

Proposta de Ensino de Arquitetura de Computadores com Gamificação e Realidade Aumentada

Geofrangite Câmara da Silva
Programa de Pós-graduação em Ciência
da Computação
UFERSA_UERN
Mossoró, Brasil
gcscamara@gmail.com

Leiva Casemiro Oliveira
Departamento de Computação
Universidade Federal Rural do Semi
Árido (UFERSA)
Mossoró, Brasil
leiva.casemiro@ufersa.edu.br

Sílvio Roberto Fernandes
Departamento de Computação
Universidade Federal Rural do Semi
Árido (UFERSA)
Mossoró, Brasil
silvio@ufersa.edu.br

Resumo—Neste artigo é apresentado um simulador que faz uso de Realidade Aumentada com uma abordagem de gamificação para o ensino/aprendizagem de Organização e Arquitetura de Computadores. O objetivo dessa proposta é permitir maior integração entre o principal livro usado nesta disciplina, “Organização e Projeto de Computadores” de Patterson e Hennessy, e o simulador proposto. Para isso, tal simulador reconhece as figuras do livro e cria modelos 3D sobre elas, por meio da realidade aumentada, com os quais o aluno pode interagir obtendo informações e visualizando o funcionamento interno do processador MIPS. As figuras apresentadas no livro seguem uma ordem na qual refinam o nível de detalhamento, assim o simulador proposto utiliza os aspectos de gamificação para usar tais figuras como fases de um jogo. A ferramenta proposta ainda inclui avaliação do conhecimento de cada fase, de modo que essa avaliação é responsável por determinar se o usuário poderá avançar para a próxima fase.

Palavras-chave—Realidade Aumentada, Gamificação, Organização e Arquitetura de Computadores, MIPS.

I. INTRODUÇÃO

Disciplinas que abordam os conceitos de Organização e Arquitetura de Computadores (OAC) são tidas como complexas em termos de aprendizado devido seus conceitos terem baixo nível de abstração. Uma forma de tentar equilibrar o aumento da abstração sem perder essencialmente os conceitos e detalhes cruciais dessa disciplina, tem sido a adoção de simuladores [1 - 4].

Simuladores 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 [5]. No contexto do ensino de OAC, muitos simuladores foram e continuam sendo desenvolvidos. No entanto, a não integração desses simuladores com um material de estudo básico de disciplina (livro didático), de forma que o aluno não precise se “desligar” de um para utilizar o outro, diminui a efetividade desses *softwares*.

Outro fator que tem se destacado nas práticas educativas são as intervenções de gamificação. Através do uso de pensamentos, abordagens e elementos de jogos aplicados no contexto educacional a gamificação permite o aumento da motivação dos alunos por engajamento nas atividades da disciplina, obtidos com o uso de intervenções individuais ou em grupo, acabando com o absentismo causador dos baixos índices de desempenho que tais disciplinas costumemente apresentam [6].

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, uma das referências mais utilizadas no ensino de OAC é o livro “Organização e Projeto de Computadores” [7]. Esse livro aborda um subconjunto de instruções do processador MIPS e apresenta seu caminho de dados de maneira incremental, por meio de figuras.

Diante do exposto, neste artigo é apresentado um simulador, chamado de ARtEMIS (*Augmented Reality MIPS Simulator*), pensado para ser uma ferramenta integrada ao livro “Organização e Projeto de Computadores” [7] para auxiliar no aprendizado dos conteúdos nele abordados. Como o livro apresenta o caminho de dados do MIPS de forma incremental, por meio de figuras, então, a ideia com o simulador é “dar vida” a essas figuras para facilitar a assimilação dos conceitos apresentados no livro. Para alcançar a integração recorreu-se a tecnologia de Realidade Aumentada (RA) que possibilita o reconhecimento das figuras do livro, recriando-as como modelos interativos com as quais o aluno pode interagir obtendo informações e visualizando o funcionamento interno do MIPS, podendo avaliar também os seus conhecimentos. Com isso, o aluno tem um simulador mais fiel ao livro evitando dificuldades de entendimento dos conteúdos nele abordados devido diferenças entre a sua representação do caminho de dados e a representação no simulador. Além disso, a abordagem da gamificação ocorre de modo que o reconhecimento das figuras obedece a uma ordem de progressão em que a avaliação do conhecimento nas figuras anteriores deve ter sido satisfatória - em um processo de verificação de status, para que a próxima figura (o próximo nível de jogo), seja liberada como elemento de recompensa.

O restante do trabalho está organizado da seguinte forma: na seção II são apresentados os conceitos de RA e de gamificação; a seção III apresenta os trabalhos relacionados; a seção IV dá uma visão geral do ARtEMIS e descreve o processo de implementação do mesmo; a seção V descreve o funcionamento do ARtEMIS; e a seção VI apresenta as conclusões e trabalhos futuros.

II. CONCEITOS BÁSICOS

Nessa seção são apresentados os conceitos de RA e gamificação com destaque para os benefícios de aplicá-los no campo educacional.

A. Realidade Aumentada

A Realidade Aumentada (RA) consiste na sobreposição em tempo real de objetos virtuais ao espaço físico do usuário por meio do uso de um dispositivo que permite a manipulação e visualização desses objetos [8]. Ela tem sido aplicada em vários contextos como turismo, treinamentos militares, medicina, entretenimento e educação [9].

O funcionamento da RA requer alguns elementos básicos, os quais consistem em monitor, câmera, *software* e alvos (símbolos ou marcadores, objetos 3D, geolocalização). Esses elementos são mostrados na Figura 1.

