

**UNIVERSIDADE ESTADUAL DO PIAUÍ-UESPI**  
**CAMPUS PROF. ALEXANDRE ALVES DE OLIVEIRA**  
**CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**DARIELSON ARAÚJO DE SOUZA**

**PROTÓTIPO DE UM JOGO EDUCATIVO PARA O AUXÍLIO DO ENSINO E**  
**APRENDIZAGEM EM SISTEMAS OPERACIONAIS**

**Biblioteca UESPI PMS**  
Registro N° \_\_\_\_\_  
CDD 004.775  
CUTTER S719r  
V \_\_\_\_\_ EX. 01  
Data 28/02/2008  
Voto \_\_\_\_\_

**PARNAÍBA**

**2014**

**DARIELSON ARAÚJO DE SOUZA**

**PROTÓTIPO DE UM JOGO EDUCATIVO PARA O AUXÍLIO DO ENSINO E  
APRENDIZAGEM EM SISTEMAS OPERACIONAIS**

Monografia apresentada ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Piauí – UESPI, Campus Prof. Alexandre Alves de Oliveira, como parte das exigências da disciplina de Estágio Supervisionado, requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof.º Msc. Átila Rabelo Lopes.

**PARNAÍBA**

**2014**

**DARIELSON ARAÚJO DE SOUZA**

**PROTÓTIPO DE UM JOGO EDUCATIVO PARA O AUXÍLIO DO ENSINO E  
APRENDIZAGEM EM SISTEMAS OPERACIONAIS**

Monografia apresentada ao Curso de Bacharelado em  
Ciência da Computação da Universidade Estadual do  
Piauí – UESPI, Campus Prof. Alexandre Alves de  
Oliveira, como parte das exigências da disciplina de  
Estágio Supervisionado, requisito parcial para  
obtenção do título de Bacharel em Ciência da  
Computação.

Orientador: M.Sc. Átila Rabelo Lopes

Monografia Aprovada em: **14 de julho de 2014.**

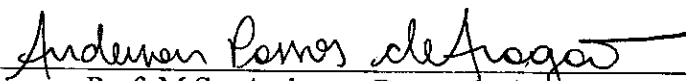
Banca Examinadora:



Prof. M.Sc. Átila Rabelo Lopes  
UESPI/Parnaíba – Orientador



Prof. M.Sc. Sérgio Barros de Sousa  
UESPI/Parnaíba – Avaliador Interno



Prof. M.Sc. Anderson Passos de Aragão  
Faculdade Maurício de Nassau – Avaliador Externo

**“A ciência é a aproximação progressiva do homem com o mundo real”. (MAX PLANCK)**

## **AGRADECIMENTO**

**Agradeço toda minha família, pelo apoio e carinho.**

**Aos meus colegas de turma que conviveram comigo por 4 anos.**

**Ao meu orientador que sempre acreditou em mim, e me incentivou a pesquisas.**

**A todos os professores da UESPI que passaram pela minha turma.**

**A coordenação do curso pelo apoio ao longo dessa jornada.**

**A direção da UESPI que me ajudou diversas vezes.**

**A todos os meus amigos que torcem por mim.**

## RESUMO

Este trabalho propõe-se o desenvolvimento de um jogo educativo para auxiliar o processo de ensino e aprendizagem do tópico memória virtual, da disciplina de sistemas operacionais. O objetivo da ferramenta é oferecer soluções para alguns problemas identificados no ensino de memória virtual, tais como a dificuldade na compreensão de certos conceitos teóricos, técnicas e mecanismos de endereçamento, devido à complexidade do assunto e a escassez de ferramentas para representar, de forma visual e dinâmica, o funcionamento da técnica. Além do mais, através do ambiente estimulante e desafiador dos jogos, busca-se aumentar a motivação dos alunos para o estudo deste assunto. O jogo é do tipo plataforma, executado em um ambiente 2D (duas dimensões), onde o personagem tem como desafio, por exemplo, traduzir um endereço virtual, desviando de alguns obstáculos para encontrar a porta correta, identificada pelo endereço real. A ferramenta foi desenvolvida por meio do *construct 2*. Foram realizados alguns experimentos práticos com alguns alunos do curso de Ciência da Computação, da Universidade Estadual do Piauí. Eles validaram a ferramenta, de acordo com os objetivos propostos.

**PALAVRAS-CHAVE:** Jogo Educativo. Memória Virtual. Ensino-Aprendizagem. Sistemas Operacionais.

## **ABSTRACT**

This work proposes a development of an educational game for the teaching and learning process of the virtual memory, in the field of operating systems. The aim of this tool is to offer solutions for some identified problems in the teaching of virtual memory, like the difficulty in the understanding of some theoretical concepts, techniques, addressing mechanisms, due to the complexity of the subject and the lack of tools to represent, visually and dynamically, virtual memory functions. Moreover, through the stimulating and challenging environment of games, we look forward to improve the motivation of the students to study this subject. The game is a 2D platform game (two-dimensions), where the character has the challenge, for example, of translating a virtual address space, bypassing obstacles to find the right door, identified by the real address. The tool was developed by construct 2. Some practical experiments were done with students of the Computer Science Course, at the State University of Piauí. They validated the tool, and they achieved the proposed goals.

**KEY-WORDS:** Educative game. Virtual Memory. Teaching-Learning. Operational Systems.

## LISTA DE FIGURAS

Figura 1 – Esquema da paginação .....	17
Figura 2 – Exemplo da Tradução do Endereço Lógico para o Endereço Físico.....	18
Figura 3 – Memória Segmentada e Paginada .....	19
Figura 4 – Substituição de Páginas.....	21
Figura 5 – Tela do SOSIM.....	24
Figura 6 – VM Simulator.....	24
Figura 7 – Parte da memória virtual do Sime .....	25
Figura 8 – Simulator Sim Memory.....	25
Figura 9 – Parte da MV paginada do MOSS .....	26
Figura 10 – Diagramas Da UML .....	30
Figura 11 – Desempenho de canvas 2d e WebGL.....	31
Figura 12 – Comparativo da engine construct 2 com outras .....	31
Figura 13 – Diagrama de caso de uso do protótipo .....	36
Figura 14 – Diagrama de sequência.....	36
Figura 15 – Criando o personagem principal .....	40
Figura 16 – Definindo as funções do personagem principal .....	40
Figura 17 – Usando a função espelho nos lados direito e esquerdo .....	42
Figura 18 – Mecanismo de auto-explicação durante o jogo .....	42
Figura 19 – As fases com o botão do tutorial .....	43
Figura 20 – Simulação da MV paginada .....	43
Figura 21 – Introdução do jogo.....	44
Figura 22 – Primeira fase.....	45
Figura 23 – Simulação da MV segmentada.....	46
Figura 24 – Fase 2 da MV segmentada.....	47
Figura 25 – Gráfico da questão sobre a utilidade do jogo .....	49
Figura 26 – Gráfico da questão sobre a organização do jogo .....	49
Figura 27 – Gráfico da questão sobre a interface do jogo .....	50



## **LISTA DE TABELAS**

1 – Softwares Educativos de SO.....	23
2 – Requisitos Funcionais do Protótipo.....	35
3 – Requisitos Não-Funcionais do Protótipo .....	35
4 – Descrições dos Personagens .....	37
5 – Descrições dos Cenários.....	38
6 – Cronograma .....	39

## LISTA DE LISTAGENS

1 – Código em <i>JavaScript</i> da Função <code>keydown</code> .....	41
2 – Código em <i>JavaScript</i> do Efeito Espelho.....	42

## **LISTA DE ABREVIATURAS E SIGLAS**

**BITMAP** – Mapa de Bits

**CPU** – Unidade Central de Processamento

**CSS** – Cascading Style Sheets.

**DOM** – Document Object Model

**DP** – Descritor de processo

**HD** – Hard Disk

**HTML** – Linguagem para Estruturação e Apresentação de Hipertexto.

**MV** – Memória Virtual

**NPV** – Número da Página Virtual

**NUPEC DELTA** – Núcleo de Pesquisa do Delta do Parnaíba

**OMG** – Object Management Group

**RAM** - Memória de Acesso Aleatório

**SE** – Software Educativo

**SO** – Sistemas Operacionais

**UESPI** – Universidade Estadual do Piauí

**UML** – Linguagem Unificada de Modelagem

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>16</b>
2.1 FUNCIONAMENTO DA MEMÓRIA VIRTUAL .....	16
2.1.1 Memória Virtual por Paginação.....	16
2.1.2 Memória Virtual por Segmentação.....	18
2.1.3 Memória Virtual – Paginada e Segmentada .....	19
2.2 PROBLEMAS NO ENSINO E APRENDIZAGEM DA TEMÁTICA.....	19
2.3 A IMPORTANCIA DAS FERRAMENTAS EDUCATIVAS .....	21
2.4 FERRAMENTAS EDUCATIVAS EXISTENTES .....	22
2.5 JOGOS EDUCATIVOS .....	26
2.5.1 A Grande Vantagem dos Jogos Educativos .....	27
<b>3 MÉTODOS E TÉCNICAS</b> .....	<b>28</b>
3.1 TECNOLOGIAS UTILIZADAS PARA A CONSTRUÇÃO DO JOGO .....	28
3.1.1 Game Design .....	28
3.1.2 A UML .....	29
3.1.3 A Engine Construct 2 .....	30
3.1.4 A Linguagem HTML 5.....	32
3.1.5 A Linguagem Javascript e Canvas.....	32
3.1.6 A Ferramenta CSS3.....	33
3.2 PLANEJAMENTO DO PROTÓTIPO .....	33
3.2.1 Público-alvo.....	33
3.2.2 Plataforma .....	33
3.2.3 Utilização do Game Design .....	34
3.2.3.1 A Ideia .....	34
3.2.3.2 Game Design Document.....	34
3.2.3.3 Descrição do Protótipo .....	34
3.2.3.4 Análise de Requisitos .....	34
3.2.3.4.1 Requisitos Funcionais.....	35
3.2.3.4.2 Requisitos Não-Funcionais .....	35
3.2.3.5 Modelagem .....	35
3.2.3.6 Definições dos Personagens .....	36
3.2.3.7 Definições dos Cenários .....	38

