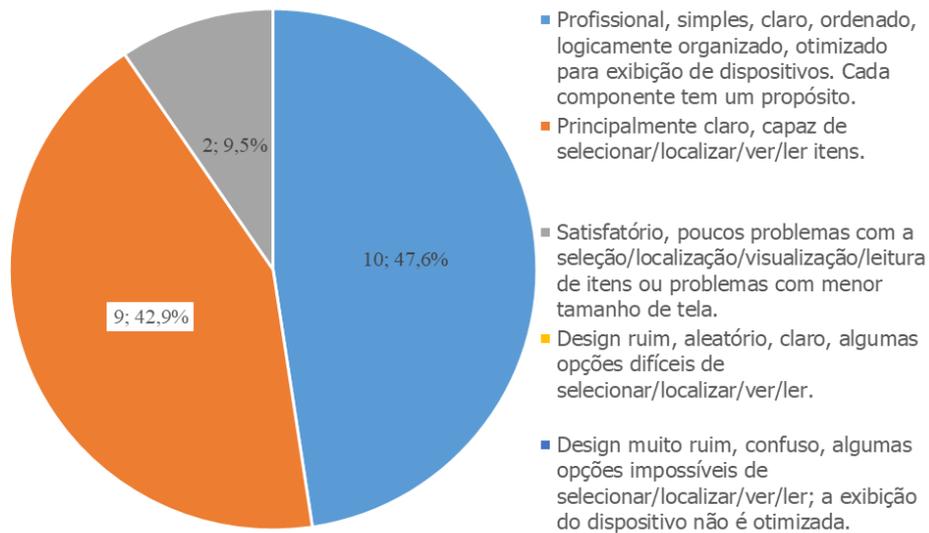
Figura 45 – Questão 03: As interações (toques/swipes/pinches/scrolls) são consistentes e intuitivas em todos os componentes/telas?



Fonte: Autoria própria

Na quarta questão foi avaliado o *layout* do ARtEMIS considerando a organização e o tamanho dos botões, ícones, menus e conteúdo na tela. Como exhibe a Figura 46, aproximadamente 90,5% (47,6% + 42,9%) dos participantes achou simples, claro, logicamente organizado e capaz de selecionar, localizar, ver e ler itens.

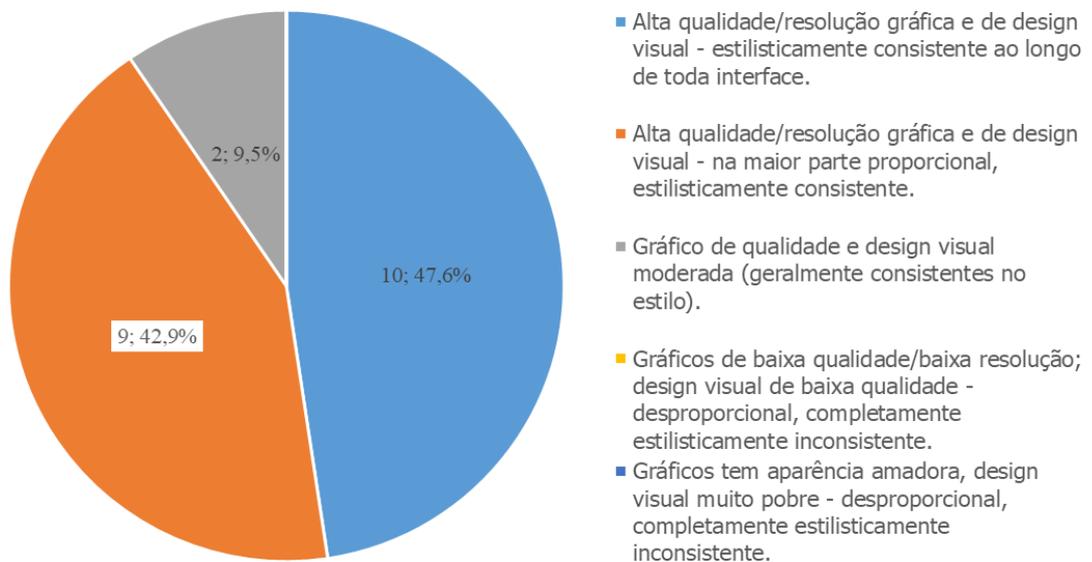
Figura 46 – Questão 04: A organização e o tamanho dos botões/ícones/menus/conteúdo na tela são apropriados ou ampliáveis (zoom) se necessário?



Fonte: Autoria própria

Na quinta questão foram avaliados os gráficos do ARtEMIS considerando a qualidade/resolução dos gráficos usados nos botões, ícones, menus e conteúdo. Como se pode ver na Figura 47, em torno de 90,5% (47,6% + 42,9%) dos participantes acharam a qualidade/resolução gráfica alta e o *design* visual consistente ao longo de toda *interface* ou na maior parte.

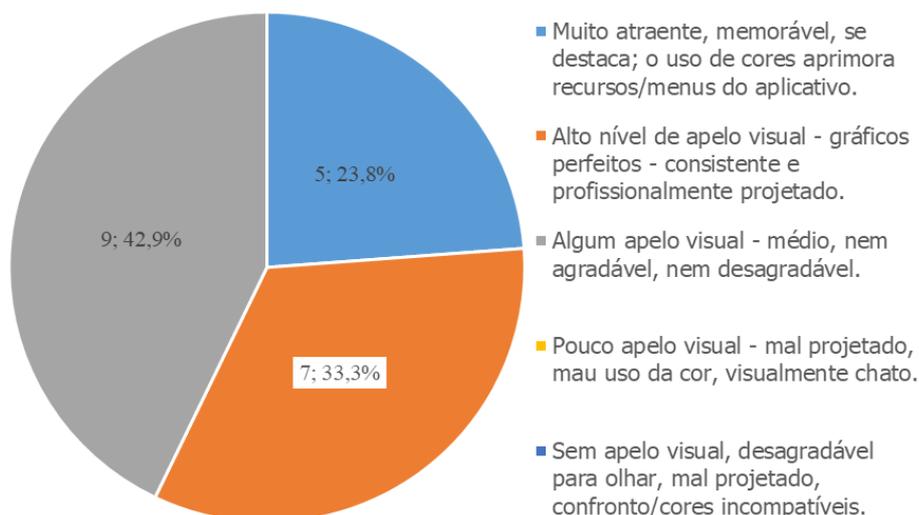
Figura 47 – Questão 05: Qual é a qualidade/resolução dos gráficos usados para os botões/ícones/menus/conteúdo?



Fonte: Autoria própria

Na sexta questão foi avaliado o apelo visual do ARtEMIS, isto é, quão boa é a aparência do aplicativo. Como mostra a Figura 48 a maioria (33,3% + 42,9%) dos participantes achou o nível de apelo visual médio ou alto.

Figura 48 – Questão 06: Quão boa é a aparência do aplicativo?

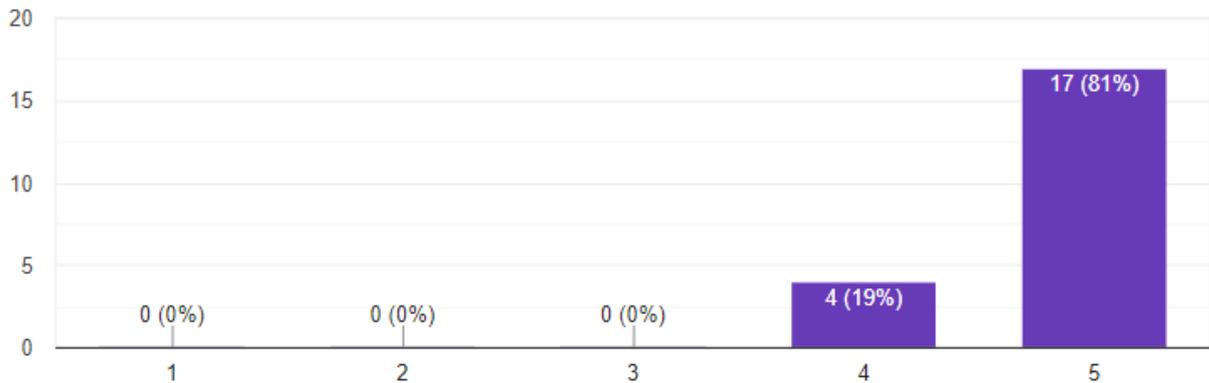


Fonte: Autoria própria

5.1.3.2 Avaliação das estratégias empregadas no desenvolvimento do ARtEMIS

As questões dessa seção permitiram aos participantes avaliar o emprego da gamificação e da RA no desenvolvimento do ARtEMIS. Na primeira questão buscou-se saber o quão importante os participantes acharam a inclusão da gamificação no ARtEMIS para estimular seu uso e o aprendizado de OAC. A Figura 49 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Não importante e 5 - Extremamente importante). A média dessa distribuição foi 4,8 $((5*17+4*4)/21)$.

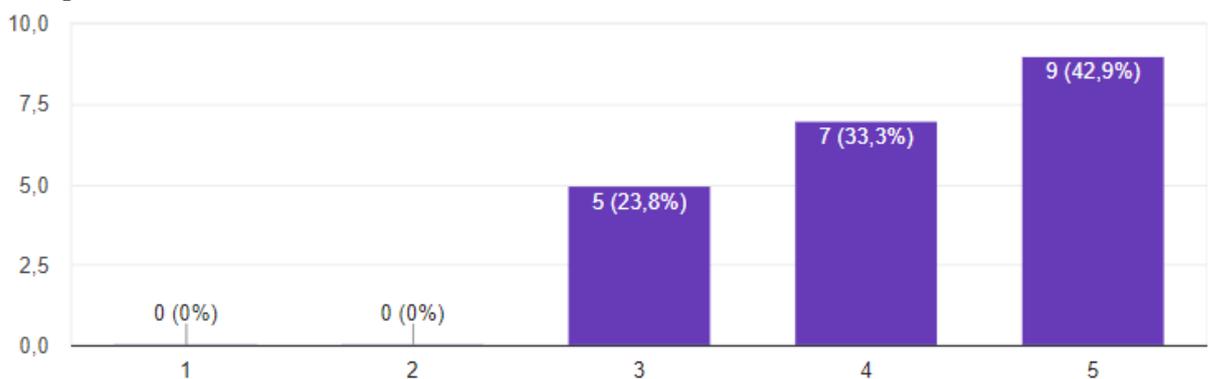
Figura 49 – Questão 01: Quão importante você achou a inclusão da gamificação (inserção dos elementos de jogo fases, pontos e conquistas) no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores?



Fonte: Autoria própria

Na segunda questão procurou-se saber o quão importante os participantes acharam a inclusão da Realidade Aumentada no ARtEMIS para estimular seu uso e o aprendizado de OAC. A Figura 50 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Não importante e 5 - Extremamente importante). A média dessa distribuição foi 4,2 $((5*9+4*7+3*5)/21)$.