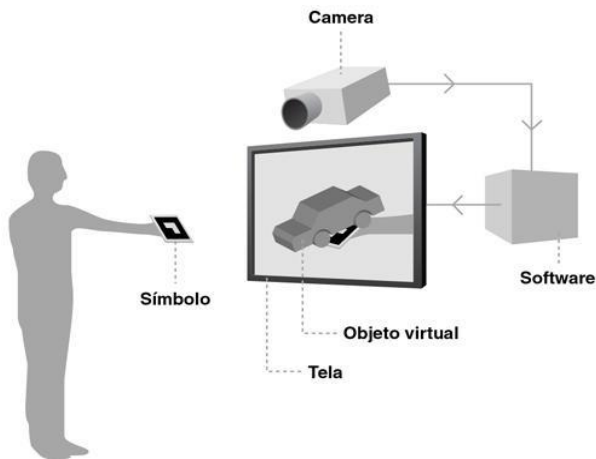


Figura 1. Esquema de funcionamento da RA. Fonte: [10]

Como se pode ver na Figura 1, o símbolo é o elemento no qual os objetos virtuais são sobrepostos. Para isso, a câmera captura a imagem do símbolo e a envia para um *software* que realiza o processamento necessário e depois envia para o monitor a imagem com os objetos virtuais sobrepostos ao símbolo.

Nos últimos anos, houve um aumento significativo da 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 além da redução do custo do material de laboratório [11].

B. Gamificação

Gamificação (do original em inglês *gamification*) é um conceito introduzido em 2002 por Nick Pelling e se popularizou em 2010, e se refere ao emprego de elementos de jogo em ambientes não jogáveis. Seu principal objetivo é aumentar o engajamento e a motivação das pessoas na resolução de problemas e no desempenho de tarefas [12]. A gamificação como uso de mecanismos de jogos com o objetivo de resolver problemas práticos ou de estimular engajamento entre um público específico tem sido abordado na literatura [12].

Os mecanismos de jogos tipicamente ligados à gamificação são pontos, crachás, tabelas de liderança, progressão, *status*, níveis, recompensas e papéis [13]. Uma breve definição desses mecanismos é apresentada na Tabela 1.

TABELA I. 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.
Status	Alcunhas que indicam progresso.
Níveis	Ambientes com dificuldade crescente.
Recompensas	Itens tangíveis e desejáveis.
Papéis	Personagens de jogo.

A gamificação tem sido aplicada em vários campos como: educação e interação homem-máquina [13], *marketing* e saúde [14], entretenimento e negócios [15], entre outros. Para os autores de [16] a aplicação da gamificação no campo educacional é capaz de modernizar a metodologia de ensino para adaptá-la 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.

Em [17] os autores conduziram uma revisão sistemática sobre a aplicação da gamificação no nível superior. Seus achados revelam o crescimento dessa abordagem 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.

A combinação das vantagens presentes em RA e em gamificação se credencia como promissora na atual geração digital, na medida em que as necessidades dos alunos e os objetivos educacionais das disciplinas encontram-se atendidos.

III. TRABALHOS RELACIONADOS

Na literatura são encontrados diversas proposta de simuladores para o ensino de OAC. Uma das mais clássicas arquiteturas usadas no ensino de OAC é a do processador MIPS I. A seguir são apresentados alguns que, assim como o proposto nesse artigo, trabalham com o caminho de dados dessa versão do MIPS, provendo uma visualização das sequências de interações entre suas unidades funcionais. Também são mostradas algumas arquiteturas próprias e uma abordagem RA para o ensino de OAC.

O MARS [18] (*Mips Assembly and Runtime Simulator*) simula parte do conjunto de instruções do MIPS32, ou seja, o conjunto de instruções de 32 bits publicada oficialmente pela MIPS Technologies. O MARS disponibiliza um editor de texto e um montador MIPS o que permite ao usuário fornecer um código *assembly* e simular sua execução. Durante a simulação o usuário pode controlá-la, alterando a velocidade de execução das instruções e também pode visualizar/modificar os valores dos registradores e da memória de dados e visualizar o estado da memória de instruções. Por meio da extensão criada em [19] disponível para as versões a partir da 4.5, o usuário pode visualizar o

caminho de dados e acompanhar, por meio da alteração das cores dos barramentos, a sequência em que os dados são transferidos entre os componentes. No entanto, essa extensão não exibe os valores desses dados.

O WEBMIPS [20] é um ambiente Web que tem como objetivo a simulação da arquitetura MIPS com *pipeline* para parte do conjunto de instruções do MIPS. Assim como o MARS, ele disponibiliza um editor de texto e um montador MIPS e o usuário pode controlar a simulação, escolhendo entre executar o código etapa por etapa, onde 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. O usuário pode modificar os valores dos registradores e da memória de dados, e na memória de instruções as instruções em execução são destacadas de acordo com o estágio do *pipeline* em que se encontram. 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 de *clock* é exibido.

O DRMIPS [21] é um simulador que permite tanto a simulação do MIPS monociclo quanto *pipeline*. Como o MARS e o WEBMIPS ele contém um editor de texto e um montador MIPS. Após a montagem do código o usuário pode modificar os valores da memória de dados e dos registradores e então, realizar a simulação sendo possível executar todas as instruções de uma vez ou uma por uma. O simulador também mostra o caminho de dados graficamente onde é possível visualizar os valores contidos nos barramentos dos caminhos de dados e de controle. Além da versão multiplataforma para PCs ele apresenta uma versão para dispositivos móveis com Sistema Operacional (SO) Android. Contudo, os recursos de visualização e usabilidade tornam o uso desse simulador desmotivante.

O SIMUS [2] simula a arquitetura do processador hipotético SAPIENS [2]. A proposta do simulador SIMUS é apresentar uma arquitetura que possa ser explorada em diferentes níveis de complexidade, com o emprego de exemplos mais simples, para os cursos iniciais, até exemplos mais complexos, para os cursos mais avançados, sempre com o uso da mesma ferramenta. O simulador apresenta um IDE (*Integrated Development Environment*) onde o aluno pode editar, compilar, depurar e executar código de programas escritos na linguagem de montagem do processador Sapiens.