3.2.3.8 Gameplay.....	38
3.2.3.9 Cronograma de Desenvolvimento .....	39
3.3 DESENVOLVIMENTO DO PROTÓTIPO. ....	39
3.3.1 Usando Construct 2 .....	39
3.3.2 Os Testes para Validação.....	42
<b>4 O PROTÓTIPO DESENVOLVIDO.....</b>	<b>44</b>
<b>5 TESTES E RESULTADOS .....</b>	<b>48</b>
<b>6 CONCLUSÃO.....</b>	<b>51</b>
6.1 TRABALHOS FUTUROS.....	52
<b>REFERÊNCIAS.....</b>	<b>53</b>
<b>APÊNDICE A – Questionário de Avaliação do Protótipo.....</b>	<b>56</b>

## 1 INTRODUÇÃO

Sistema Operacional (SO), área do conhecimento abordada neste trabalho, se trata de um programa que serve como interface entre o usuário e os dispositivos de hardware, e está presente em todos os computadores atuais. Além do mais, o SO também é uma disciplina obrigatória nos cursos de computação. Desta forma, para melhorar a compreensão desse estudo, é importante contextualizar o seu papel como programa, para entender sua natureza como disciplina.

Apesar de a disciplina possuir um apelo prático, tendo em vista que os conceitos são todos aplicáveis em SO reais, ela é essencialmente teórica, com a exposição de muitos conceitos envolvidos e algoritmos existentes. Dessa forma, seu conteúdo muitas vezes não é absorvido adequadamente. Um desses assuntos é Gerencia de Memória, que aborda um tópico muito importante, chamado Memória Virtual (MV). Este tópico foi ganhando importância à medida que ocorriam avanços nos componentes de hardware e software, e o desempenho da memória principal (RAM - memória de acesso aleatório), se tornaria insuficiente para a execução de algumas aplicações.

A MV é um método, que foi criado para auxiliar a memória RAM no momento que ela se encontra com dificuldade na hora da execução de um processo. Quando isso ocorre, a MV se responsabiliza por criar um tipo de concatenação entre a memória principal e um espaço de endereçamento do disco rígido. Quando a memória RAM se encontra com o espaço insuficiente a MV automaticamente carrega todos dados daquela para o espaço de endereçamento conhecido como arquivo de paginação, dessa forma a memória RAM fica com seu espaço livre para concluir suas tarefas. Através de diferentes técnicas e mecanismos complexos e de difícil assimilação, como por exemplo, o mapeamento, paginação, segmentação, que são técnicas necessárias para o entendimento do assunto.

A complexidade envolvida nos conceitos teóricos, assim como a falta de ferramentas para representar visualmente tais conceitos, são os principais motivos para dificultar a boa assimilação de alguns assuntos do seu conteúdo. Com isso alguns problemas são clássicos como a dificuldade de abstrair o conhecimento de forma sucinta pelos alunos, a carência de recursos didáticos capazes de representar de forma visual e dinâmica, as técnicas de funcionamento da MV, aplicadas pelo SO, pouca motivação em estudar o assunto pela metodologia e recursos tradicionais adotados pela maioria dos professores da disciplina.

Pode-se observar então que é bastante relevante e justificável a criação de uma ferramenta educacional para o auxílio no ensino dos tipos de MV. Para tanto, a ferramenta

desenvolvida é um jogo educativo que é um objeto que estimula e favorece o aprendizado de pessoas, através de um processo de socialização que contribui para a formação de sua personalidade.

A pouca quantidade de ferramentas para simular o funcionamento da MV, de forma dinâmica, é outro motivo que justifica este trabalho. O desenvolvimento do jogo poderá facilitar o entendimento dos complexos conceitos teóricos de memória virtual, nos quais estão incluídos a paginação e segmentação. Essa ferramenta possibilitará ao aprendiz uma boa assimilação do assunto, assim, amenizará o problema referente a pouca disponibilidade de ferramentas, com recursos visuais e dinâmicos, referentes ao assunto abordado. O jogo irá proporcionar uma maior motivação na aprendizagem de MV, através de um ambiente atrativo, iterativo e desafiador.

A principal finalidade deste trabalho é auxiliar no processo de ensino e aprendizagem na disciplina de Sistemas Operacionais, através de um jogo educativo sobre MV, que possa representar o seu funcionamento e aumentar o estímulo para os estudos desse assunto. Para atender o objetivo geral, alguns objetivos específicos são válidos, como colocar no jogo as partes mais complexas e aprofundadas do tópico MV como as funções do endereço da tabela de páginas, do deslocamento, do endereço de frame, do endereço virtual, do endereço físico, etc.; Melhorar a compreensão da teoria da disciplina; Aumentar a motivação dos alunos, quanto ao estudo do assunto, através de um recurso atraente e dinâmico; Suprir a carência de materiais didáticos, capaz de representar visualmente os conceitos de memória virtual.

Para o desenvolvimento do projeto, foi utilizada a metodologia *Game Design*, onde são especificadas as características e requisitos do jogo, e feita a modelagem com alguns diagramas da linguagem de modelagem UML. Essa metodologia serviu como base para o desenvolvimento do protótipo, que utilizou a *engine* Construct 2, que trabalha com a linguagem HTML 5, que é uma linguagem para estruturação e apresentação de hipertexto para web e tem a função de fazer a parte de codificação do jogo, e também funciona junto ao *JavaScript* que é uma linguagem de programação interpretada. O jogo foi validado com os testes de regressão, atualizando o sistema a cada erro encontrado.

O trabalho encontra-se estruturado em 6 capítulos, incluindo este “INTRODUÇÃO”. No capítulo seguinte “FUNDAMENTAÇÃO TEÓRICA”, aborda uma revisão da literatura quanto algumas ferramentas existentes sobre a temática estudada, ressaltando as suas vantagens e desvantagens, também será estudado os conceitos fundamentais sobre a importância da MV e dos problemas no gerenciamento de memórias

virtuais, especialmente, o que tange a memória paginada e segmentada, e ainda será abordado um estudo sobre jogos educativos.

No capítulo 3, “MÉTODOS E TÉCNICAS” mostrará todas as tecnologias utilizadas para a construção do protótipo, destacando os métodos para sua construção.

O capítulo 4, “O PROTÓTIPO DESENVOLVIDO” será apresentado todas as funcionalidades do jogo, destacando os objetivos que o jogador terá.

No capítulo 5, “TESTES E RESULTADOS” irá comentar um teste em forma de questionário aplicado em discentes que estavam estudando a temática abordada.

O último capítulo, “CONSIDERAÇÕES FINAIS”, contém uma avaliação quanto à contribuição do estudo para a comunidade acadêmica, destacando as possibilidades para trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção contém alguns conceitos da temática estudada, assim como algumas ferramentas educativas, os problemas que assolam o assunto e um estudo sobre jogos educativos.

### 2.1 FUNCIONAMENTO DA MEMÓRIA VIRTUAL

Com os avanços dos componentes computacionais o desempenho da memória principal, se tornou insuficiente para a execução de algumas aplicações. Dessa forma tornou-se necessária à criação de um novo método para suprir tais necessidades.

Em meados de 1957 um cientista alemão Fritz-Rudolf Güntsch criou um método muito conhecido chamado MV. A MV é um método, que foi criado para auxiliar a memória RAM no momento que ela se encontra com dificuldade na hora da execução de um processo. Quando isso ocorre a MV se responsabiliza em criar um tipo de concatenação entre a memória RAM e um espaço de endereçamento do disco rígido. Assim quando a memória RAM se encontra com o espaço de memória insuficiente a MV automaticamente carrega todos dados da memória RAM para o espaço de endereçamento conhecido como arquivo de paginação. Desse modo a memória RAM fica com seu espaço livre para concluir sua tarefa (DENNING & PETER, 1997).