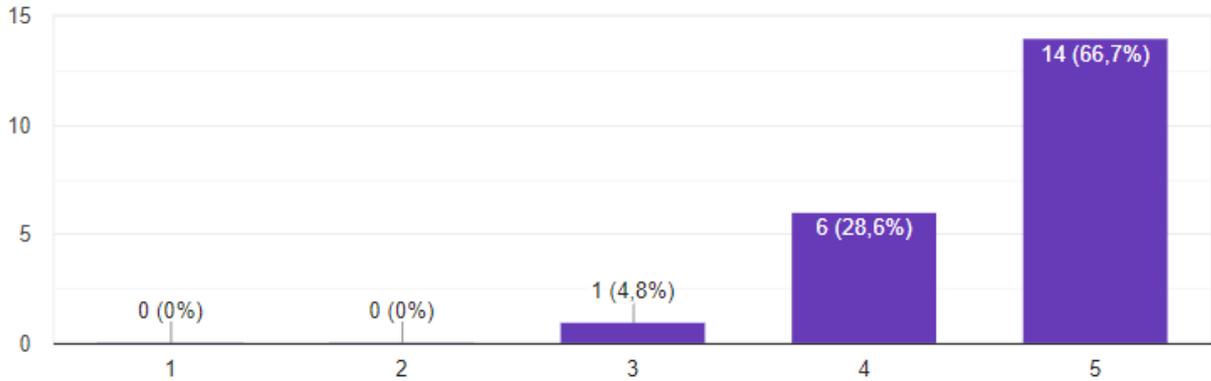
Figura 50 – Questão 02: Quão importante você achou a inclusão da Realidade Aumentada no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores?



Fonte: Autoria própria

Na terceira questão buscou-se saber o quão útil os participantes acharam o ARtEMIS para aprender sobre OAC. A Figura 51 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Inútil e 5 - Extremamente útil). A média dessa distribuição foi 4,6 $((5*14+4*6+3*1)/21)$.

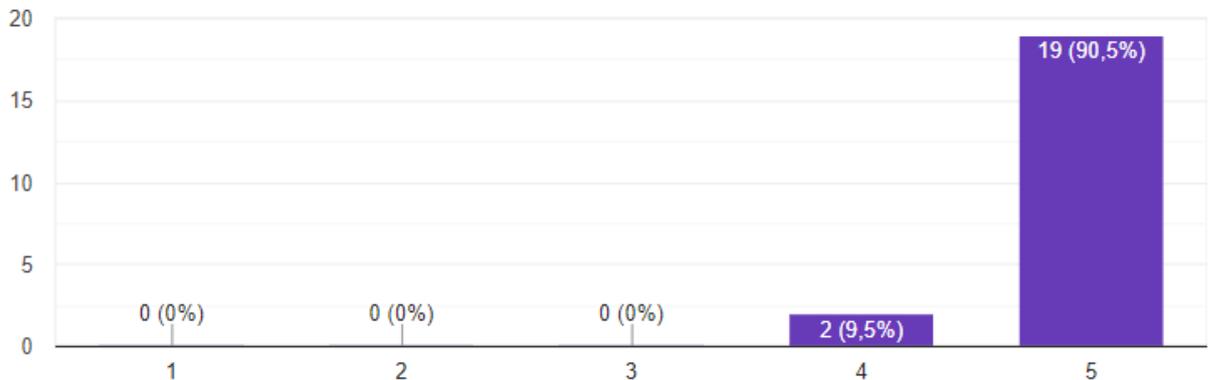
Figura 51 – Questão 03: Quão útil você achou o ARtEMIS para aprender sobre Organização e Arquitetura de Computadores?



Fonte: Autoria própria

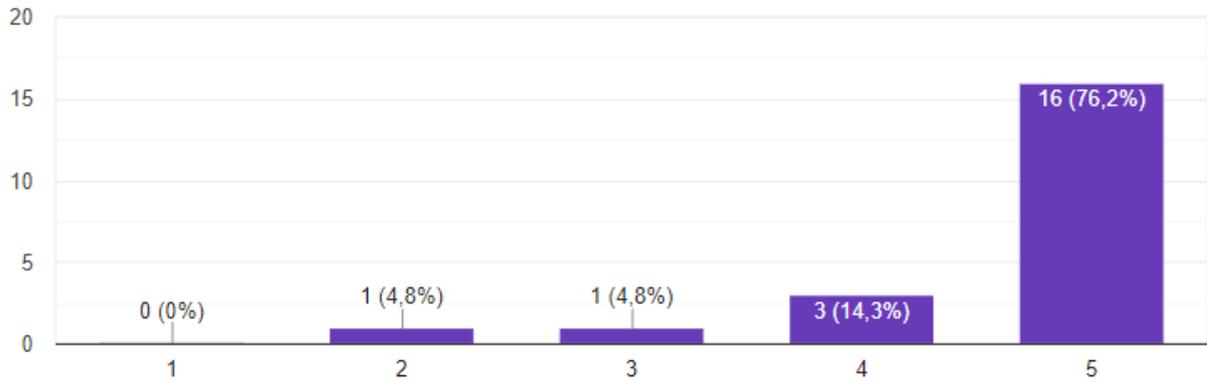
Na quarta questão procurou-se saber se o reconhecimento das figuras foi preciso. A Figura 52 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Impreciso e 5 - Extremamente preciso). A média dessa distribuição foi 4,9 $((5*19+4*2)/21)$.

Figura 52 – Questão 04: O reconhecimento das figuras foi preciso?



Fonte: Autoria própria

Na quinta questão buscou-se saber se o reconhecimento das figuras foi rápido. A Figura 53 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Extremamente lento 5 - Extremamente rápido). A média dessa distribuição foi 4,6 $((5*16+4*3+3*1+2*1)/21)$.

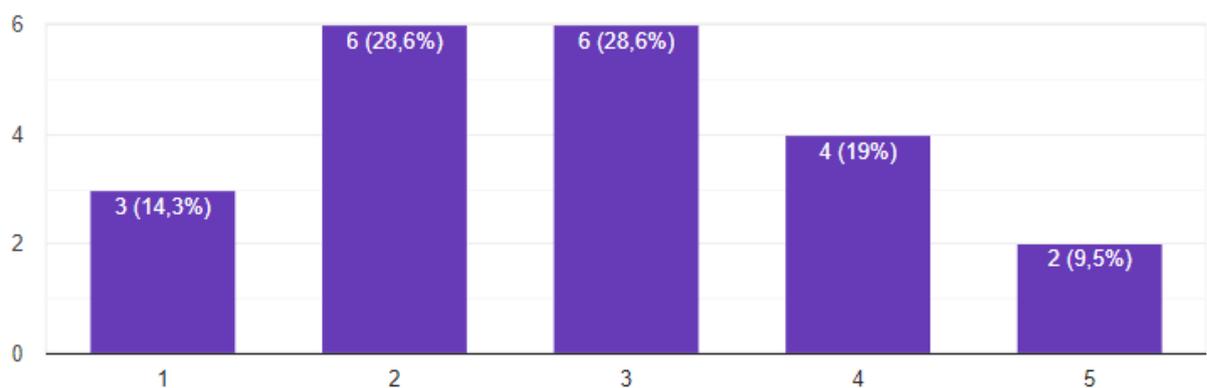
Figura 53 – Questão 05: O reconhecimento das figuras foi rápido?

Fonte: Autoria própria

5.1.3.3 Avaliação das atividades propostas

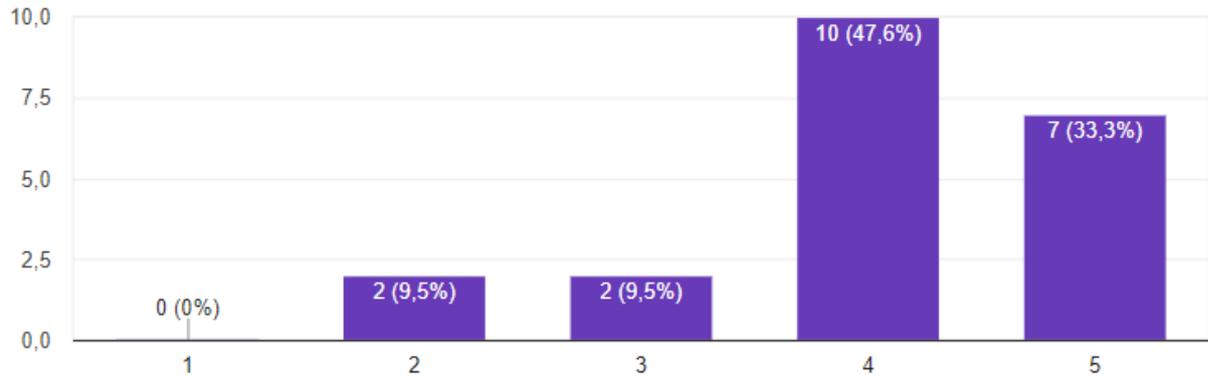
As questões dessa seção permitiram aos participantes avaliar a contribuição do pré-teste e do ARtEMIS para o aprendizado e os exercícios presentes neles.

Na primeira questão buscou-se saber quanto os participantes acham que aprenderam respondendo o questionário pré-teste. A Figura 54 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Nenhum aprendizado e 5 - Aprendi bastante). A média dessa distribuição foi 2,8 $((5*2+4*4+3*6+2*6+1*3)/21)$.

Figura 54 – Questão 01: Quanto você acha que aprendeu respondendo o questionário pré-teste?

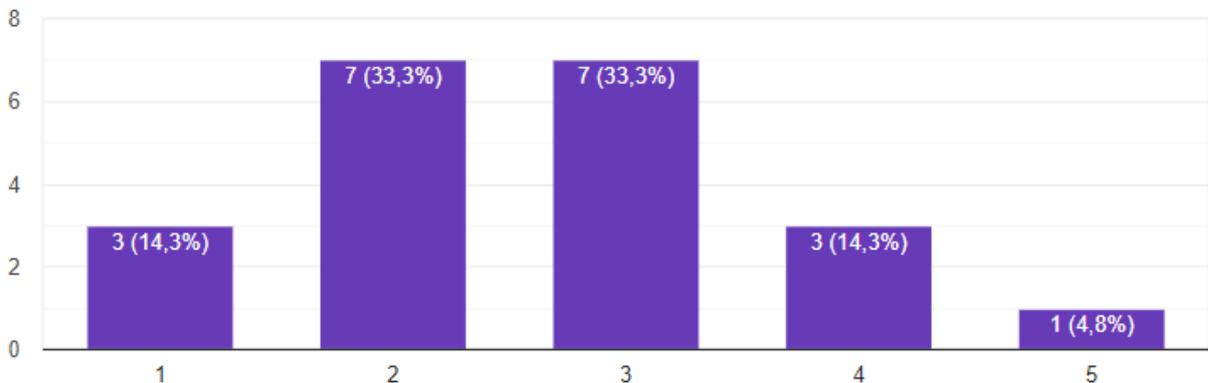
Fonte: Autoria própria

Na segunda questão procurou-se saber quanto os participantes acham que aprenderam usando o ARtEMIS. A Figura 55 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Nenhum aprendizado e 5 - Aprendi bastante). A média dessa distribuição foi 4 $((5*7+4*10+3*2+2*2)/21)$.