A família de processadores BIP (*Basic Instruction-set Processor*) [22] juntamente com o IDE BIPIDE [22], foram criados com o objetivo de auxiliar aprendizado de conceitos de OAC por alunos iniciantes na área de Computação. A ideia é que cada simulador da família possa ser aplicado para auxiliar no aprendizado do conteúdo de uma disciplina promovendo a interdisciplinaridade. Os autores sugerem como essa plataforma pode ser usada nas disciplinas

Algoritmos e Programação, Circuitos Digitais, Compiladores e OAC.

Em [23] a RA foi empregada para criar um objeto de aprendizagem (OA) para o ensino da arquitetura do processador Neander. 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. Contudo os autores não mencionam se no OA criado o usuário pode inserir código *assembly* para visualizar sua execução.

Embora os simuladores voltados para o ensino da arquitetura do MIPS proporcionem uma noção visual do seu funcionamento interno, a falta de integração deles com um material de apoio, em termos de notação, terminologia, organização e apresentação dos conteúdos, deixa-os limitados como ferramentas didáticas.

IV. O SIMULADOR ARtEMIS

A. Visão Geral

O simulador proposto foi inicialmente denominado ARMS (*Augmented Reality MIPS Simulator*) [24], uma vez que foi idealizado para ser usado integrado ao livro [7] usando RA e gamificação. No entanto, devido a semelhança sonora com a arquitetura ARM decidiu-se “rebatizá-lo” com o nome ARtEMIS (*Aumenged REality MIPS Simulator*). Na mitologia grega Ártemis é uma deusa ligada à vida selvagem e à caça e também associada à lua e à magia¹.

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. A atual versão o ARtEMIS suporta o MIPS monociclo. 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 criar uma RA que usa modelos 3D dos caminhos de dados com os quais é possível interagir.

B. Implementação

A implementação do ARtEMIS é realizada usando as ferramentas Unity 2017.2.0², Vuforia SDK (*Software Development Kit*) para Unity³, Tinkercad⁴, Android SDK⁵ e a linguagem C# [25]. Para escolher essas ferramentas foi feito um levantamento.

Para desenvolvimento de RA foram encontradas várias ferramentas como: Vuforia, Wikitude⁶, EasyAR⁷, ARKit⁸, ARCore⁹ e ARToolKitX¹⁰. Para escolher 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 e documentação disponibilizada. A que

¹ <https://pt.wikipedia.org/wiki/%C3%81rtemis>

² <https://unity3d.com/pt/>

³ <https://www.vuforia.com/>

⁴ <https://www.tinkercad.com/>

⁵ <https://developer.android.com/>

⁶ <https://www.wikitude.com/>

⁷ <https://www.easyar.com/>

⁸ <https://developer.apple.com/arkit/>

⁹ <https://developers.google.com/ar/>

¹⁰ <http://www.artoolkitx.org/docs/about/>

atendeu nossas restrições e necessidades de maneira mais satisfatórias foi Vuforia, e por isso foi escolhida. Para desenvolver para a plataforma Android o Vuforia dá as opções de trabalhar com o Android Studio ou com o Unity. Inicialmente foi escolhido o Android Studio, no entanto foi encontrada pouca documentação para auxiliar na implementação. Então optou-se por trabalhar com o Unity, com mais documentação disponível.

Para a modelagem 3D também existem muitas ferramentas como 3ds Max¹¹, Blender¹² e Maya¹³. No entanto, 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 o processo de modelagem 3D é simples.

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. Vuforia é um SDK para desenvolvimento de aplicações RA voltadas para as plataformas Android, iOS, UWP e óculos digitais. Ela pode ser integrada ao Android Studio, Xcode, Visual Studio e Unity. Tinkercad é 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. 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.

O desenvolvimento do ARtEMIS seguiu o fluxo apresentado na Figura 2.

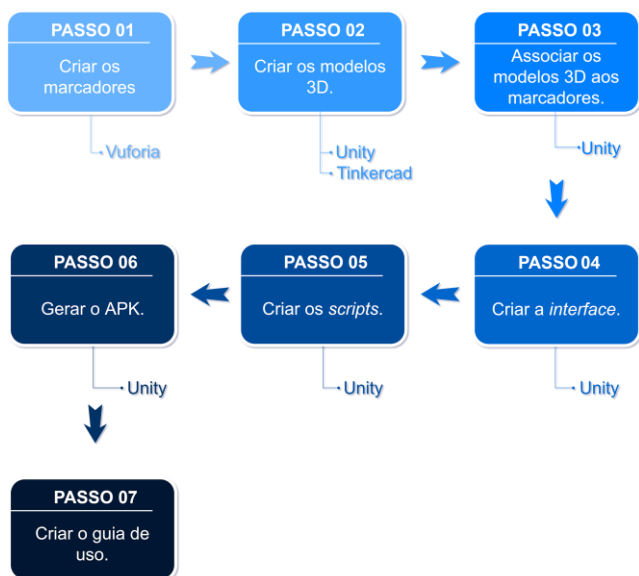


Figura 2. Fluxo de desenvolvimento do ARtEMIS.

Um dos passos da implementação é transformar as figuras do livro em marcadores (passo 01). A transformação é feita convertendo-as 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 indicar 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.

O passo 02 é criar os modelos 3D dos componentes do processador. Embora os modelos 3D de algumas formas geométricas como cubo, esferas, cilindros e cápsulas possam ser criados no próprio Unity, ele não oferece meios para transformá-los em modelos mais elaborados (por exemplo, o modelo 3D de uma porta AND). Assim, o Tinkercad é utilizado para modelagem de todos os componentes que não podem ser representados pelas formas disponíveis no Unity.