#### 2.1.1 MEMÓRIA VIRTUAL POR PAGINAÇÃO

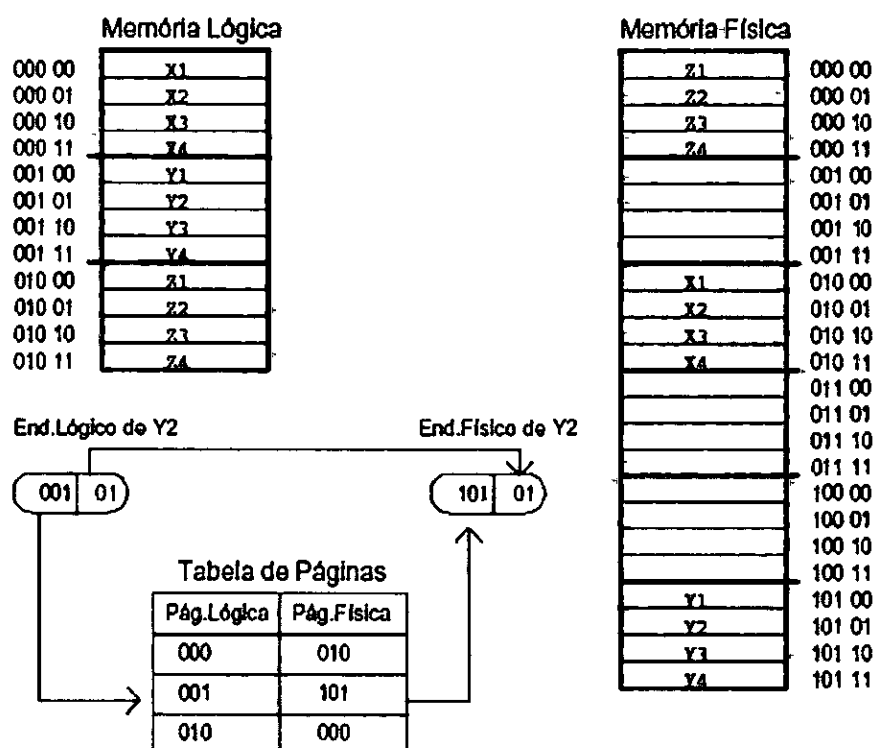
Uma das técnicas mais utilizadas em computadores com MV é denominada paginação. Em computadores que não utilizam a MV os endereçamentos virtuais são idênticos aos endereços físicos. Porém ao utilizar a MV esse endereço tem suas particularidades diferenciadas. A memória física é dividida em blocos de tamanho fixos denominados molduras de páginas (*page frames*). Já a memória lógica é dividida em blocos de tamanho fixos denominados páginas (*pages*) (TANENBAUM, 1999). As páginas e as molduras de páginas são sempre do mesmo tamanho. Quando um programa tenta usar uma página virtual que não está mapeada é gerada uma interrupção da CPU para o sistema operacional a fim de buscar esta página na memória.

A tabela de páginas é montada durante o instante em que o processo é carregado e servirá para informar qual é a página física correspondente a cada página lógica. Para que um

programa possa ser executado corretamente, é necessário que o endereço lógico especificado em cada instrução seja traduzido no endereço físico que representa as posições reais de memória.

Na Figura 1 o endereço lógico é constituído de 2 partes: a primeira (bits de mais alta ordem) corresponde ao número da página lógica e a segunda representando o deslocamento dentro desta página (bits de mais baixa ordem).

FIGURA 1. ESQUEMA DA PAGINAÇÃO.



FONTE: SILBERSCHATZ,2002.

A forma como a tabela de páginas é implementada, é um fator importante e que deve ser pensado, uma vez que ela deve ser consultada a cada acesso à memória. Quando um processo é escalonado e outro assume o processador, a tabela de páginas será executada e deverá ser copiada do descritor de processo (DP) para os registradores (TANENBAUM,1999). As entradas na tabela de páginas são compostas por alguns campos que permitem localizar o endereço efetivo de cada página na memória principal (quando carregada) ou na memória secundária (área de *swap*).

As seguintes informações são mantidas no número da página virtual (NPV) que identifica uma única página virtual correspondente à entrada. De acordo com o tamanho de página utilizado, o endereço na memória secundária pode ser calculado.

1. Bit de presença: indica se a página está ou não carregada na memória principal (a página está carregada se o bit possui valor 1).
2. Endereço do bloco: se a página estiver carregada na memória principal, este campo informa o endereço físico em que está alocada; um bloco, ou janela (frame), é um conjunto de células da memória principal que comporta dados de uma página.
3. Bits de proteção: definem as permissões de acesso à página, normalmente através de 2 ou 3 bits (leitura, gravação e eventualmente execução).
4. Bit de modificação: indica se a página carregada na memória teve ou não seu conteúdo alterado; se teve, o conteúdo deverá ser atualizado na memória secundária antes de a página deixar a memória principal.

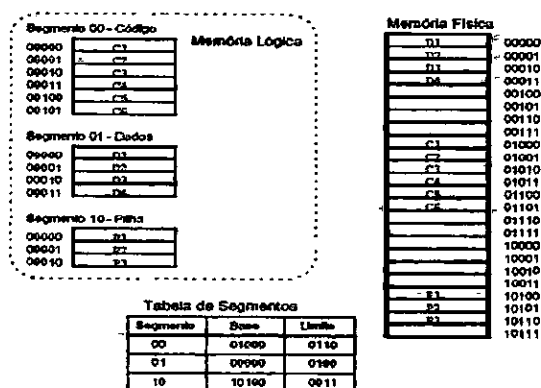
### 2.1.2 Memória Virtual por Segmentação

O real motivo da criação da MV paginada é facilitar o gerenciamento da memória, porém os compiladores não tem a capacidade de reconhecer a memória lógica dividida por páginas, sendo assim, faz-se necessário a divisão da memória lógica por meio de segmentação.

Existem muitas características a serem consideradas para descobrir se um programa pode ser segmentado, tais como: código, dados alocados dinamicamente, dados alocados estaticamente e pilha de execução.

Em um sistema que utiliza segmentação, o endereço lógico também é dividido em: número do segmento e deslocamento dentro do segmento (semelhante ao esquema de paginação), como é mostrado na Figura 2. Os segmentos não necessariamente precisam ter o mesmo tamanho. No entanto, existe um tamanho máximo definido para os segmentos.

**FIGURA 2. EXEMPLO DA TRADUÇÃO DO ENDEREÇO LÓGICO PARA O ENDEREÇO FÍSICO.**



FONTE: SILBERSCHATZ, 2002. ADAPTADO PELO AUTOR

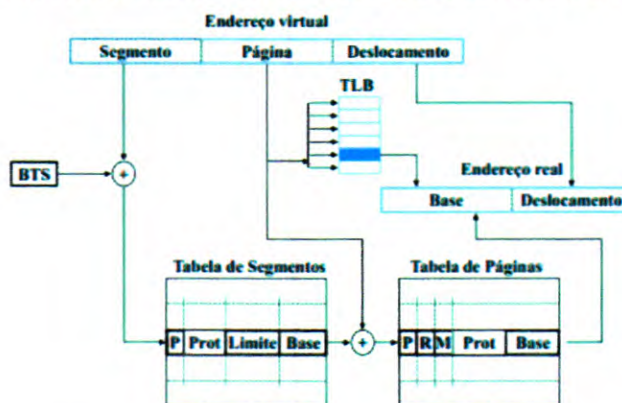
Quando o deslocamento é maior que o limite especificado na tabela de segmentos significa que o processo que está executando está tentando endereçar além do final do segmento, ou seja, fora da sua área de memória lógica. Nesse instante será gerada uma interrupção de proteção.

### 2.1.3 Memória Virtual – Paginada e Segmentada

Essa técnica aborda os dois métodos de MV em um só. As vantagens da MV Paginada é que todos os blocos têm o mesmo tamanho, a sua localização e a substituição é mais fácil e sua tradução é mais rápida. Já na MV Segmentada seu bloco de divisão é lógico, não físico e não há nenhuma fragmentação interna (mas existem externas).

A MV Paginada e Segmentada é dividida em segmentos de tamanho variável, mas sempre múltiplos de um tamanho básico, assim, um segmento é dividido em páginas. A tradução tem duas etapas, que é o endereço virtual (segmento *offset*) e o endereço físico (como na MV paginada), de acordo com a Figura 3.

FIGURA 3. MEMÓRIAVIRTUAL SEGMENTADA E PAGINADA



FONTE: TANENBAUM, CAP 3, 1999, ADAPTADA PELO AUTOR.

## 2.2 PROBLEMAS NO ENSINO E APRENDIZAGEM DA TEMÁTICA

A disciplina de SO envolve uma grande quantidade de conteúdo teórico, com muitos conceitos e técnicas sobre o funcionamento dos sistemas reais. Segundo Maia (2001), “As disciplinas que envolvem SOs se caracterizam por possuírem uma alta carga de conteúdo teórico somados a conceitos, em sua maioria, abstratos e interdependentes”. Essa característica causa certo desinteresse dos alunos quando não conseguem absorver claramente o assunto.