Figura 55 – Questão 02: Quanto você acha que aprendeu usando o ARtEMIS?

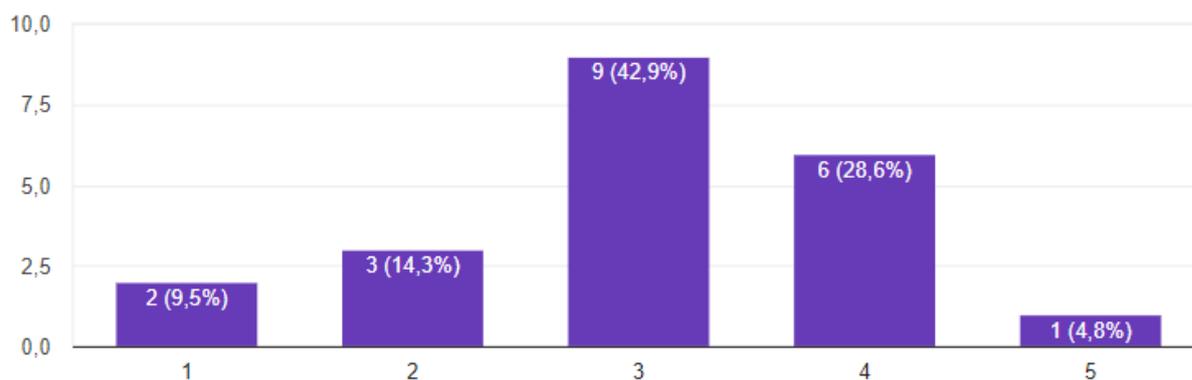
Fonte: Autoria própria

Na terceira questão buscou-se saber quão difíceis os participantes acharam as questões apresentadas no questionário pré-teste. A Figura 56 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Extremamente difíceis e 5 - Extremamente fáceis). A média dessa distribuição foi 2,6 $((5*1+4*3+3*7+2*7+1*3)/21)$.

Figura 56 – Questão 03: Quão difíceis foram as questões apresentadas no questionário pré-teste?

Fonte: Autoria própria

Na quarta questão procurou-se saber quão difíceis os participantes acharam as questões apresentadas no ARtEMIS. A Figura 57 mostra a distribuição das opiniões dos participantes em uma escala Likert de 5 pontos (1 - Extremamente difíceis e 5 - Extremamente fáceis). A média dessa distribuição foi 3 $((5*1+4*6+3*9+2*3+1*2)/21)$.

Figura 57 – Questão 04: Quão difíceis foram as questões apresentadas no ARtEMIS?

Fonte: Autoria própria

5.2 Considerações finais

Este capítulo apresentou os resultados da validação do ARtEMIS. Com base na análise dos resultados obtidos nas 3 etapas da validação foram feitas algumas constatações. Com relação ao impacto no desempenho, verificou-se que o ARtEMIS é capaz de promover melhora no desempenho dos alunos. A usabilidade foi bem avaliada na maioria dos aspectos analisados. Tanto a RA quanto a gamificação foram consideradas estratégias importantes para estimular o aprendizado. O ARtEMIS foi considerado muito útil para auxiliar no aprendizado de OAC e os participantes acharam que aprenderam muito usando-o.

Na Figura 40 e Figura 41 são mostrados, respectivamente, que os participantes que cursaram OAC melhoram seu desempenho mais que os participantes que não cursaram OAC e que os que estudaram o processador MIPS melhoraram o desempenho mais do que os que não estudaram. Na Figura 42, embora os participantes que cursaram OAC e não estudaram o processador MIPS tenham tido uma melhora de desempenho superior à dos participantes que cursaram OAC e estudaram o processador MIPS, o ARtEMIS possibilitou que ambos os grupos revisassem conteúdos estudados em OAC. Com isso, pode-se inferir que o ARtEMIS é mais indicado não para ensinar um conceito novo, mas para reforçar conteúdos já apresentados pelo professor. Isso reforça a ideia do ARtEMIS como material auxiliar/apoio para ser usado em conjunto com o livro.

6 CONCLUSÃO

Neste capítulo, apresenta-se as conclusões deste trabalho, os trabalhos futuros e as publicações.

O livro “Organização e Projeto de Computadores” de Patterson e Hennessy é um dos livros mais utilizados no ensino de OAC e do processador MIPS e por isso já foram criadas diversas soluções de simulação desse processador. Entretanto, elas não apresentam integração com tal livro. Considerando a importância de promover a integração entre o material de estudo e as ferramentas de apoio, neste trabalho, foi desenvolvido o simulador ARtEMIS.

O ARtEMIS foi desenvolvido para dispositivos móveis com SO Android, e permite reconhecer 4 ilustrações do livro (PATTERSON; HENNESSY, 2005), usando-as como marcadores para projetar a RA com a qual os alunos podem interagir. O simulador é uma ferramenta plenamente integrada ao livro a qual os alunos podem recorrer para facilitar a assimilação dos conceitos apresentados tendo como possibilidades a execução de instruções e obtenção de informações sobre o funcionamento das unidades funcionais do processador MIPS.

Considerando a capacidade da gamificação para aumentar a motivação e o engajamento das pessoas na realização de tarefas e os benefícios promovidos com sua aplicação no ensino superior (SUBHASH; CUDNEY, 2018), resolveu-se empregá-la no desenvolvimento do ARtEMIS, procurando promover não somente uma ferramenta integrada, mas também capaz de aumentar a motivação dos alunos ao estudar OAC. Dessa forma, no desenvolvimento do ARtEMIS foram empregados os elementos de jogos: fases, pontos de experiência (XP) e conquistas.

Assim como o livro, que apresenta as ilustrações do caminho do MIPS de forma incremental, cada vez com mais detalhes e compatibilidade com mais instruções, o esquema de fases faz com que o reconhecimento das figuras seja feito em sequência. Dessa forma, o ARtEMIS cria um fluxo para sua utilização como recurso didático. Nessa abordagem o reconhecimento de cada figura do livro funciona como uma fase contendo uma ou mais subfases. Em cada subfase há uma missão com algum propósito didático que o aluno precisa cumprir sendo levado a interagir com os recursos disponibilizados pelo ARtEMIS e em seguida realiza desafios sobre o que aprendeu. Os desafios são constituídos de algumas questões. A quantidade de questões que o aluno acerta é contabilizada, e caso a quantidade mínima requerida seja atingida, ele pode avançar para a próxima subfase/fase, caso contrário, é necessário refazer o desafio.

Os XPs e as conquistas visam estimular o aluno a interagir com o simulador e a buscar liberar todas as fases. O aluno ganha XP executando instruções (1 ponto), passando de subfase (100 pontos), passando de fase (1000 pontos) e resolvendo os desafios (2 pontos para cada questão correta mais 10 pontos se todas as questões estiverem corretas). As conquistas são liberadas pelo usuário ao longo do uso do ARtEMIS. No total são 12 conquistas que estimulam a execução de todas instruções de uma determinada categoria e o avanço das fases.

Outra contribuição do ARtEMIS é estender o caminho de dados representados nas ilustrações 5.17 e 5.24 do livro, adicionando componentes ao modelo 3D delas que são necessários para a execução de instruções que são abordadas no livro, mas cujos caminhos de dados necessários à sua execução não são apresentados. Isso torna o ARtEMIS um recurso adicional ao livro.

Na validação realizada buscou-se verificar o impacto do ARtEMIS no aprendizado e avalia-lo em termos de usabilidade e utilidade para o ensino/aprendizado. Para verificar o impacto no aprendizado foi realizado um pré-teste e um pós-teste com os participantes e o desempenho que eles obtiveram em ambos foi comparado. Verificou-se que 60% dos participantes melhoraram o desempenho do pós-teste em relação ao pré-teste. Esse resultado mostra a capacidade do ARtEMIS em promover melhora no desempenho dos alunos. Após a realização do pós-teste os participantes responderam um questionário de avaliação do ARtEMIS. Nesse questionário foram considerados aspectos como a usabilidade, a importância das estratégias de RA e gamificação para o aprendizado e a utilidade do simulador para aprender OAC. A usabilidade foi bem avaliada na maioria dos aspectos analisados; tanto a RA quanto a gamificação foram consideradas estratégias importantes para estimular o aprendizado; o ARtEMIS foi considerado muito útil para auxiliar no aprendizado de OAC e os participantes acharam que aprenderam muito usando-o.

A análise dos dados obtidos na validação permitiu também observar que os alunos que já haviam cursado OAC e/ou estudado o processador MIPS tiveram desempenho melhor que os outros participantes. Com essa constatação infere-se que o ARtEMIS é mais indicado não para ensinar um conceito novo, mas para reforçar conteúdos já apresentados pelo professor.

6.1 Trabalhos Futuros

Como trabalhos futuros para melhorar o ARtEMIS lista-se:

- Reconhecimento de mais das figuras do livro relativas os caminhos de dados *pipeline*. Bem como o reconhecimento das figuras na versão em inglês do livro e também de outras edições.
- Adicionar mais exercícios às figuras, os quais poderão retornar um *feedback* para o professor, e indicar quais assuntos determinado aluno ou grupo precisam de mais reforço.
- Incluir na gamificação do ARtEMIS um mecanismo de *ranking*, como forma de intensificar a motivação.
- Desenvolver os Guias de Uso de cada fase para o professor e para os alunos.
- Disponibilizar tanto a ferramenta quanto os Guias de Uso via web.

6.2 Publicações

Durante a realização desse trabalho, foram publicados os seguintes artigos:

- **Título:** Uso de Realidade Aumentada para Ensino de Arquitetura de Computadores com MIPS.
Veículo: Anais do XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018) – Workshop sobre Educação em Computação (WEI).
Situação: Aceito.
- **Título:** Proposta de Ensino de Arquitetura de Computadores com Gamificação e Realidade Aumentada.
Veículo: International Journal of Computer Architecture Education (IJCAE 2018)
Situação: Aceito.