Com os componentes modelados e as figuras transformadas em marcadores, o passo seguinte (passo 03) é criar o modelo 3D de cada figura associando os modelos dos componentes aos marcadores. 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 próximo passo (passo 06) é gerar o APK. 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 APK. Feito isso o APK é gerado e está pronto para ser instalado.

Por fim é criado o Guia de Uso (passo 07), um documento que contém as 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.

V. PROPOSTA DE GAMIFICAÇÃO COM ARtEMIS

Até o momento, o ARtEMIS suporta 4 figuras do livro “Organização e Projeto de Computadores” de Patterson e Hennessy [7], 3a. edição em português. Nesta edição do livro as figuras são respectivamente numeradas como 5.1, 5.2, 5.17 e 5.24, as quais são apresentadas na Figura 3. As figuras do livro serão chamadas nesse texto de “ilustração” para não confundir com as próprias figuras do artigo. As ilustrações, são apresentadas em ordem incremental de detalhamento do caminho de dados, inclusive aumentando o suporte de mais instruções.

O ARtEMIS foi originalmente pensado para ser usado como uma ferramenta auxiliar no ensino/aprendizado do MIPS permitindo o usuário interagir com qualquer uma das quatro ilustrações em qualquer ordem [24]. Entretanto, aproveitando a apresentação didática das ilustrações no livro e o conceito de fases empregados na gamificação, neste artigo propomos uma versão gamificada do ARtEMIS.

¹¹ <https://www.autodesk.com.br/products/3ds-max/overview>

¹² <https://www.blender.org/>

¹³ <https://www.autodesk.com.br/products/maya/overview>

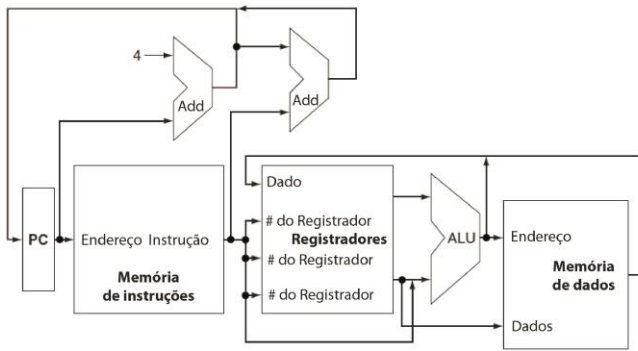


Ilustração 5.1

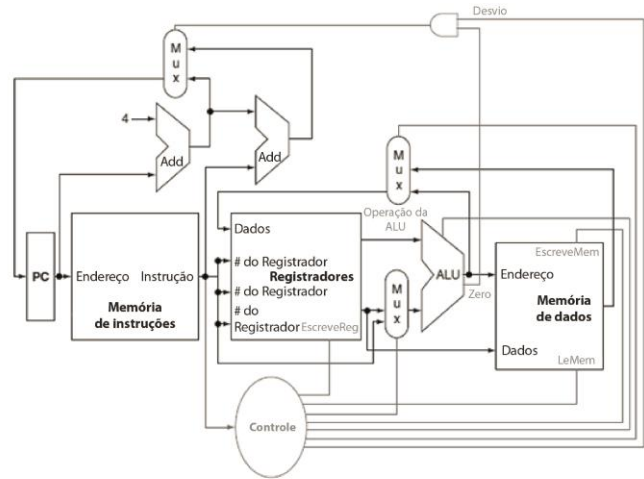


Ilustração 5.2

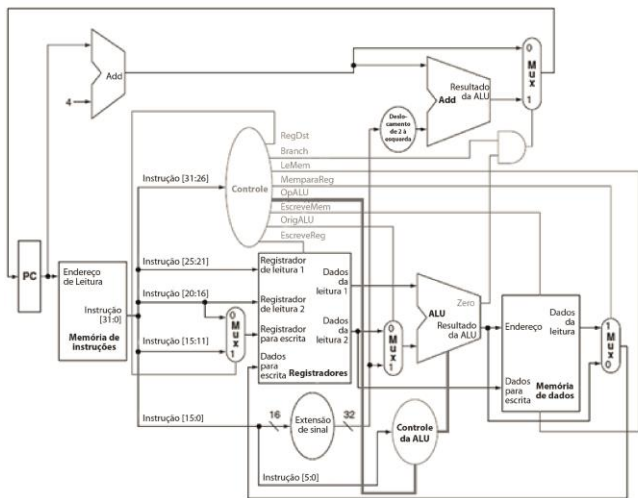


Ilustração 5.17

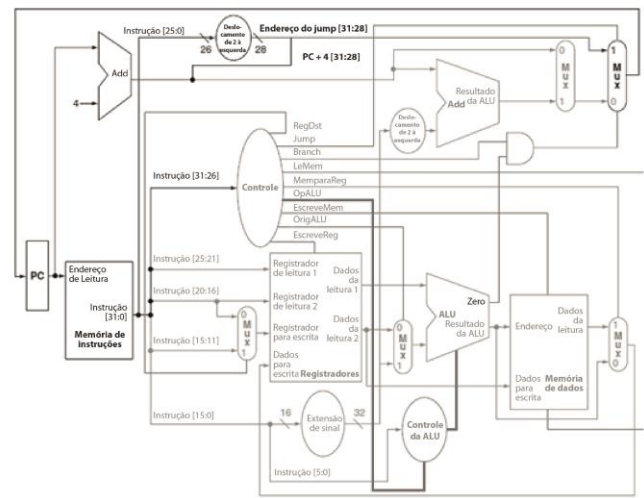


Ilustração 5.24

Figura 3. Ilustrações 5.1, 5.2, 5.17 e 5.24 do livro texto [7] utilizadas como marcadores pela RA do ARtEMIS.