Essa complexidade, associada à baixa quantidade de softwares educativos (SE) de SO, que representem visualmente os mecanismos de funcionamento dos sistemas, causam outra dificuldade, pois exigem que os alunos criem diferentes abstrações para que possam assimilar corretamente o assunto. Maia (2001), afirma que “Lecionar disciplinas da área de SOs é um desafio em virtude das características e peculiaridades das informações a serem repassadas”. Já Lopes, A (2012) reforça que uma das causas desse problema está na escassez de ferramentas para traduzir a teoria apresentada em conceitos práticos.

A necessidade de atividades práticas é fundamental para os alunos de SO, pois permite aplicar os conhecimentos aprendidos em sala de aula, nos sistemas reais, oportunizando o desenvolvimento de novos projetos e permitindo melhorias nos sistemas já existentes. Entretanto, esse tipo de atividade é mais uma das dificuldades presente nessa disciplina, pois exigem dos professores e alunos, bons conhecimentos em outras áreas do curso, como programação e arquitetura de computadores (LOPES, 2008).

Para dar início as inúmeras dificuldades do tópico MV, uma delas é quando o aluno confunde ou pensa que a MV é a mesma coisa de *swapping*.

A técnica de *swapping* permite aumentar o número de processos compartilhando a memória principal e, conseqüentemente, o sistema. Em sistemas que implementam essa técnica, quando existem novos processos que desejam ser executados e não existe memória real suficiente, o sistema seleciona um ou mais processos que deverão sair da memória para ceder espaço aos novos processos (MAIA, 2001).

De modo geral a MV se diferencia do *swapping*, pois ela tende a aumentar, ou seja, a memória física concatenada com uma parte do HD, e utiliza o *swapping* para fazer a trocas entre os processos da memória principal com a secundária.

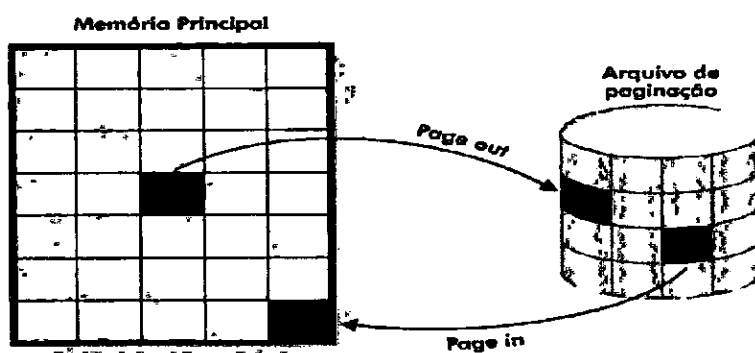
No tópico MV, o aluno tem grandes dificuldades em entender o assunto, pois na parte que fala de paginação fica difícil abstrair quando um programa é executado, as páginas virtuais são passadas da memória secundária para a memória principal e colocadas em endereços de frames. Toda vez que o programa fizer relação a um endereço virtual, o mecanismo de mapeamento localizará, na tabela de páginas da tabela do processo, o endereço físico do frame (MACHADO & MAIA, 2002).

Outro problema clássico em MV é a política de substituição de páginas, pois o aluno não consegue visualizá-la, ou seja, às vezes o mesmo não sabe que páginas remover. Quando não existem páginas livres disponíveis na memória e novos *frames* devem ser alocados, a política de substituição (*replacement policy*) de páginas determina, dentre as diversas páginas residentes, quais devem ser realocadas (SILBERSHATZ & GALVIN & GAGNE, 2004).

Qualquer estratégia de substituição de páginas deve considerar se uma página foi ou não modificada, antes de liberá-la, caso contrário, possíveis dados armazenados na página serão perdidos. Sempre que o sistema liberar uma página desse tipo, ele antes deverá gravá-la na memória secundária, preservando seu conteúdo. O sistema mantém um arquivo de paginação onde as páginas modificadas são armazenadas. Sempre que uma destas páginas for novamente referenciada, ela será trazida novamente para a memória principal. (MACHADO & MAIA, 2002).

A classificação da política de substituição de páginas poderá ser entendida conforme seu escopo, ou seja, local ou global. Na política local, somente as páginas do processo que gerou o *page fault* como mostra a figura 4, são candidatas à realocação. Já na política global, todas as páginas restantes são avaliadas independente do processo que gerou o *page fault*.

FIGURA 4-SUBSTITUIÇÃO DE PAGINAS



FONTE: MACHADO E MAIA (2002).

Independente se a política seja local ou global, os algoritmos de substituição de páginas devem ter o objetivo de selecionar aquelas que tenham poucas chances de serem utilizadas novamente num futuro próximo. Quanto mais elaborado e sofisticado é o algoritmo, maior também é o *overhead* para o sistema. (MAIA, 2001).

### 2.3 A IMPORTÂNCIA DAS FERRAMENTAS EDUCATIVAS

Segundo Komosinski (2000), o processo de ensino é um método muito complexo, principalmente quando se usa meios típicos, ou seja, giz, lousa, livro, etc. As ferramentas educativas buscam a facilidade no entendimento do aluno. A inovação, a criatividade e a dinamização são encontradas nas mesmas e há diversos recursos que possam ajudar na aprendizagem. Atualmente, os docentes encontram muitas alternativas para proporcionar seu ensino rico em informações. Estudos mostram a importância do uso dessas ferramentas educativas que são simuladores, jogos educativos, tutoriais e objetos de aprendizagem.

Segundo Almeida (2005, p. 8), o manuseio das inovações tecnológicas ocorre alterações nos métodos de trabalho dos docentes, gerando modificações no funcionamento das instituições e no sistema educativo. Portanto, é importante analisar em que medida o uso de ferramentas educativas pelo docente, está colaborando para a aprendizagem dos discentes. (SOFFA, M, M & ALCÂNTARA, P,R,C, 2008).

O uso da tecnologia não nos autoriza a pensar que nesse novo contexto a figura do educador venha a se tornar desnecessária. Os educadores devem transformar a utilização do computador numa abordagem educacional que favoreça efetivamente o processo de conhecimento do aluno. Além disso, é preciso uma modernização no sistema educacional e da adoção de novos programas, métodos e estratégias de ensino, e a compreensão, por parte dos educadores, da transitoriedade do sistema. “É preciso também uma organização política e de competência.” (MORAIS .R, 2003, p11).

Para Ajudar a resolver os problemas de ensino e aprendizagem na educação, estão como alternativas, os jogos educativos como uma maneira de atrair os alunos a melhorar seus desempenhos. Os jogos educativos trabalham a coordenação motora, a rapidez, agilidade, e ajudam na evolução intelectual dos alunos.

Com o avanço das ferramentas educativas, ocorrem muitos impactos positivos, mas atualmente alguns professores não usam essas ferramentas, dificultando às vezes o assunto dado na disciplina.

## 2.4 FERRAMENTAS EDUCATIVAS EXISTENTES

Os Softwares Educativos (SE) são programas com características essencialmente didáticas, que utilizam os recursos potenciais do computador para auxiliar o processo de ensino-aprendizagem em diferentes áreas do conhecimento. Neste contexto, podem ser considerados como uma alternativa viável para minimizar as dificuldades presentes na disciplina de SO, focando as atenções para o tópico de MV.

Para demonstrar a capacidade dos SEs em tratar as dificuldades aqui discutidas, pode-se dizer que, tanto a complexidade, quanto as abstrações necessárias para o entendimento do assunto, podem ser reduzidas por mecanismos visuais que representem dinamicamente os complexos conceitos teóricos. Essa característica pode ser encontrada nos simuladores, jogos educativos, animações e objetos de aprendizagem. De acordo com Maia (2001), “[...] decorrente da facilidade imposta por estas ferramentas, entre elas os simuladores, que por meio de interação visual, permitem ao aluno absorver melhor o conteúdo sem agregar a habitual desmotivação imposta pelo método tradicional de ensino”.

A questão da motivação dos alunos para o estudo pode ser melhorada com o uso de jogos educativos, do tipo de SE, que propõe desafios dentro de um ambiente estimulante e mais atraente que muitos recursos didáticos tradicionais. Conforme REIS (2009), “[...] o uso de jogos na educação favorece o desenvolvimento da lógica, estratégia, análise e algumas vezes da memória, partindo-se da tentativa e erro no vencer das fases do jogo”.

Uma vez demonstradas às características que favorecem o uso dos SEs, como alternativa para os problemas da disciplina de SO. Serão apresentadas algumas ferramentas educativas sobre o tópico em questão, que embora sejam simuladores, categoria diferente dos jogos educativos, ajudaram no desenvolvimento deste trabalho.

A pesquisa sobre os SE identificou vários programas que abordam o mesmo tema, embora não se tenha encontrado nenhum jogo educativo do assunto. Todos os softwares pesquisados foram analisados sob o ponto de vista de usabilidade, assunto abordado e, principalmente, capacidade de resolução dos problemas da disciplina.

A Tabela 1 apresenta os softwares encontrados na pesquisa e que, de alguma forma, contribuíram para o desenvolvimento deste trabalho.