REFERÊNCIAS

- ARAÚJO, M. R. D.; PÁDUA, F. L. C.; ANDRADE, F. V.; CORRÊA JUNIOR, F. L. MIPS X-Ray: A MARS Simulator Plug-in for Teaching Computer Architecture. **International Journal Of Recent Contributions From Engineering, Science & It (ijes)**, [s.l.], v. 2, n. 2, p.36-42, 2014. International Association of Online Engineering (IAOE). Disponível em: <<http://dx.doi.org/10.3991/ijes.v2i2.3527>>.
- AKRAM, A.; SAWALHA, L. A Survey of Computer Architecture Simulation Techniques and Tools. **Ieee Access**, [s.l.], v. 7, p. 78120-78145, 2019. Disponível em: <<http://dx.doi.org/10.1109/access.2019.2917698>>.
- AKÇAYIR, M.; AKÇAYIR, G. Advantages and challenges associated with augmented reality for education: A systematic review of the literature. **Educational ResearchReview**, [s.l.], v. 20, p.1-11, 2017. Disponível em: <<http://dx.doi.org/10.1016/j.edurev.2016.11.002>>.
- AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, [s.l.], v. 6, n. 4, p. 355-385, 1997. MIT Press - Journals. Disponível em: <<http://dx.doi.org/10.1162/pres.1997.6.4.355>>.
- BIANCHINI, C. P.; SILVA L. Sistemas de Realidade Aumentada Móvel Suportados por Computação em Nuvem. **Tendências e Técnicas em Realidade Virtual e Aumentada**, Salvador, v. 4, p. 9-32, 2014. Disponível em: <http://hiperrealidade.com.br/ce-rv/MC_SVR_2014.pdf>.
- BIOLCHINI, J. C. A.; MIAN, P. G.; NATALI, A. C. C. CONTE, T. U; TRAVASSOS, G. H. Scientific research ontology to support systematic review in software engineering. **Advanced Engineering Informatics**, [s.l.], v. 21, n. 2, p.133-151, 2007. Disponível em: <<http://dx.doi.org/10.1016/j.aei.2006.11.006>>.
- BRANOVIC, I; GIORGI, R; MARTINELLI, E. WebMIPS. **Proceedings Of The 2004 Workshop On Computer Architecture Education Held In Conjunction With The 31st International Symposium On Computer Architecture - Wcae '04**, Munique, p.93-98, 2004. Disponível em: <10.1145/1275571.1275596>.
- BRUM, L. M. L.; PINHO, L. B.; CAMARGO, S. S. Avaliação do Uso de Realidade Aumentada no Ensino de Arquitetura e Organização de Computadores. **International Journal of Computer Architecture Education (IJCAE)**, [s.l.], v. 6, n. 1, p. 10-17, 2017. Disponível em: <http://www2.sbc.org.br/ceacpad/ijcae/v6_n1_dec_2017/IJCAE_v6_n1_dez_2017_paper_2_vf.pdf>.
- CAPES. Tabela de Áreas do Conhecimento. Disponível em: <http://www.capes.gov.br/images/documentos/documentos_diversos_2017/TabelaAreasConhecimento_072012_atualizada_2017_v2.pdf>

DEITEL, P; DEITEL, H. **Java: como programar**. 8.ed. São Paulo: Pearson, 2010. 1176 p

ESMERALDO, G. A.; LISBOA E. B., Uma Ferramenta para Exploração do Ensino de Organização e Arquitetura de Computadores. **International Journal of Computer Architecture Education (IJCAE)**, [s.l.], v. 6, n. 1, p. 68-75, 2017. Disponível em: <http://www2.sbc.org.br/ceacpad/ijcae/v6_n1_dec_2017/IJCAE_v6_n1_dez_2017_paper_9_vf.pdf>.

FELIX, A. F.; POUSA, C. V.; CARVALHO, M. B. DIMIPSS: Um simulador didático e interativo do MIPS. **Workshop Sobre Educação em Arquitetura de Computadores (weac)**, Ouro Preto, p.49-52, 2006.

FERNANDES, S. R.; SILVA, I. S. Relato de Experiência Interdisciplinar Usando MIPS, **International Journal of Computer Architecture Education (IJCAE)**, [s.l.], v. 6, n. 1, p. 52-61, 2017. Disponível em: <http://www2.sbc.org.br/ceacpad/ijcae/v6_n1_dec_2017/IJCAE_v6_n1_dez_2017_paper_7_vf.pdf>.

GERSNOVIEZ, A. et al. UCOMIPSIM 2.0: Pipelined MIPS Architecture Simulator. **2018 XIII Technologies Applied To Electronics Teaching Conference (taee)**, Tenerife, p. 406-411, 2018. IEEE. Disponível em: <<http://dx.doi.org/10.1109/taee.2018.8476063>>.

HENDRICH, N. A Java-Based Framework for Simulation and Teaching: HADES — the Hamburg Design System. **Microelectronics Education**, Dordrecht, p. 285-288, 2000. Springer Netherlands. Disponível em: <http://dx.doi.org/10.1007/978-94-015-9506-3_66>.

HERVAS, R.; RUIZ-CARRASCO, D.; MONDEJAR, T.; BRAVO, J. Gamification mechanics for behavioral change: A systematic review and proposed taxonomy. **Proceedings Of The 11th Eai International Conference On Pervasive Computing Technologies For Healthcare - Pervasivehealth '17**, Barcelona, p. 395-404, 2017. Disponível em: <<http://dx.doi.org/10.1145/3154862.3154939>>.

KABIR, Md T.; BARI, M. T.; HAQUE, A. L. ViSiMIPS: Visual simulator of MIPS32 pipelined processor. **2011 6th International Conference On Computer Science & Education (iccse)**, Singapura, p. 788-793, 2011. IEEE. Disponível em: <<http://dx.doi.org/10.1109/iccse.2011.6028756>>.

KIM, Y.; CHOI, Y.; LEE, H.; LEE, G.; BIANCHI, A. VirtualComponent: A Mixed-Reality Tool for Designing and Tuning Breadboarded Circuits. **Proceedings Of The 2019 Chi Conference On Human Factors In Computing Systems - Chi '19**, Glasgow, 2019. ACM Press. Disponível em: <<http://dx.doi.org/10.1145/3290605.3300407>>.

KIRNER, C.; KIRNER T. G.. Evolução e Tendências da Realidade Virtual e da Realidade Aumentada. **XIII Simpósio de Realidade Virtual e Aumentada**, Uberlândia, v. 1, p.10-25, 2011. Disponível em: <http://www.de.ufpb.br/~labteve/publi/2011_svrps.pdf>.

KIRNER, C.; SISCOOTTO, R. A. Fundamentos de Realidade Virtual e Aumentada. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações**. Petrópolis, p.2-21, 2007. Disponível em: < <http://www.ckirner.com/download/livros/Livro-RVA2007-1-28.pdf>>.

KITCHENHAM, B. Procedures for performing systematic reviews Technical Report TR/SE-0401, Keele University and NICTA, jul. 2004.

LEE, J.; HAMMER, J. Gamification in education: What, how, why bother?. **Academic Exchange Quarterly**, [s.l.], v. 15, n. 2, p. 146-151, 2011.

MARTIN-GUTIERREZ, J.; NAVARRO, R. E.; GONZALEZ, M. A. Mixed reality for development of spatial skills of first-year engineering students. **2011 Frontiers In Education Conference (fie)**, Rapid City, 2011. IEEE. Disponível em: < <http://dx.doi.org/10.1109/fie.2011.6142707>>.

MARWEDEL, P.; SIROCIC, B. Multimedia components for the visualization of dynamic behavior in computer architectures. **Proceedings Of The 2003 Workshop On Computer Architecture Education Held In Conjunction With The 30th International Symposium On Computer Architecture - Wcae '03**, San Diego, 2003. ACM Press. Disponível em: < <http://dx.doi.org/10.1145/1275521.1275539>>.

MILGRAM, Paul; KISHINO, Fumio. A taxonomy of mixed reality visual displays. **Ieice Transactions On Information And Systems**. vol. E77-D, n. 12, p. 1321-1329. 1994.

MIPS. MIPS – Market-leading RISC CPU IP processor solutions, 2019. Disponível em: < <https://www.mips.com/>> Acesso em: 05 julho 2019.

NAKOW, S.; KOLEV, V. **Fundamentals of programming with C#**. Veliko Tarnovo: Sofia, 2013. 1121 p.

NOVA, Bruno; FERREIRA, Joao C.; ARAUJO, Antonio. Tool to support computer architecture teaching and learning. **2013 1st International Conference Of The Portuguese Society For Engineering Education (cispee)**, Porto, 2013. IEEE. Disponível em: < <http://dx.doi.org/10.1109/cispee.2013.6701965>>.

PATTERSON, D.; HENNESSY, J. L. **Organização e projeto de computadores**: a interface hardware/software. 3. ed. Rio de Janeiro: Campus, 2005. 484 p.

PATTERSON, D.; HENNESSY, J. L. **Computer Organization and Design**: the hardware software interface. 5. ed. Rio de Janeiro: Morgan Kauffman, 2013. 793 p.

PENHA, J. C.; FONTES, G.; FERREIRA, R. MIPSFPGA: Um Simulador MIPS Incremental com Validação em FPGA. **International Journal of Computer Architecture Education (IJCAE)**, [s.l.], v.5, n.1, p. 19-25, dez. 2016.

SEABORN, K.; FELLS, I. F. Gamification in Theory and Action: A Survey. **International Journal of Human-Computer Studies**, [s.l.], v. 74, p. 14-31, 2015. Disponível em: < <http://dx.doi.org/10.1016/j.ijhcs.2014.09.006>>.

SIGNORI, G.; GUIMARÃES, J. C. F. Gamificação como Método de Ensino Inovador. **Journal International Active Learning**, Rio de Janeiro, v. 1, n. 1, p. 66-77, 2016.

SILVA, G. C.; OLIVEIRA, L. C.; FERNANDES, S. R. Uso de Realidade Aumentada para Ensino de Arquitetura de Computadores com MIPS. **WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO**, Natal, p. 577-586, 2018. Disponível em: <http://natal.uern.br/eventos/csbc2018/?page_id=197>.

STALLINGS, William. **Arquitetura e organização de computadores**. 5. ed. São Paulo: Pearson, 2004. 767 p.

STOYANOV, S. R.; HIDES, L; KAVANAGH, D. J.; ZELENKO, O.; TJONDRONEGORO, D.; MANI, M. Mobile app rating scale: a new tool for assessing the quality of health mobile apps. **JMIR Mhealth Uhealth**, [s.l.], v.3, n. 1, 2015. Disponível em: <<https://doi.org/10.2196/mhealth.3422>>

SUBHASH, S.; CUDNEY, E. A. Gamified learning in higher education: A systematic review of the literature. **Computers In Human Behavior**, [s.l.], v. 87, p. 192-206, 2018. Disponível em: <<https://doi.org/10.1016/j.chb.2018.05.028>>.

TANENBAUM, A. S. **Organização estruturada de computadores**. 5. ed. São Paulo: Pearson, 2006. 464 p.

VIANNA, Y.; VIANNA, M.; MEDINA, B.; TANAKA, S. **Gamification, Inc**: Como reinventar empresas a partir de jogos, 1. ed. Rio de Janeiro: MJV Press, 2013. 115 p.

VOLLMAR, Kenneth R; SANDERSON, Pete. A MIPS assembly language simulator designed for education. **Journal Of Computing Sciences In Colleges**. [s.l.], v.21, n. 1, p. 95-101, 2005.