Nesta versão gamificada cada ilustração aparece como uma fase e entre duas fases há sempre um conjunto de exercícios sobre a ilustração anterior, assim como mostra a Figura 4. Para ter acesso a uma fase é necessário que o usuário tenha passado pela fase e o exercício anterior, atingindo pelo menos a pontuação mínima. As funcionalidades nas fases são acumulativas, ou seja, as opções adquiridas em uma fase são mantidas na fase seguinte com o acréscimo de outras funcionalidades.

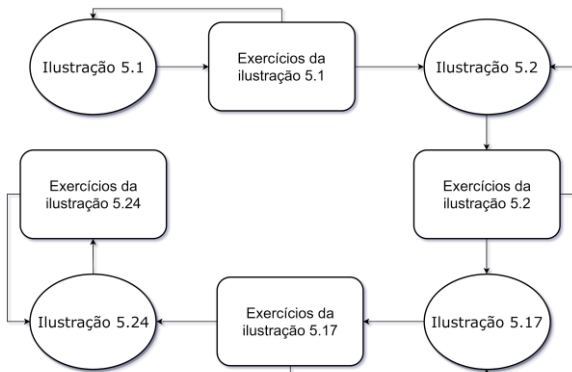


Figura 4. Fases do ARtEMIS gamificado.

O ARtEMIS permite navegar livremente pelas fases desbloqueadas, bem como o reconhecimento automático de uma ilustração (das fases desbloqueadas). Também é possível ativar o *flash* do dispositivo e evitar problemas de reconhecimento devido iluminação. Quando a ilustração é reconhecida o modelo da arquitetura é gerado sobre ela juntamente com os elementos GUI que permitem configurar e interagir com ela, bem como a visualização das informações relacionadas a ilustração. Caso a ilustração não seja reconhecida, devido problemas na qualidade da impressão do livro, o simulador permite o carregamento automático dos modelos 3D após um tempo (10 segundos) tentando fazer o reconhecimento. Também é possível fixar a imagem reconhecida para evitar ter que ficar apontando o dispositivo móvel para a ilustração do livro durante todo o tempo da realização das atividades. Quando o usuário desejar retornar à tela inicial poderá clicar no botão “Voltar”.

Inicialmente o usuário tem apenas a Fase 1 disponível, como mostra a Figura 5(a). O propósito dessa fase é exibir informações das funcionalidades dos componentes básicos do processador. A Fase 2 permite aprender sobre o

funcionamento das instruções do formato R e formato I. A Fase 3 é usada para o aprendizado do desvio condicional BEQ. E a Fase 4 acrescenta a instrução JUMP. As interações em cada fase permitem obtenção de informações e visualização de simulações. Para avançar entre as fases é necessário atingir a pontuação mínima nos exercícios daquela fase. A Figura 5(b) ilustra o momento em que todas as fases estão disponíveis.

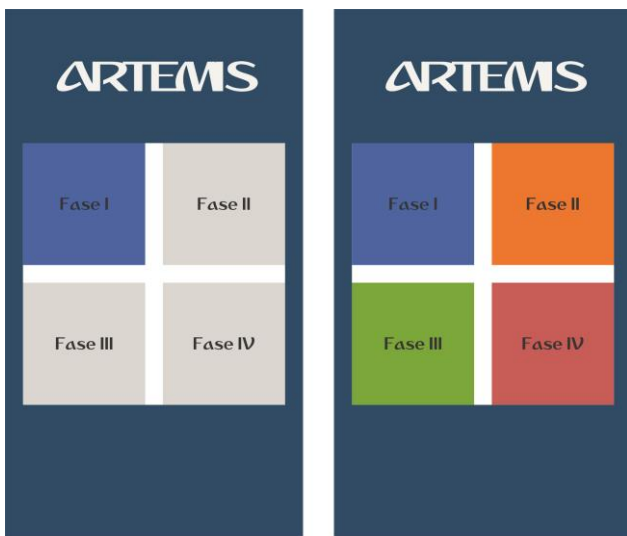


Figura 5. Tela inicial do ARTEMIS com gamificação. a) Apenas a Fase 1 está disponível. b) Todas as fases estão liberadas.

A Fase 1 reconhece a ilustração 5.1, cujo propósito é apresentar as unidades funcionais básicas (contador de programas-PC, memória de instruções, banco de registradores, somadores, unidade lógica aritmética-ULA e memória de dados) do processador MIPS. Essa fase possibilita ao usuário obter informações sobre essas unidades clicando sobre o modelo 3D gerado de cada uma delas. Quando clicada, a unidade é destacada e informações textuais sobre sua função são exibidas em um elemento de visualização do tipo *pop-up*, como mostrado na Figura 6. Para que o *pop-up* desapareça basta clicar novamente na unidade ou em clicar em uma outra unidade.

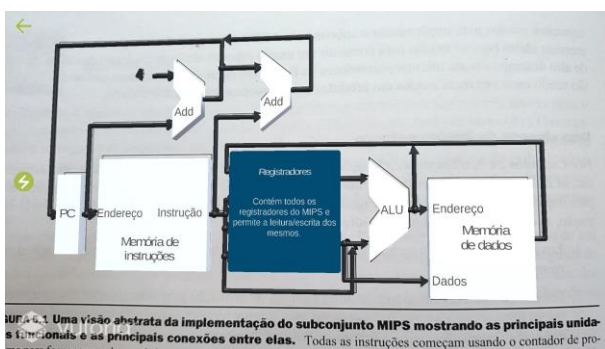


Figura 6. Reconhecimento da ilustração 5.1 pelo ARTEMIS. Detalhe para o *pop-up* com informações sobre a unidade banco de registradores.

Após interagir com todas as unidades funcionais da Fase 1, o usuário passa para os exercícios. Compostos por 5 questões, cada questão apresenta uma afirmação sobre uma das unidades funcionais básicas do processador MIPS da fase 1, e o usuário deve julgar se a afirmação é verdadeira ou falsa, conforme a Figura 7. Ao concluir os exercícios é exibido um gabarito das questões, destacando em verde aquelas respondidas corretamente e em vermelho as respondidas

erradas, além de exibir o quantitativo dos acertos, como ilustra a Figura 8.