TABELA 1–SOFTWARES EDUCATIVOS DE SO

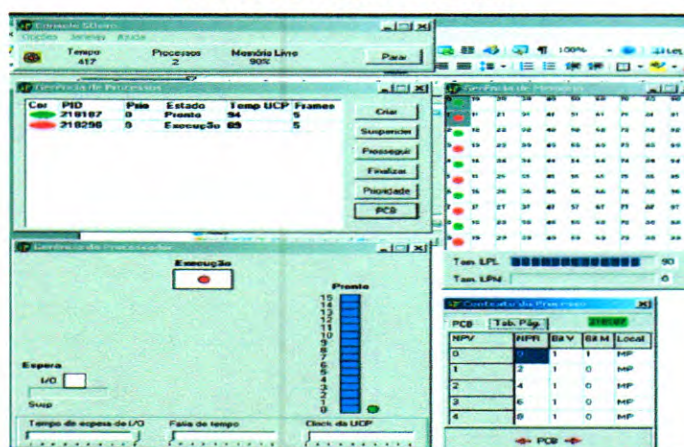
Item	Software-Educativo	Categoria
1	Sim Memory [6]	Simulador
2	Tbc-So [7]	Simulador
3	Sime [4]	Tutorial
4	Fast Dynamic Memory [8]	Simulador
5	Moss Memory Management Simulator [9]	Simulador
6	Simulador de Gerência de Memória [10]	Simulador
7	Alg OS [11]	Simulador
8	Virtual Memory Simulation [12]	Simulador
9	Sosim [3]	Simulador
10	Simulador de Sistema Operacional Genérico (SSOG) [13]	Simulador

Todos os programas foram analisados, entretanto, será discutido apenas cinco, do total de softwares analisados, que são os seguintes:



- SOsim [3]: É uma ferramenta que simula os conceitos e técnicas do SO, abordando os seguintes tópicos: Gerência de Processos e MV por paginação. Em relação ao assunto MV, o simulador trata muito pouco, restringindo-se a uma mera demonstração gráfica da alocação dos processos na memória e permitindo uma configuração básica dos parâmetros referentes ao tamanho dos processos. Além dessas limitações, foram identificados os seguintes pontos fracos: não exibe nenhum conteúdo referente aos assuntos; não possui mecanismo autoexplicativo e sua execução é restrita ao sistema operacional Windows nas versões Xp, Vista e Sete. O simulador é mostrado na figura 5.

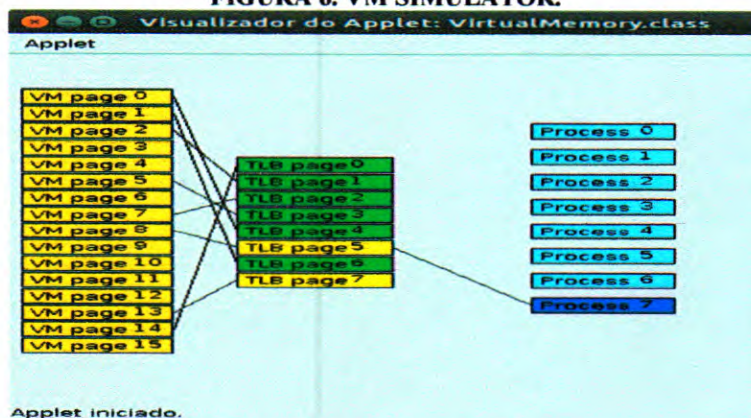
FIGURA 5: TELA DO SOSIM



FONTE: MAIA, 2001

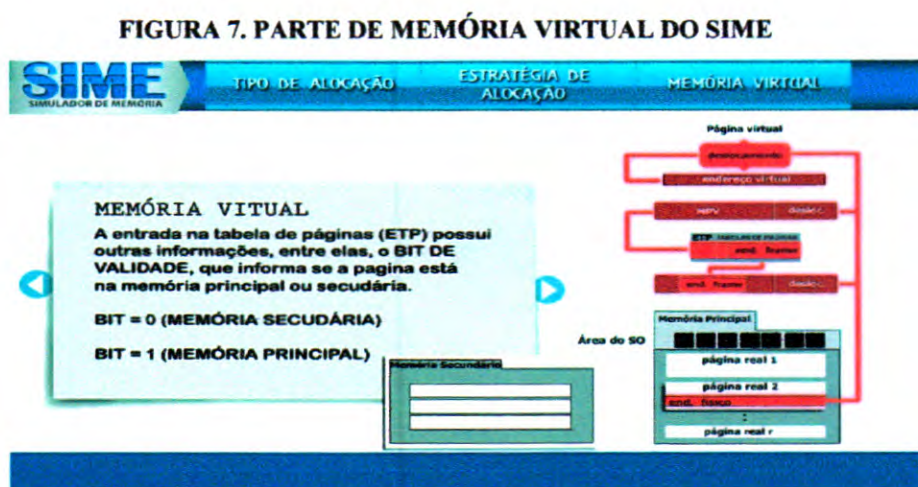
- Virtual Memory Simulation [12]: É um simulador on-line desenvolvido em Java, que possui uma interface textual, sem recursos gráficos. Sua utilização não é simples e os resultados são gerados após a execução. Assim como o SOsim, não possui mecanismos autoexplicativos. A ferramenta é mostrada na figura 6.

FIGURA 6. VM SIMULATOR.



FONTE: COUTINHO

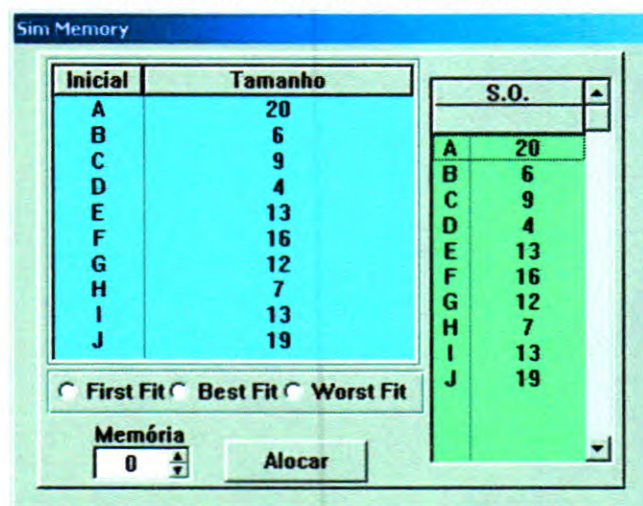
- SIME [4]: É um tutorial desenvolvido em *ActionScript 2* para a plataforma Adobe Flash que trata das técnicas de gerencia de memória. Ele possui muitos mecanismos autoexplicativos, mas a parte de memória virtual paginada ainda é um pouco confusa. O software é mostrado na figura 7



FONTE: LOPES,A,R, 2012.

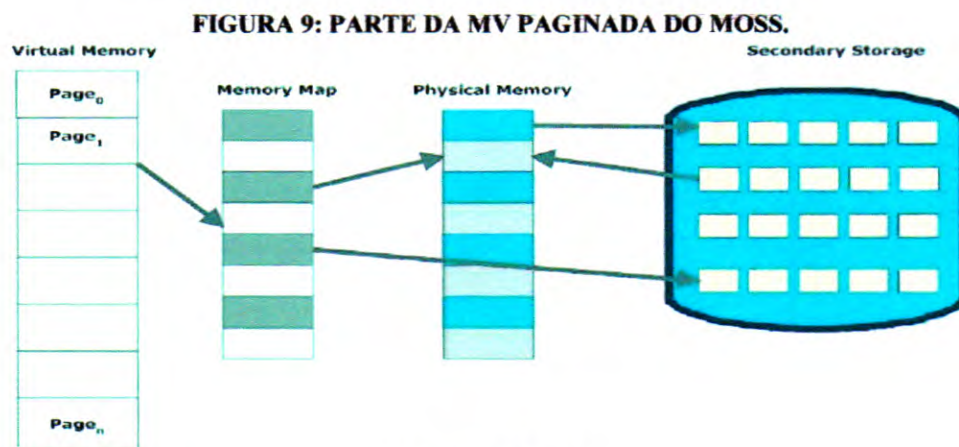
- SIM MEMORY[6]: É um simulador de Gerenciamento de Memória que trata todas suas estratégias, foi desenvolvido utilizando a linguagem em Clarion. A ferramenta como a memória é usada utilizando 3 métodos: First Fit, Best Fit e Word Fist. O problema dessa ferramenta que também não possui textos autoexplicativos para o que o usuário possa fazer o seu uso de forma adequada. O simulador é mostrado na figura 8.

**FIGURA 8. SIMULADOR SIM MEMORY**



FONTE: LOPES, ALINE BARBOSA 2008

- MOSS [9]: Se trata de um repositório com vários simuladores interativos do tópico gerencia de memória, todos esses simuladores foram desenvolvidos em Java, o ambiente é bem rico e agradável. A sua única desvantagem que só roda em plataforma Windows. A ferramenta é mostrada na figura 9.