V WTCC. V WTCC, 2017. Disponível em: <<http://wtcc.com.br/>>

APÊNDICES

APÊNDICE A – DIAGRAMAS UML

Figura 58 – Diagrama de casos de uso

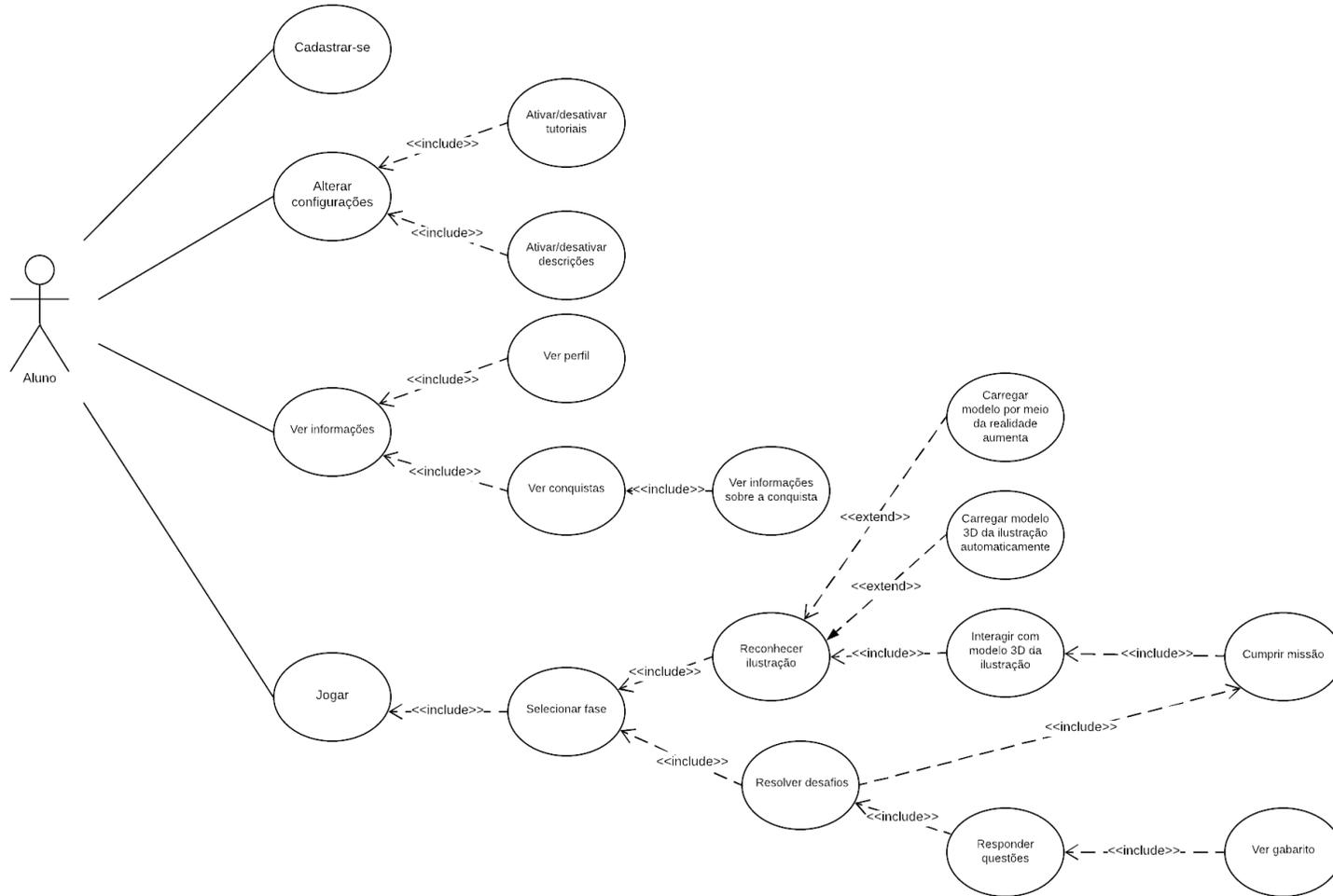


Figura 59 – Diagrama de Atividades para o caso de uso “Cadastrar-se”

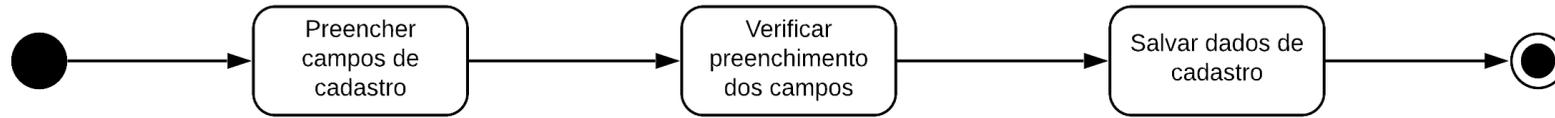


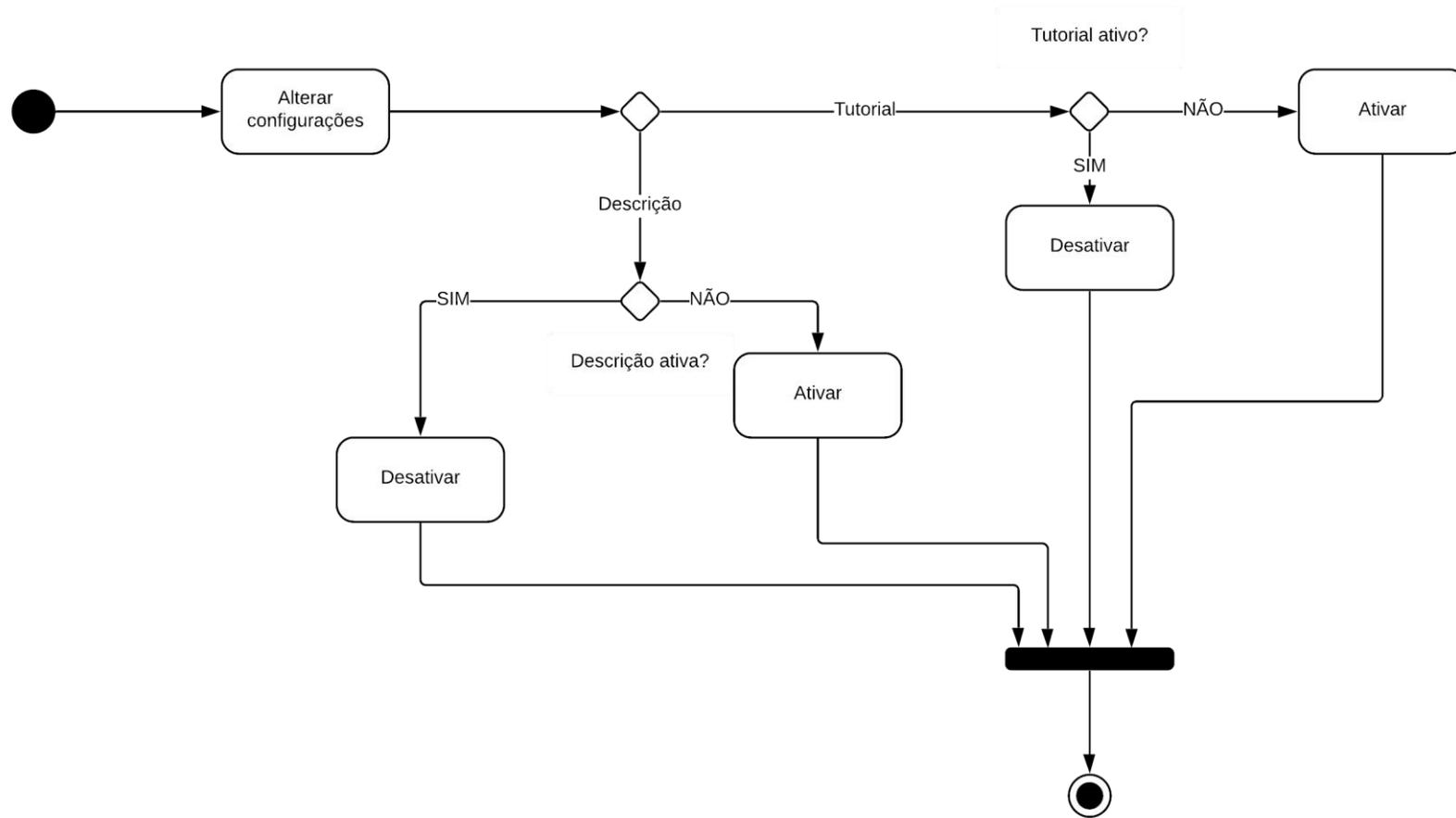
Figura 60 – Diagrama de Atividades para o caso de uso “Alterar Configurações”

Figura 61 – Diagrama de Atividades para o caso de uso “Ver Informações”