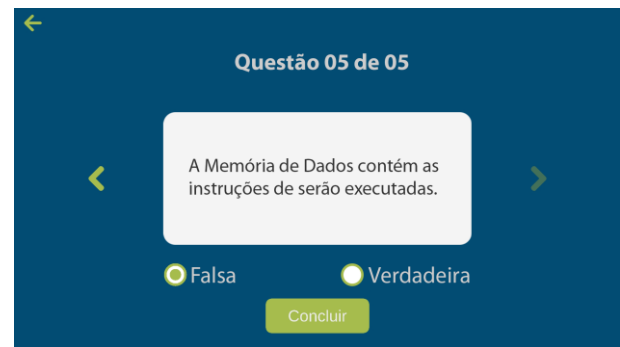


Figura 7. Exemplo de uma das questões dos exercícios da Fase 1.

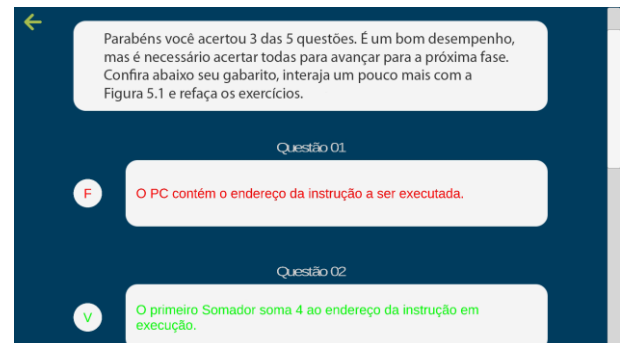


Figura 8. Tela mostrando o gabarito dos exercícios

Desse modo, quando a ilustração é reconhecida, o usuário deve selecionar na memória de instruções um tipo de instrução para visualizar a execução (*add*, *load*, *store*). Quando o tipo é selecionado é exibida uma tela sobre as unidades funcionais (Figura 9(a)) que permite selecionar os registradores envolvidos na execução dela, definir seus respectivos valores e os valores da memória de dados e outros valores (*offset* e *immediate*) quando necessários. 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 9(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 9(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 nessa 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 sendo que barramentos que tem os mesmos dados são coloridos com a mesma cor e os que apresentam valores 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.

A Figura 9 exemplifica o funcionamento do reconhecimento da ilustração 5.2 mostrando a simulação da execução de uma instrução do tipo *add*. A Figura 9(a) mostra a tela que surge sobre os componentes quando o tipo *add* é selecionado. A Figura 9(b) mostra a configuração da instrução onde o registrador *t1* foi selecionado como o *rs* e recebeu o valor 4, *t2* foi selecionado como *rt* e recebeu o valor 5 e *t0* foi selecionado como *rd*. Na Figura 9(c) é mostrada a primeira etapa da execução da instrução onde o PC repassa o endereço da instrução para a Memória de Instruções e para o Somador. Na Figura 9(d) o Somador soma 4 ao endereço vindo do PC e a instrução é decodificada. Na Figura 9(e) os valores contidos em *t1* e *t2* são somados pela ULA. E por fim, na Figura 9(f) o resultado da ULA é escrito em *t0* e o PC é incrementado em 4.

Após visualizar a execução dos tipos de instruções disponíveis o usuário tem o acesso liberado aos exercícios da Fase 2. Os exercícios são compostos por 3 questões de verdadeiro ou falso.

Nas Fases 3 (ilustrações 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. Contudo, diferentemente da Fase 2, onde 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 partir da Fase 3, a aba “Código” torna-se disponível, ver Figura 10(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 valores (*offset* e *immediate*) envolvidos na execução da instrução (Figura 10(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. Se o código inserido estiver correto, 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” que permite ao usuário visualizar as instruções em linguagem de máquina binária gerada a partir das instruções do código *assembly* inserido.

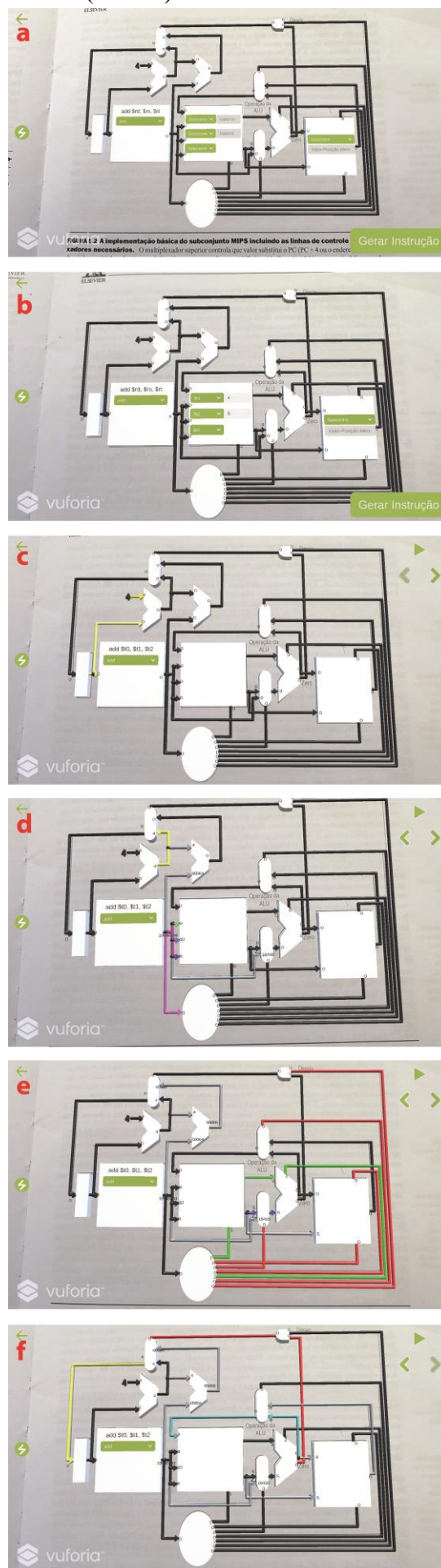


Figura 9. Funcionamento do reconhecimento da ilustração 5.2 mostrando a simulação da execução de uma instrução do tipo *add*.

soluções de simulação desse processador, entretanto elas não apresentam integração com tal livro.