FONTE: ONTKO

## 2.5 JOGOS EDUCATIVOS

O trabalho ora proposto se baseia na construção de um protótipo de um jogo educativo. O jogo educativo é uma ferramenta desenvolvida para auxiliar as pessoas sobre um determinado assunto específico. Eles podem ser compostos por objetivos, regras, adaptação, resolução de problemas, interação, todos representados como uma história. Os mesmos dão as pessoas necessidades fundamentais de aprendizagem, fornecendo prazer, envolvimento, estrutura, motivação, satisfação, adrenalina, criatividade, interação social e emoção.

De acordo com o Relatório Horizon (2010), os jogos são uma forma dos alunos experimentar as lutas e sucessos de forma colaborativa, trabalhando no sentido de uma solução para um problema complexo no interior de um enredo interessante. Ele também aponta que, os jogos têm sido uma motivação em salas de aula. O relatório define três tipos de jogos educativos – “jogos que não são digitais; jogos que são digitais, mas que não são de colaboração; e jogos digitais colaborativos”. O foco principal desta discussão é jogos digitais que não são colaborativos e alguns jogos digitais colaborativos. Os jogos que não são colaborativos são do tipo *single player*, já os digitais colaborativos são os jogos *multiplayer* que exigem que os jogadores interajam uns com os outros, ou seja, mais de um jogador. Alguns exemplos de jogos que não são colaborativo são: *Craft* e *SimCity*, ambos para celular. Um exemplo de jogos colaborativos é *World of Warcraft*, *Moonbase Alpha*, e *Americas Army*.

### **2.5.1 A Grande Vantagem de Jogos Educativos**

A principal vantagem de jogos educativos é que os alunos podem trabalhar em várias habilidades e disciplinas em todo o currículo de uma só vez. O Relatório Horizon 2010 expande essa ideia e inclui uma longa lista de benefícios. Com os jogos educativos, os alunos podem trabalhar em habilidades em todas as áreas da educação tradicional e, ao mesmo tempo, incluindo as de pesquisa, resolução de problemas e liderança.

A escola *New York City* é um grande exemplo de como um currículo pode ser estruturado de forma a permitir que os alunos trabalhem de forma colaborativa e em conjunto para resolver problemas. Eles também usam frequentemente jogos para ajudar os alunos a ter uma compreensão mais profunda do material que eles estudam (Relatório Horizon, 2010).

### 3 MÉTODOS E TÉCNICAS

#### 3.1 TECNOLOGIAS UTILIZADAS PARA A CONSTRUÇÃO DO JOGO

Nesse tópico será mencionado todas as tecnologias que foram utilizadas para a construção do projeto.

##### 3.1.1 Game Design

Segundo Curti (2006) *Game Design* é uma extensão da prática do design e significa desenho estrutural de game ou projeto de game.

Game Design é a etapa do processo de desenvolvimento de games que determina os detalhes da funcionalidade do gameplay (também chamado de jogabilidade), as escolhas que o jogador terá dentro do mundo virtual do game assim como as ramificações que suas escolhas vão originar, quais as condições de vitória e derrota, como o game será controlado e as informações que o jogador deverá receber. (PERUCIA et al., 2005).

O game design é de fundamental importância para projetar qualquer tipo de jogo, pois não adianta a pessoa ter boas habilidades em programação, *engines* e outros, mas não ter o hábito de planejar para depois projetar os jogos.

Para o projeto de um jogo digital poderá surgir varias ideias a partir de pequenos fatos como:

- Conceitos originais apresentados por uma pessoa da empresa ou grupo;
- Conceitos originais estabelecidos por uma pessoa de fora da empresa ou do grupo;
- Uma continuação para um jogo que já existe;
- Uma mudança em um personagem de um jogo já existente;
- Um game baseado em um personagem ou história existente (como personagens de filmes, TV ou quadrinhos);
- Uma simulação de outros jogos (como jogos de tabuleiro ou de cartas);
- Um jogo direcionado para uma área demográfica específica;
- Uma simulação de um fato que ocorreu no mundo real;
- Um jogo projetado para aproveitar uma plataforma de jogos específica (os jogos interativos) (TYSON, 2007).

No *game design* existem documentos para o planejamento do jogo, o nome do mesmo é *Game Design Document*, que vem a ser toda a documentação do jogo e seu andamento durante toda a fase do desenvolvimento, como resultados de pesquisas, prazos de desenvolvimento, informações de todos os objetos do jogo, objetos de mercado do jogo, etc.

Segundo Perucia et. al. (2005), os itens a seguir podem ser definidos e detalhados em um *Game Design Document*:

- Conceito: nome do *game*, descrição básica do *game*, gênero de *game*,
- Descrição do público-alvo, estória e regras principais do *game*.
- Especificações técnicas: hardware; sistema operacional; gráficos e requerimentos de software.
- Especificações do *game*: número de fases; nível de dificuldade; descrição dos modos de se jogar; quantidade de vidas dos personagens; sistema de pontuação; sistema de ranking; configuração disponível; número de jogadores; recursos de carga e gravação; descrição básica dos personagens; sistema de câmera; itens do *game*; itens do cenário; tabela de itens; evolução de fases e tabela de mensagens.
- Dispositivos de entrada: dispositivos de entrada para os menus e de jogo; definição de teclas e botões e suporte para mouse.
- Design gráfico e arte: abertura; descrição de layout de menus e telas; descrição do layout de fases; definição de fases; definição do final do *game*.
- Sonorização: definição das músicas nos menus e nas fases; definição dos efeitos sonoros de menu, nas fases e outros.
- Desenvolvimento: alocação de pessoal envolvido; cronograma e metas.

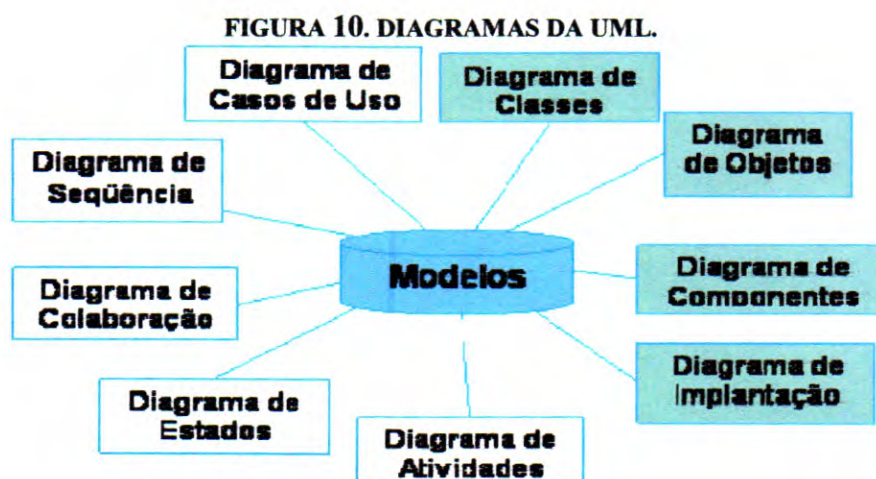
O *game design document* também pode contemplar informações como: nome do fabricante, contato com o cliente, verba disponível, histórico do fabricante, histórico do *game*, descrição da concorrência, legislação pertinente ao país ou tipo de *game* em questão, formas de distribuição, necessidade de tradução para algum idioma estrangeiro, divulgação e peças (manual de instruções, site, embalagem, etc). (AZEVEDO et. al., 2005).

### 3.1.2 A UML

UML é uma linguagem visual para especificar, construir e documentar os artefatos de sistemas intensivos de software. Projetos de software complexos são difíceis de descrever com texto. O entendimento do sistema pode ser facilmente transmitido através de diagramas

usando UML. A Modelagem UML oferece três benefícios principais como visualização, validação, uma comunicação clara.

A UML foi aprovada pela OMG como um padrão em 1997. Ao longo dos últimos anos, têm sido feitas pequenas modificações para a linguagem. A UML 2 foi a primeira grande revisão da linguagem. Na figura 10 temos todos os diagramas da UML, onde em cinza estão os estruturais e em branco, os comportamentais.



FONTE: SQL MAGAZINE

### 3.1.3 A Engine Construct 2

O *Construct 2* é uma *engine* que gera jogos 2d em HTML5. O futuro dos jogos web é desenvolvê-los para que sejam suportados por uma enorme gama de navegadores e dispositivos móveis. Essa ferramenta tem a versão gratuita que conta com até cem eventos, que já é o suficiente para fazer jogos com qualidade.