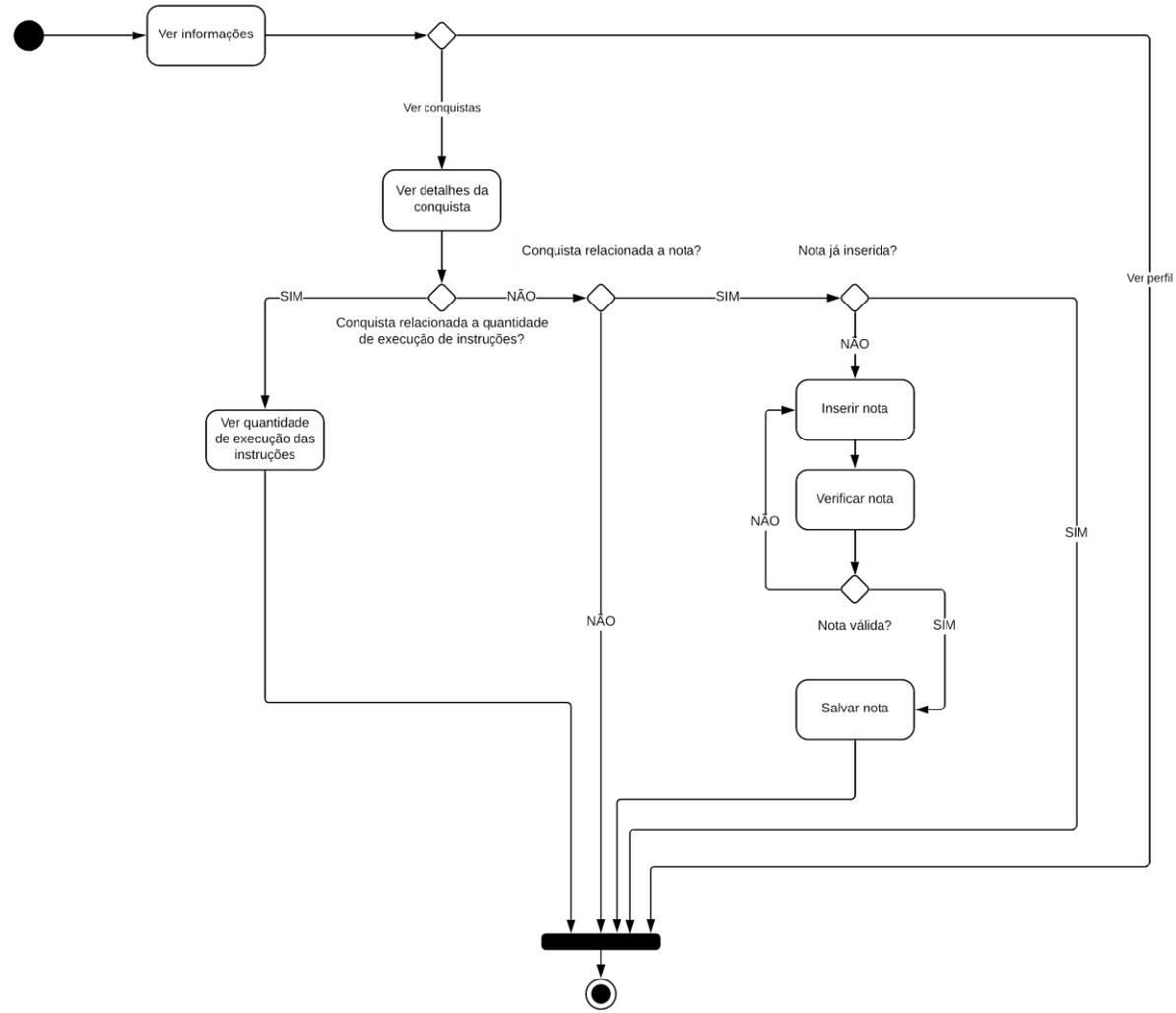
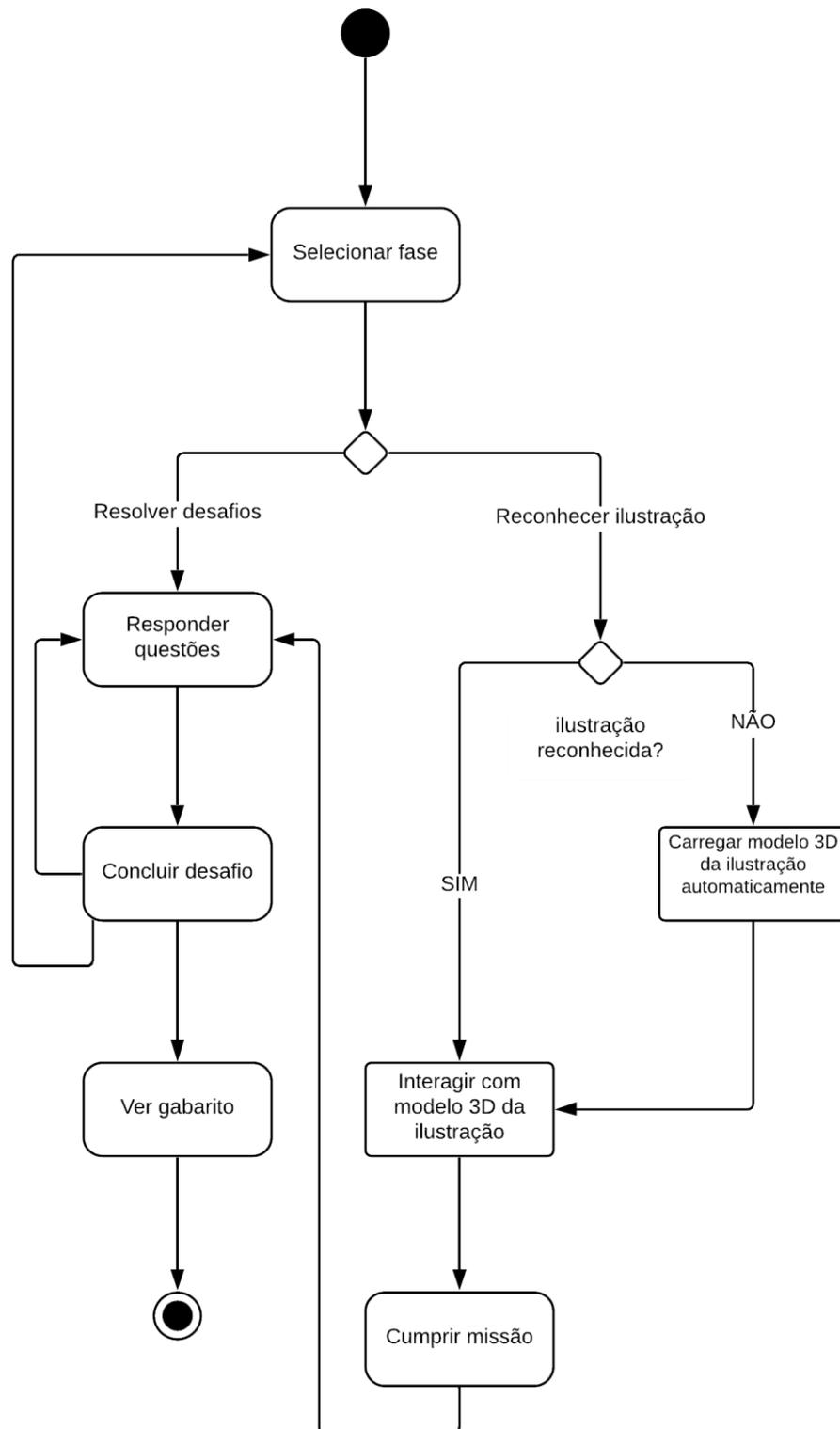


Figura 62 – Diagrama de Atividades para o caso de uso “Jogar”

APÊNDICE B – QUESTIONÁRIO PRÉ-TESTE

Este questionário é composto por 10 questões distribuídas em 5 seções e visa avaliar seus conhecimentos sobre o processador MIPS.

Informe seu nome:

Informe seu endereço de e-mail:

SEÇÃO 01

Julgue a afirmação abaixo como verdadeira ou falsa.

I (Questão 01) - O PC contém o endereço da instrução a ser executada.

- Verdadeiro
- Falso

SEÇÃO 02

Para cada questão abaixo forneça a resposta correta.

I (Questão 02) - Qual o registrador rs da instrução 'sub \$t0, \$t1, \$t2'?

II (Questão 03) - Qual valor a instrução 'and \$s0, \$s1, \$s2' salva em \$s0 se \$s1 contiver 5 e \$s2 contiver 2?

SEÇÃO 03

Para cada questão abaixo selecione a alternativa correta.

I (Questão 04) - Uma instrução 'lw' que traz para \$s1 o valor contido no endereço de memória contido no registrador \$s0 tem como sintaxe:

- a) lw \$s1, 0(\$s0)
- b) lw \$s0, 0(\$s1)
- c) lw \$s1, 8(\$s0)

II (Questão 05) - A instrução 'addi \$t0, \$t1, 4' tem como immediate o valor:

- a) de \$t0
- b) de \$t1
- c) 4

III (Questão 06) - A instrução 'beq' é uma instrução:

- a) De desvio
- b) Lógica
- c) Aritmética

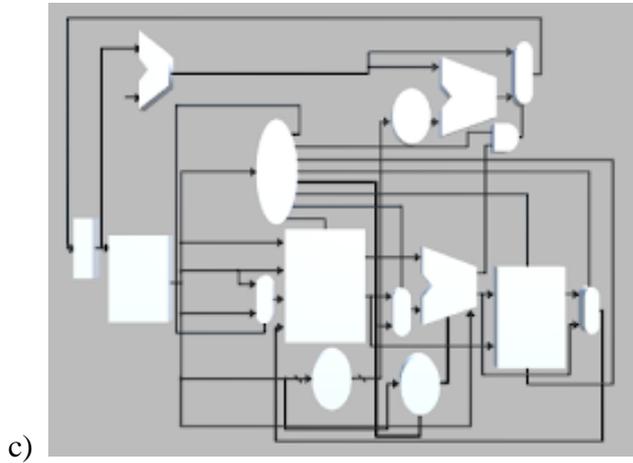
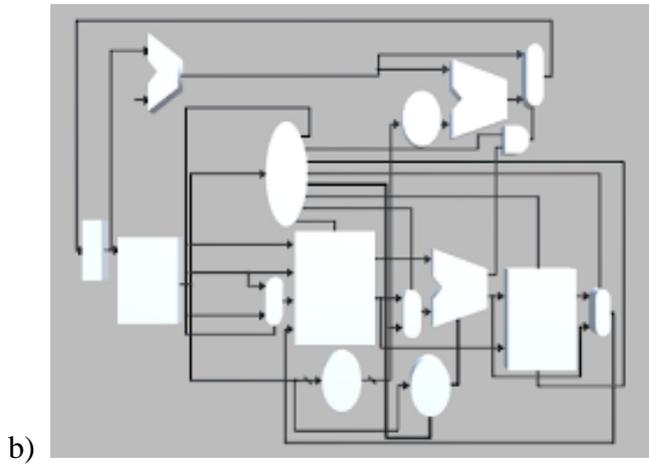
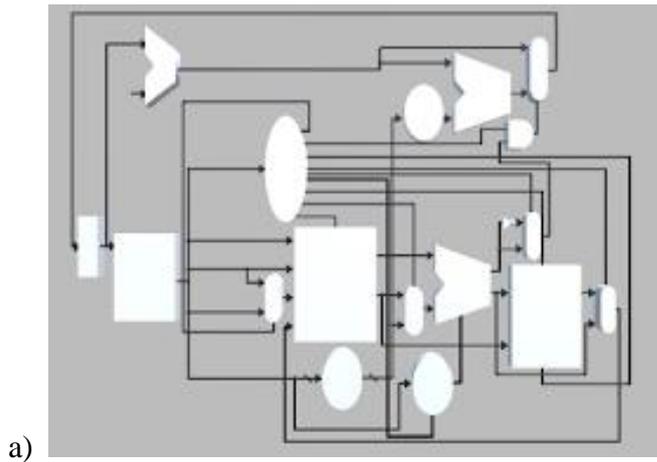
IV (Questão 07) - A instrução 'slt \$t0, \$t1, \$t2' salva o valor 1 no registrador \$t0 se:

- a) $\$t1 \neq \$t2$
- b) $\$t1 = \$t2$
- c) $\$t1 < \$t2$

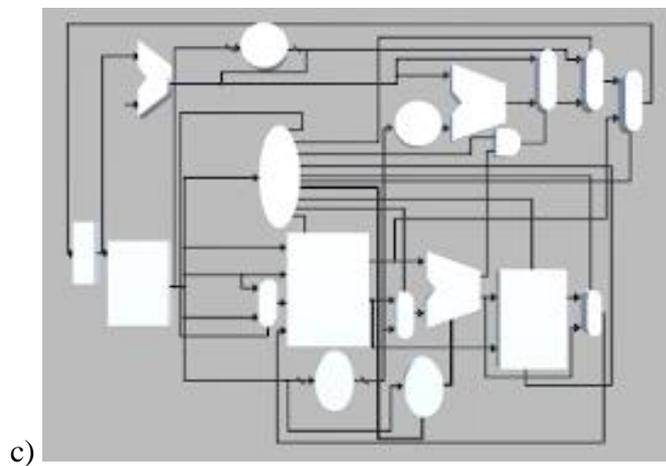
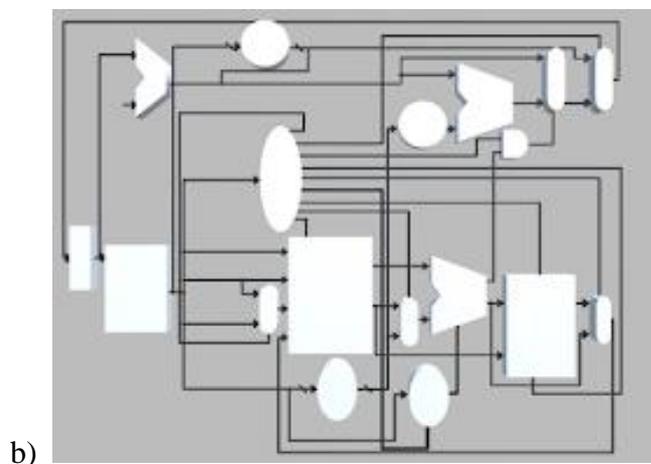
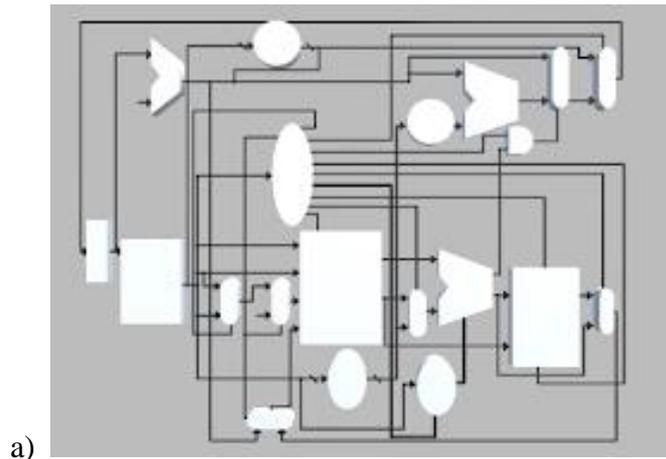
SEÇÃO 04

Para cada questão abaixo selecione a alternativa correta.

I (Questão 08) - Qual o caminho de dados mínimo para executar a instrução 'srl'?

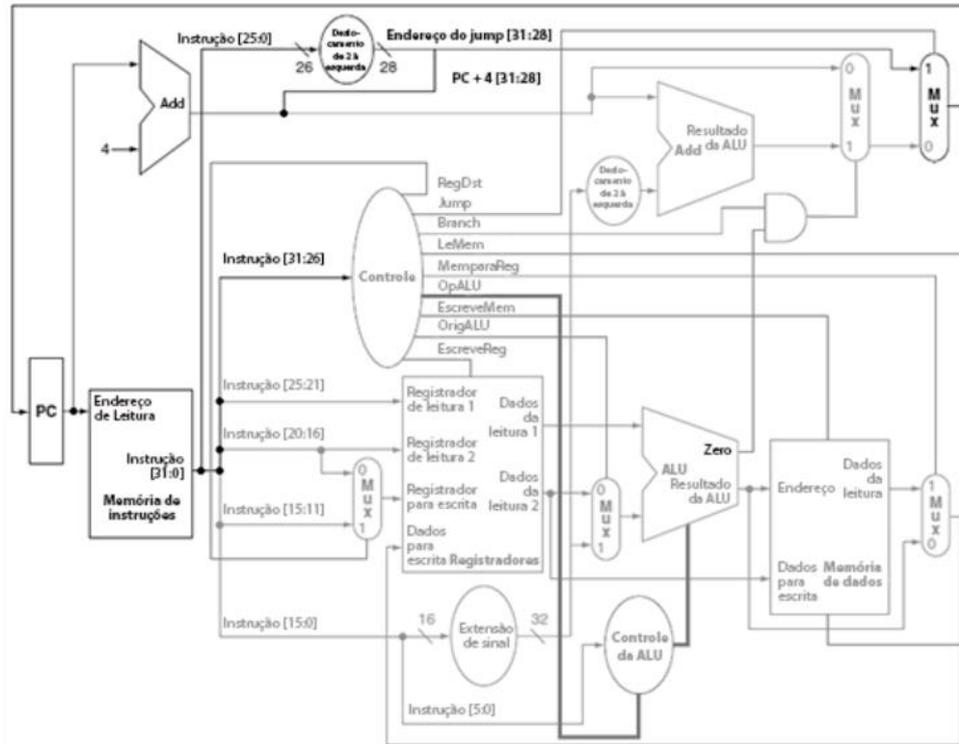


II (Questão 09) - Qual o caminho de dados mínimo para executar a instrução 'jal'?



SEÇÃO 05

I (Questão 10) - Selecione a opção que indica corretamente o valor (0, 1, 2 ou X - don't care) de cada sinal do controle quando uma instrução do tipo 'j' é executada.



- a) RegDst = 1; Jump = 0; Branch = 0; LeMem = 0; MemparaReg = 0; OpALU = 2; EscreveMem = 0; OrigALU = 0; EscreveReg = 1;
- b) RegDst = 1; Jump = 0; Branch = 0; LeMem = 0; MemparaReg = 0; OpALU = 1; EscreveMem = 0; OrigALU = 0; EscreveReg = 1;
- c) RegDst = 1; Jump = 0; Branch = 1; LeMem = 0; MemparaReg = 0; OpALU = 2; EscreveMem = 0; OrigALU = 0; EscreveReg = 1;

APÊNDICE C – QUESTIONÁRIO PÓS-TESTE

Este questionário é composto por 16 questões que foram apresentadas aos participantes sob a forma de desafios distribuídos pelas fases do ARtEMIS.

Informe seu nome:

Informe seu endereço de e-mail:

FASE 1-1

Julgue a afirmação abaixo como verdadeira ou falsa.

I (Questão 01) - O PC pode ser atualizado com o valor do primeiro ou do segundo somador.

Verdadeiro

Falso

II (Questão 02) - O primeiro somador soma 4 ao valor do PC.

Verdadeiro

Falso

III (Questão 03) - A memória de instruções contém o programa que será executado.

Verdadeiro

Falso

FASE 2-1

Para cada questão abaixo forneça a resposta correta.

I (Questão 04) - Qual o registrador *rd* da primeira instrução que você executou ('add \$t0, \$t1, \$t2')?

- a) \$t0
- b) \$t1
- c) \$t2

II (Questão 05) - Qual o valor armazenado em \$s0 após você executar a instrução 'sub'?

- a) 3
- b) 5
- c) 8

FASE 2-2

Para cada questão abaixo forneça a resposta correta.

I (Questão 06) - Qual o valor *immediate* da segunda instrução que você executou ('addi \$s0, \$t0, 5')?

- a) O valor de \$s0
- b) O valor de \$t0
- c) 5

II (Questão 07) - Qual das instruções abaixo não conseguiria cumprir a missão de inicializar \$t0 com o valor da mem[4]?

- a) lw \$t0, 4(\$t1)
- b) lw \$t0, 0(\$t1)
- c) lw \$t1, 0(\$t0)

FASE 3-1

Julgue a afirmação abaixo como verdadeira ou falsa.

I (Questão 08) - Deslocar o valor de \$t0 em 4 bits para a esquerda é o mesmo que multiplicá-lo por 16.

Verdadeiro

Falso

II (Questão 09) - O valor salvo em \$s0 se a instrução tiver sido 'srl' foi 256?

Verdadeiro

Falso

FASE 3-2

Para cada questão abaixo forneça a resposta correta.

I (Questão 10) - O valor salvo no registrador *rd* após a execução de *op1* foi:

a) 3

b) 7

c) 15

II (Questão 11) - Qual resultado da expressão apresentada na missão se *op1=nor*, *op2=or* e *op3=and*?

a) 5

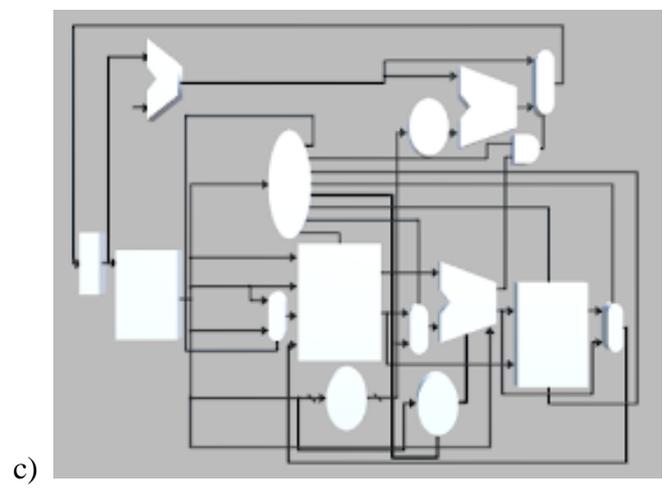
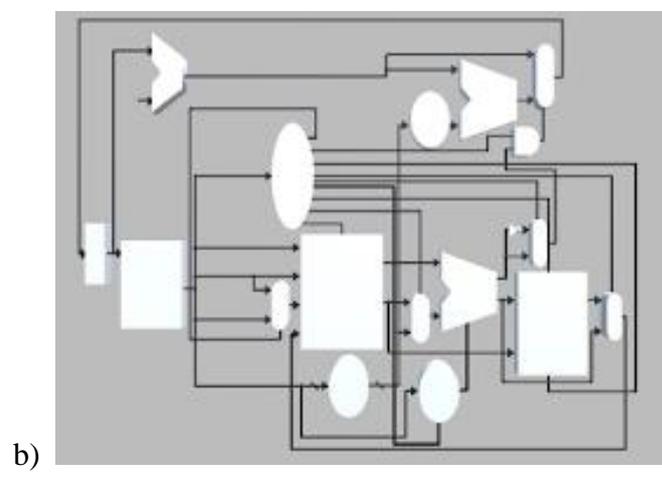
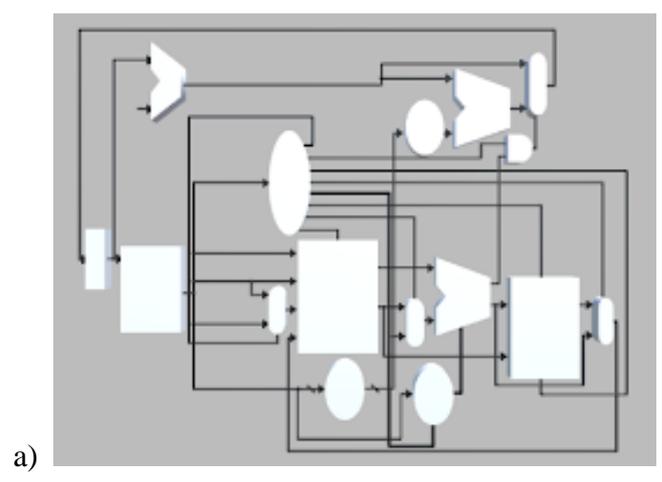
b) 8