O ARtEMIS foi desenvolvido para dispositivos móveis com sistema operacional Android, e permite reconhecer as figuras do livro, 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 possibilidade a execução de instruções e obtenção de informações sobre o funcionamento das unidades funcionais do processador MIPS.

Assim como o livro, que apresenta as figuras da arquitetura do MIPS de forma incremental, cada vez com mais detalhes e compatibilidade com mais instruções, esse esquema de níveis faz com que o reconhecimento das figuras seja feito em sequência. Dessa forma, o ARtEMIS cria um fluxo para sua utilização como recurso didático, de modo a implementar uma abordagem de gamificação. Nessa abordagem o reconhecimento de cada figura do livro funciona como uma fase, de modo que os recursos mais avançados serão disponibilizados a medida que as etapas anteriores são cumpridas pelo usuário.

Em cada fase, o aluno é levado a interagir com os recursos disponibilizados pelo ARtEMIS com algum propósito didático e em seguida realiza exercícios sobre o que aprendeu. Os exercícios contabilizam a quantidade de acertos, e em caso de atingir a pontuação mínima requerida, o usuário/aluno pode avançar para a próxima fase, caso contrário as figuras das fases seguintes não são reconhecidas, necessitando jogar a fase anterior novamente.

Desse modo, a ferramenta extrapola o que conceitualmente se define como simulador visto que a plena integração ao material, atrelada à RA gamificada, introduz uma nova metodologia de ensino e aprendizagem para a disciplina de OAC. Novas formas de interação com o material, autonomia no processo de aprendizagem, possibilidades de construção de novos conteúdos, novas formas de avaliação e acompanhamento do aprendiz são factíveis com o ARtEMIS.

Dentre os trabalhos futuros para melhorar o ARtEMIS lista-se:

- Extensão do caminho de dados/controle das figuras atuais, de modo que possam “executar” outras instruções, tais como as instruções *jr* (*jump register*) e *lui* (*load upper immediate*), uma vez que tais instruções são suportadas pelo MIPS e não estão explicitadas nas figuras do livro. Dessa forma o ARtEMIS pode se tornar um recurso adicional ao livro.
- Reconhecimento de mais das figuras do livro relativas os caminhos de dados multiciclo e *pipeline*. Bem como o reconhecimento das figuras na versão em inglês do livro.
- 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 uma ferramenta para o professor criar seus próprios exercícios, definindo o grau de dificuldade e percentual de acerto para conclusão de uma fase, além da organização por turmas/grupos de alunos.
- Incluir na gamificação do ARtEMIS um mecanismo de *ranking*, como forma de intensificar a motivação.

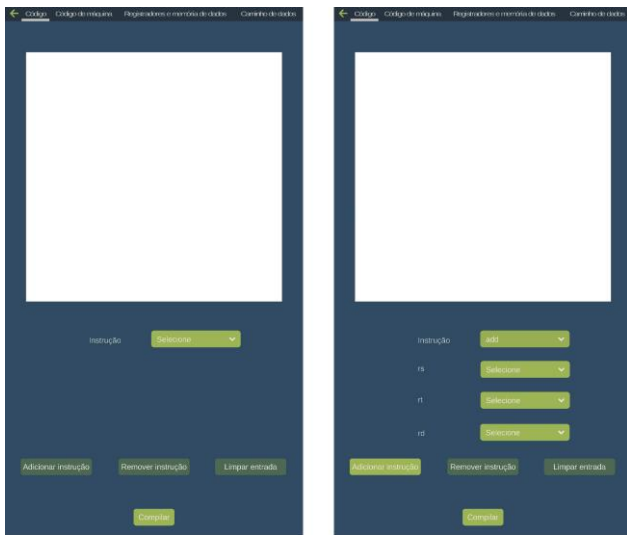


Figura 10. Aba para inserir o código de máquina para o ARtEMIS.

A aba “Registadores e memória de dados” (Figura 11), 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. 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, exibindo a instrução atual em execução.



Figura 11. Tela para visualizar/alterar os valores da memória de dados e dos registradores.

Para as Fases 3 e 4 os exercícios propostos focam na interpretação do caminho de dados, de modo que aluno consiga diferenciar os aspectos envolvidos na execução de diferentes tipos instruções. Também existem 2 exercícios de múltipla escolha em cada uma das fases semelhantes ao da Fase 2. A taxa de acerto necessárias para avanço dessas fases é maior que 50%.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou uma proposta de ferramenta didática que faz uso de RA e gamificação, denominada ARtEMIS (*Augmented Reality MIPS Simulator*), para ser usada em conjunto com o livro “Organização e Projeto de Computadores” de Patterson e Hennessy, 3ª edição, em português. Esse é o livro mais utilizado no ensino de OAC e o processador MIPS e por isso já foram criadas diversas

- 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.

Ao fim da implementação completa do simulador será realizada a validação. Um levantamento feito pelos autores desse artigo permitiu identificar as metodologias utilizadas para avaliar aplicações RA. Verificou-se que geralmente os participantes da pesquisa são agrupados em grupo de teste (que usa a aplicação RA) e o grupo de controle (que é submetido ao método tradicional de ensino). Também constatou-se que a forma mais empregada para verificar o impacto da aplicação da RA no desempenho dos alunos é a aplicação de Pré-Teste (antes de participar da pesquisa) e Pós-Teste (após participar da pesquisa) e que geralmente esses testes são testes escritos. Dessa forma, a validação do simulador será feita por meio de uma metodologia baseada nos resultados desse levantamento. Os participantes serão alunos da disciplina de Arquitetura de Computadores do curso de Ciência da Computação da Universidade Federal Rural do Semi-Árido (UFERSA). Aos alunos que utilizarem o simulador também será aplicado um questionário para avaliação da ferramenta.