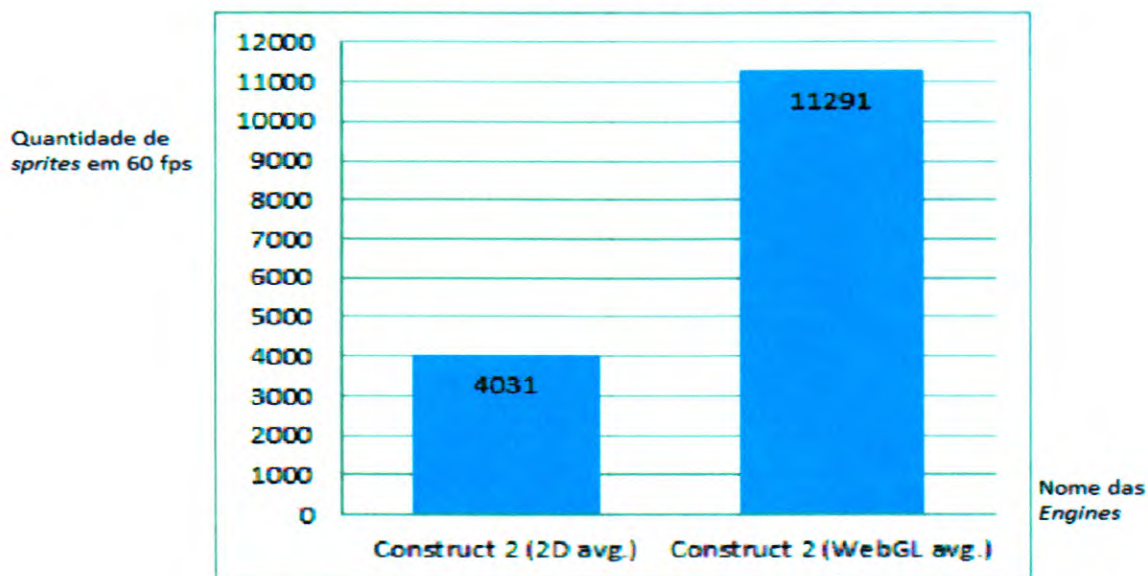
Em se falando de desempenho, um dos requisitos que mais interessa, aos desenvolvedores, durante as pesquisas foram comprovados que o *construct 2* se saiu muito bem em relação a outras *engines* que trabalham também com HTML5, que são: GameSalad, Stencyl e Game Maker.

Para resumir os resultados foram feitos alguns testes de desempenho da seguinte forma, o usuário teria que criar uma quantidade de *sprites*<sup>1</sup> na tela até que a velocidade caia de 60 fps para 30 fps. Antes de testar *Construct 2* com outras ferramentas, foi obtido uma análise do mesmo, com Canvas 2d e WebGL. Na figura 11 está o resultado médio para *Construct 2* da última versão, tanto para o renderizador Canvas 2D e WebGL. Em média é de 2,8 x WebGL

<sup>1</sup> <http://www.sirmastronger.eti.br/jogos2d/conceitos1.htm>

mais rápido, e alguns ganhos para navegadores individuais foram ainda maiores (SCIRRA, 2014).

**Figura 11. Desempenho de canvas 2d e webGL.**

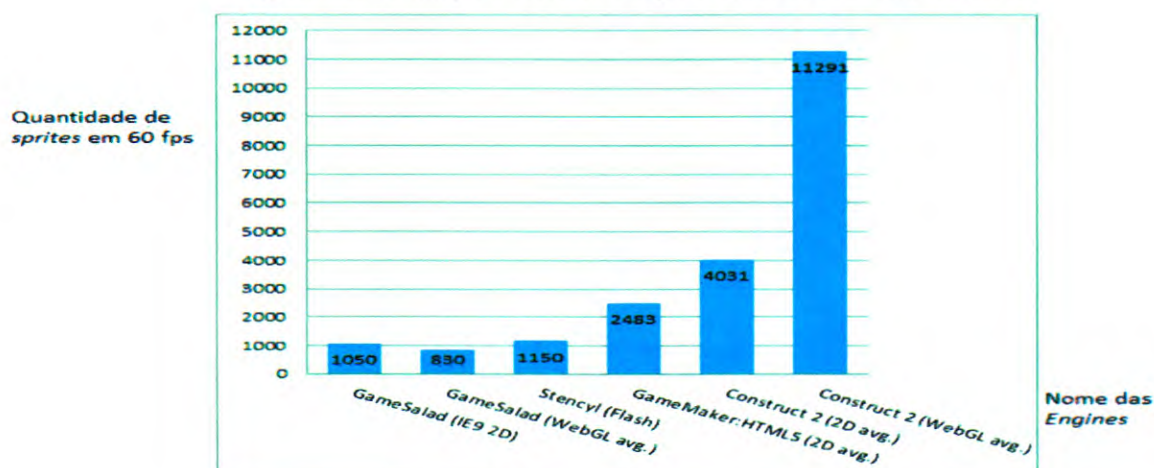


FONTE: SCIRRA, 2014.

No site da *scirra* que é a empresa que desenvolveu a *engine construct 2*, tem uma análise comparativa de muitas *engines* 2d em todo o mundo, segue o gráfico da figura 12.

Na figura 12 estão os resultados gerais, de desempenho das *engines* GameSalad, Stencyl e Game Maker junto com os resultados anteriores, onde pode-se observar que *Construct 2* com WebGL é bem superior das demais.

**Figura 12. Comparativos da engine construct 2 com outras.**



FONTE: SCIRRA, 2014.



A *engine construct 2* está ganhando espaço aos poucos, pois além de ser leve, e de exportar para HTML5, ela exporta também para inúmeras plataformas como Chromo Web Store, Kongregate, Android, Cocoon JS, Black Berry, Windows Phone, Node Webkit, Nitendo Wii, etc. Pode-se observar que é uma ferramenta muito poderosa, e sem dúvida foi por isso que foi escolhida, além disso trabalha totalmente com HTML5 que são agregados ao *JavaScript*, *Canvas* e *CSS*.

### 3.1.4 A Linguagem HTML5

O HTML5 foi concebido para substituir o HTML4, XHTML e HTML DOM Nível 2. Foi especialmente concebido para proporcionar um conteúdo rico, sem a necessidade de plugins adicionais. A versão atual oferece muitas coisas, desde animação para gráficos, música de filmes, e também pode ser usado para construir aplicações web complexas. O HTML5 também é multi-plataforma, ele é projetado para trabalhar usando um PC, um *Tablet*, um smartphone ou *Smart TV*(w3scholl, 2014).

No HTML5 foram estabelecidas algumas padronizações, onde as novas aplicações devem ser baseadas em HTML, CSS, DOM e *Java Script*, sem as necessidades de *plugins* externos. Os tratamentos de erros são mais fáceis de solucionar que as outras versões, os scripts são mudados por mais de marcação, a linguagem HTML5 deve ser independente de dispositivos e o desenvolvimento do projeto deve ser visualmente elegante.

### 3.1.5 Javascript e Canvas

A linguagem *JavaScript* é construída para funcionar em todos os principais navegadores, incluindo Internet Explorer, Firefox e Safari (javascript.about,2014).

A maneira ideal de unir o *JavaScript* a uma página web é necessário ter apenas uma ou mais tags de script no cabeçalho da página que referencia o *JavaScript* que é completamente separado do HTML. Não há necessidade de quaisquer marcas de *script* dentro do corpo da página ou quaisquer manipuladores de eventos ligados dentro de todas as tags HTML. Todo o *JavaScript* para a página web pode ser trabalhado a partir desses arquivos externos e o único código necessário dentro do próprio HTML será o ID de classe e atributos que são necessários para identificar as tags HTML que o *JavaScript* precisa interagir.

O elemento *canvas* é usado para desenhar gráficos, em tempo real, em uma página web, pode ser usado através do *JavaScript*. (w3school,2014).

O elemento *canvas* fornece *scripts* com uma tela de *bitmap* dependendo da resolução, que pode ser usada para o gráfico de renderização, gráficos do jogo, entre outros.

Não é adequado usar um elemento *canvas* para processar um cabeçalho da página, se a apresentação desejada do título é graficamente intensa, ele deve ser marcado por utilizar os elementos adequados, em seguida, usando um estilo CSS e tecnologias de suporte.

### 3.1.6 CSS 3

O termo CSS significa *Cascading Style Sheets*, tem a função de definir como serão mostrados os elementos que estão no código fonte de uma página web e sua maior vantagem é efetuar a segmentação entre o formato e o conteúdo de um documento.

O CSS3 foi dividido em módulos. Ele contém especificações dos modelos anteriores, que foi dividido em partes menores. Além disso, novos módulos foram adicionados. Alguns dos módulos mais importantes são os seletores, o *box model*, fundos e fronteiras, efeitos de texto, transformações 2d e 3d, animações, *layout* e interface com o usuário (W3school, 2014).

## 3.2 PLANEJAMENTOS DO PROTÓTIPO

Para o desenvolvimento de um jogo, assim como de um software tem que haver um processo de desenvolvimento. A fase de planejamento é um ponto inicial para que o jogo alcance seus objetivos. Esta fase conta com três características que é o público-alvo, a plataforma, e a metodologia *game design*.

### 3.2.1 Público-alvo

O público alvo para o protótipo são os alunos de graduação em Ciência da Computação ou cursos afins que possuem SO como disciplina, porém poderão jogar também as pessoas que queiram se divertir.

### 3.2.2 Plataforma

O protótipo será desenvolvido para diversas tecnologias e sistemas, que poderá rodar em qualquer sistema Windows, Linux ou Mac, só precisa ter um navegador. Além disso, o protótipo rodará em dispositivos móveis com sistemas *Android* ou *Ios*.