c) 15

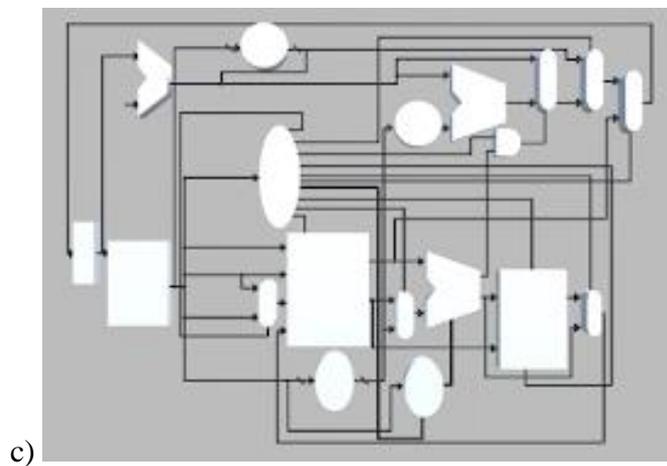
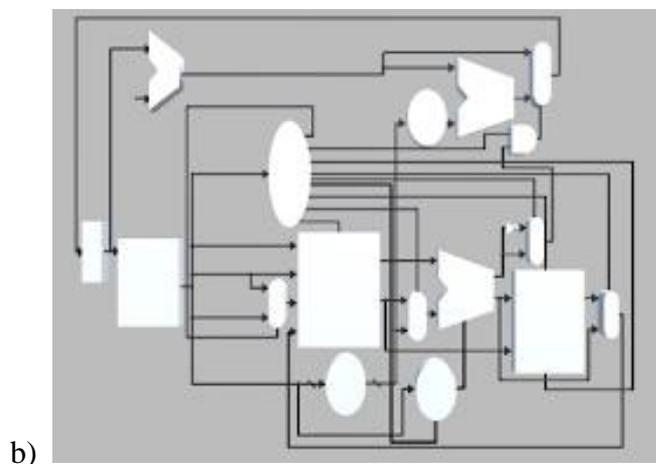
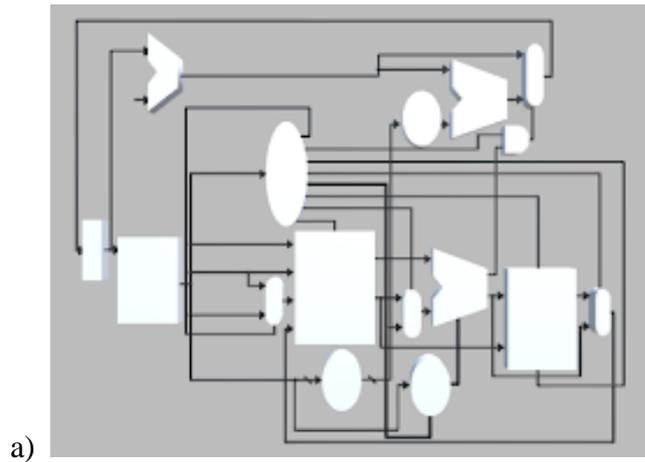
FASE 3-4

Para cada questão abaixo selecione a alternativa correta.

I (Questão 12) - Qual o caminho de dados mínimo para executar a instrução 'beq'?



II (Questão 13) - Qual o caminho de dados mínimo para executar a instrução 'jal'?



FASE 4-1

Julgue a afirmação abaixo como verdadeira ou falsa.

I (Questão 14) - Na missão foi necessário executar a instrução 'j' 2 vezes para que o valor de \$s0 fosse igual a 16.

Verdadeiro

Falso

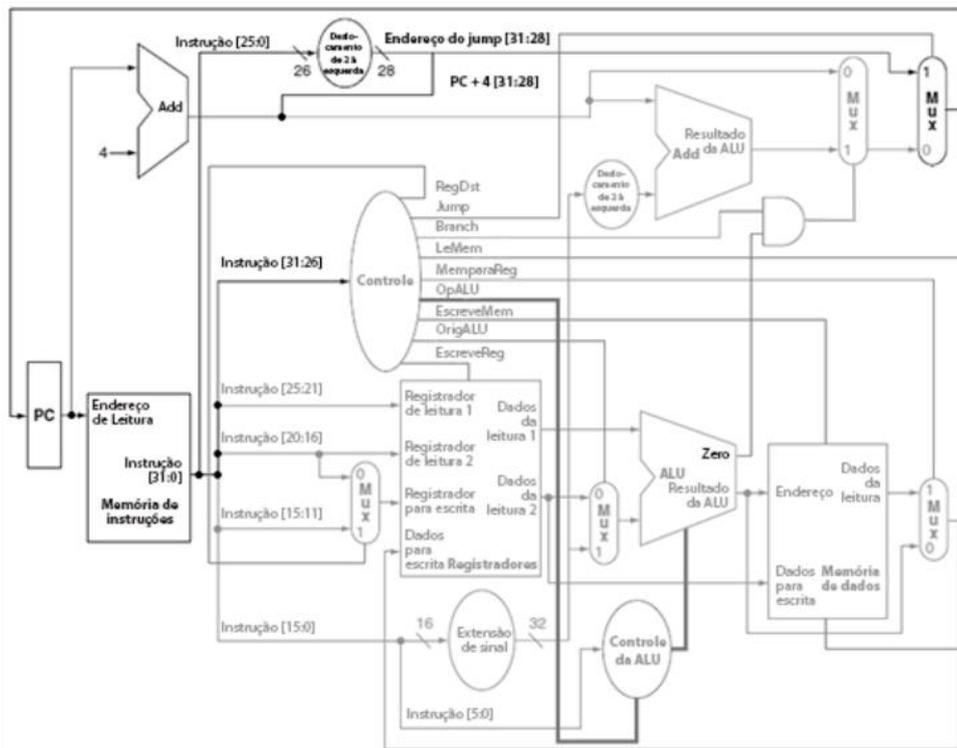
II (Questão 15) - A instrução 'j' é uma instrução de desvio incondicional.

Verdadeiro

Falso

FASE 4-2

I (Questão 16) - Indique o valor (0, 1, 2 ou X - don't care) de cada sinal do controle quando uma instrução do tipo 'add' é executada.



RegDst = _____

OpALU = _____

Jump = _____

EscreveMem = _____

Branch = _____

OrigALU = _____

LeMem = _____

EscreveReg = _____

MemparaReg = _____

APÊNDICE D – QUESTIONÁRIO DE AVALIAÇÃO DO ARTEMIS E DAS ATIVIDADES PROPOSTAS

Este questionário é composto por 16 questões distribuídas em 2 seções nas quais você avaliará o ARtEMIS e as atividades propostas.

Informe seu nome:

Informe seu endereço de e-mail:

SEÇÃO 01

Avaliação da interface do ARtEMIS.

I) Facilidade de uso: Quão fácil é aprender a usar o aplicativo; quão clara são as etiquetas/ícones do menu e instruções de uso?

- Capaz de usar o aplicativo imediatamente; intuitivo; simples;
- Fácil de aprender como usar o aplicativo (ou tem instruções de uso claras).
- Utilizável após algum tempo/esforço.
- Utilizável após muito tempo/esforço.
- Não há ou são instruções de uso limitadas; rótulos/ícones de menu são confusos; complicado.

II) Navegação: A movimentação entre as telas é lógica/precisa/apropriada/ininterrupta; todos os links de tela necessários estão presentes?

- Fluxo de tela perfeitamente lógico, fácil, claro e intuitivo ou oferece atalhos.
- Fácil de usar ou com problema de link insignificante.
- Utilizável após algum tempo/esforço.
- Utilizável após muito tempo/esforço.
- Seções diferentes dentro do aplicativo parecem logicamente desconectadas e aleatórias/confusas/ navegação é difícil.

III) Projeto gestual: As interações (toques/swipes/pinches/scrolls) são consistentes e intuitivas em todos os componentes/telas?

- Perfeitamente consistente e intuitivo.
- Principalmente consistente/intuitivo com problemas insignificantes.
- OK, mas com algumas inconsistências/elementos confusos.
- Frequentemente inconsistente/confuso.
- Completamente inconsistente/confuso.

IV) Layout: A organização e o tamanho dos botões/ícones/menus/conteúdo na tela são apropriados ou ampliáveis (zoom) se necessário?

- Profissional, simples, claro, ordenado, logicamente organizado, otimizado para exibição de dispositivos. Cada componente tem um propósito.
- Principalmente claro, capaz de selecionar/localizar/ver/ler itens.
- Satisfatório, poucos problemas com a seleção/localização/visualização/leitura de itens ou problemas com menor tamanho de tela.
- Design ruim, aleatório, claro, algumas opções difíceis de selecionar/localizar/ver/ler.
- Design muito ruim, confuso, algumas opções impossíveis de selecionar/localizar/ver/ler; a exibição do dispositivo não é otimizada.

V) Gráficos: Qual é a qualidade/resolução dos gráficos usados para os botões/ícones/menus/conteúdo?

- Alta qualidade/resolução gráfica e de design visual - estilisticamente consistente ao longo de toda interface.
- Alta qualidade/resolução gráfica e de design visual - na maior parte proporcional, estilisticamente consistente.
- Gráfico de qualidade e design visual moderada (geralmente consistentes no estilo).
- Gráficos de baixa qualidade/baixa resolução; design visual de baixa qualidade - desproporcional, completamente estilisticamente inconsistente.
- Gráficos tem aparência amadora, design visual muito pobre - desproporcional, completamente estilisticamente inconsistente.

VI) Apelo visual: Quão boa é a aparência do aplicativo?

- Muito atraente, memorável, se destaca; o uso de cores aprimora recursos/menus do aplicativo.

- Alto nível de apelo visual - gráficos perfeitos - consistente e profissionalmente projetado.
- Algum apelo visual - médio, nem agradável, nem desagradável.
- Pouco apelo visual - mal projetado, mau uso da cor, visualmente chato.
- Sem apelo visual, desagradável para olhar, mal projetado, confronto/cores incompatíveis.

SEÇÃO 02

Avaliação das estratégias empregadas no ARtEMIS

I) Quão importante você achou a inclusão da gamificação (inserção dos elementos de jogo fases, pontos e conquistas) no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores?

1 2 3 4 5
 Não importante Extremamente importante

II) Quão importante você achou a inclusão da Realidade Aumentada no ARtEMIS para estimular seu uso e o aprendizado de Organização e Arquitetura de Computadores?

1 2 3 4 5
 Não importante Extremamente importante

III) Quão útil você achou o ARtEMIS para aprender sobre Organização e Arquitetura de Computadores?

1 2 3 4 5
 Inútil Extremamente útil

IV) O reconhecimento das figuras foi preciso?

1 2 3 4 5
 Impreciso Extremamente preciso

V) O reconhecimento das figuras foi rápido?

1 2 3 4 5
 Extremamente lento Extremamente rápido