VII. AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

- [1] F. A. M. Alves, D. Almeida, L. Bragança, A. B. M. Gomes, R. S. Ferreira and J. A. M. Nacif, "Ensinando Arquiteturas Vetoriais Utilizando um Simulador de Instruções MIPS", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 4, nº 1, pp. 9-12, 2015.
- [2] G. P. Silva and J. A. S. Borges, "SimuS Um Simulador Para o Ensino de Arquitetura de Computadores", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 5, nº 1, pp. 7-12, 2016.
- [3] L. Duenha and R. Azevedo, "Utilização dos simuladores do MPSoCBench para o ensino e aprendizagem de Arquitetura de Computadores", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 5, nº 1, pp. 26-31, 2016.
- [4] A. S. Roque, G. Schievelbein, M. Losekann and D. R. Silva, "AD3W: Um simulador educacional para análise de dependências de dados em nível de instrução", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 2, nº 1, pp. 41-44, 2013.
- [5] G. A. Esmeraldo and E. B. Lisboa, "Uma Ferramenta para Exploração do Ensino de Organização e Arquitetura de Computadores", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 6, nº 1, pp. 68-75, 2017.
- [6] S. R. Fernandes and I. S. Silva, "Relato de Experiência Interdisciplinar Usando MIPS", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 6, nº 1, pp. 52-61, 2017.
- [7] D. Patterson and J. Hennessy, *Organização e projeto de computadores - a interface hardware software.*, 3th ed. Campus, 2005.
- [8] C. Kirner and R. A. Siscoutto, "Fundamentos de Realidade Virtual e Aumentada", in *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*, 2007.
- [9] R. V. Krevelen and R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations", in *The International Journal of Virtual Reality*, vol. 9, nº 2, pp. 1-20, 2010.
- [10] C. P. Bianchini and L. Silva, "Sistemas de Realidade Aumentada Móvel Suportados por Computação em Nuvem", in *Tendências e Técnicas em Realidade Virtual e Aumentada*, vol. 4, pp. 9-32, 2014.
- [11] M. Akçayir and G. Akçaayir, "Advantages and challenges associated with augmented reality for education: A systematic review of the literature" in *Educational Research Review*, vol. 20, pp. 1-11, 2017.
- [12] Y. Vianna, M. Vianna, B. Medina and S. Tanaka, *Gamification, Inc: Como reinventar empresas a partir de jogos*, 1st ed. MJV Press, 2013.
- [13] K. Seaborn and D. Fels, "Gamification in Theory and Action: A Survey", in *International Journal of Human-Computer Studies*, vol. 74, pp. 14-31, 2015.
- [14] J. Lee and J. Hammer, "Gamification in education: What, how, why bother?", in *Academic Exchange Quarterly*, vol. 15, nº 2, pp. 146-151, 2011.
- [15] R. Hervas, D. Ruiz-Carrasco, T. Mondejar and J. Bravo, "Gamification mechanics for behavioral change: A systematic review and proposed taxonomy", in *11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pp. 395-404, 2017.
- [16] G. Signori and J. Guimarães, "Gamificação como Método de Ensino Inovador", in *Journal International Active Learning*, vol 1, nº 1, pp. 66-77, 2016.
- [17] S. Subhash and E. A. Cudney, "Gamified learning in higher education: A systematic review of the literature", in *Computers In Human Behavior*, vol. 87, pp. 192-206, 2018.
- [18] K. Vollmar, P. Sanderson, "MARS: an education-oriented MIPS assembly language simulator", in *37th Sigcse Technical Symposium On Computer Science Education - Sigcse '06*, pp.239-243, 2006.
- [19] M. R. D. Araújo, F. L. C. Pádua e F. L. Corrêa Junior (2014), "MIPS X-Ray: A MARS Simulator Plug-in for Teaching Computer Architecture", in *Recent Contributions from Engineering, Science & IT (iJES)*, vol. 2, nº 2, pp. 36-42, 2014.
- [20] I. Branovic, R. Giorgi, and E. Matinelli, "WebMIPS: A New Web-Based MIPS Simulation Environment for Computer Architecture Education", in *31st Annual International Symposium on Computer Architecture*, 2014.
- [21] B. Nova, J. C. Ferreira, and A. Araújo, "Tool to Support Computer Architecture Teaching and Learning", in *International Conference of the Portuguese Society for Engineering Education (CISPEE)*, 2013.
- [22] P. V. Vieira, P. R. M. Rech, R. C. Mensch, C. A. Zeferino and A. L. A. Raabe, "Estendendo a Arquitetura dos Processadores BIP para Ampliar o Seu Potencial de Uso em Disciplinas de Introdução a Programação", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 1, nº 1, pp. 1-10, 2012.
- [23] L. M. L. Brum, L. B. Pinho and S. S. Camargo, "Avaliação do Uso de Realidade Aumentada no Ensino de Arquitetura e Organização de Computadores", in *International Journal of Computer Architecture Education (IJCAE)*, vol. 6, nº 1, pp. 10-17, 2017.
- [24] G. C. Silva, L. C. Oliveira and S. R. Fernandes, "Uso de Realidade Aumentada para Ensino de Arquitetura de Computadores com MIPS", in *26º Workshop sobre Educação em Computação (WEI)*, 2018.
- [25] S. Nakov and V. Kolec, *Fundamentals of Computer Programming with C#*, 1st ed. Sofia, 2013.