### 3.2.3 Utilização do Game Design

O *game design* foi de grande importância para o desenvolvimento do jogo, pois a partir da descrição, pode-se ter uma visão geral de como funcionará o jogo, além de uma documentação. Segundo as recomendações do *game design* deve-se seguir algumas etapas, que são descritas nos tópicos abaixo.

#### 3.2.3.1 A Ideia

A ideia do protótipo surgiu da necessidade de um assunto no curso de Ciência da Computação, o qual era MV, que tem conceito e definição complexos, que levam o aluno a abstraí-los. Então a ideia foi desenvolver um jogo de plataforma que melhorasse o entendimento de como funciona a memória virtual.

#### 3.2.3.2 Game Design Document

Nessa etapa foi feito um relatório que engloba todas as informações pertinentes ao protótipo, informações essas que são os requisitos, a modelagem, os conceitos, especificações técnicas, especificações do *game*, dispositivos de entrada, design gráfico, arte, sonorização e desenvolvimento.

#### 3.2.3.3 Descrição do Protótipo

O jogo tem 2 fases, cada uma com subfases, onde o objetivo é fazer com que o jogador tente gerenciar a MV de forma correta, o mesmo tem que buscar os endereços que formam a MV, para encontrar a porta correta, onde passa de subfase. Cada fase tem suas particularidades, como por exemplo, a primeira fase será a gerência da MV Paginada, já a segunda fase a gerência da MV Segmentada.

#### 3.2.3.4 Análise de Requisitos

Nesse tópico será visto todos os requisitos funcionais e não-funcionais.

### 3.2.3.4.1 Requisitos Funcionais

Foram utilizados os requisitos funcionais para descrever todo o comportamento do sistema.

TABELA 2 – REQUISITOS FUNCIONAIS DO PROTÓTIPO

Requisito	Descrição
Controle do personagem	Controlar o personagem para que possa se locomover
Aparição de inimigos	Possuir inimigos em cada fase
Exibir endereços virtuais e físicos	Possuir os endereços que forma a técnica através da combinação de blocos
Representação de portas e portais	Ajudará o personagem a passar das subfases com as devidas combinações de endereços
Destruir inimigos	Além de pegar alguns itens para passar de fase o personagem terá também que destruir inimigos.

### 3.2.3.4.1 Requisitos Não Funcionais

Os requisitos não funcionais são informações que se enquadram com padrões de *software*, ou seja, a sua qualidade.

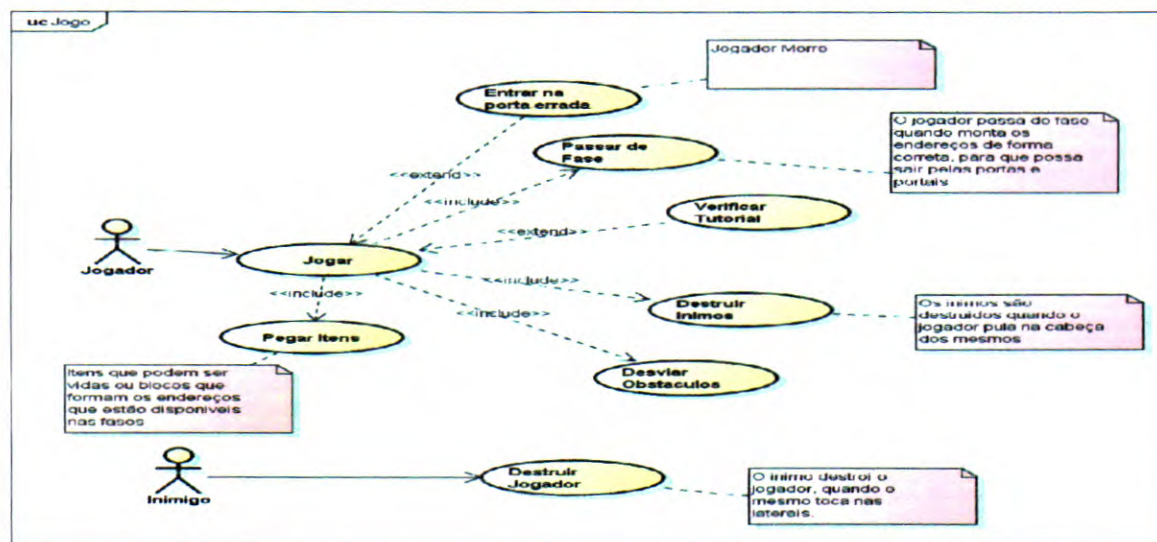
TABELA 3 – REQUISITOS NÃO-FUNCIONAIS DO PROTÓTIPO

Requisito	Descrição
Mecanismos de Auto explicação	Mostrar informações relevantes ao assunto
Tutorial do jogo	Ajudará o jogador nas subfases
Pertinência do conteúdo	Fidelidade do conteúdo com a disciplina.
Exibir introdução do jogo	Mostrar informações na abertura do jogo
Facilidade de uso	Operacionalidade e simplicidade.

### 3.2.3.5 Modelagem

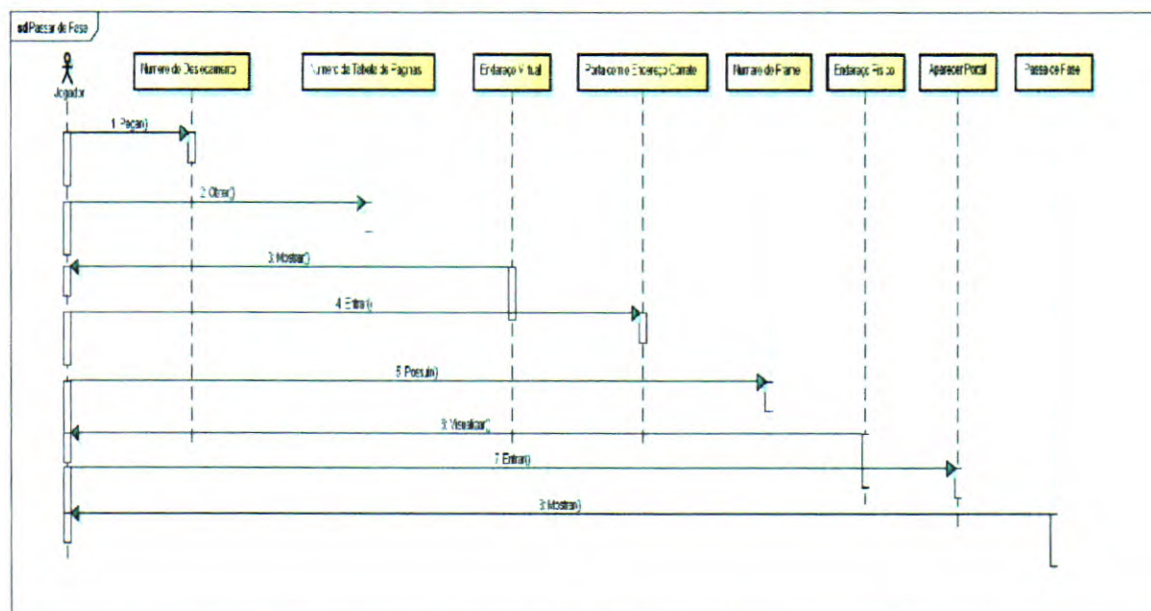
Para a modelagem do protótipo foi utilizada a ferramenta Astah UML, e como foi utilizada a metodologia *Game Designe*, usou-se o diagrama de caso de uso na figura 13 e o de sequência na figura 14, para um melhor entendimento do *game*.

FIGURA 13: DIAGRAMA DE CASO DE USO DO PROTÓTIPO



FONTE DO AUTOR

FIGURA 14: DIAGRAMA DE SEQUÊNCIA











FONTE DO AUTOR

### 3.2.3.6 Definição dos Personagens

A Tabela 4 aborda como todos os personagens foram criados, tendo todas as descrições completas dos objetos.


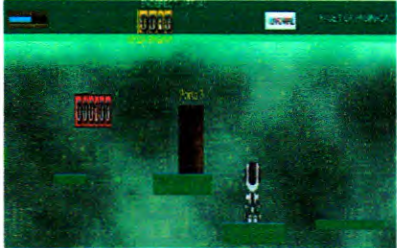
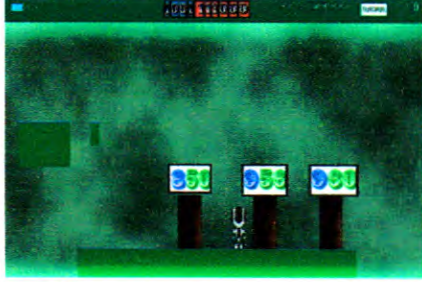
TABELA 4 – DEFINIÇÕES DOS PERSONAGENS

<b>Persogens</b>	<b>Tarefa</b>
Fritz 	Personagem principal se desloca para esquerda e para a direita e pula, também mata o inimigo com o pulo em cima da cabeça do mesmo.
Inimigos 	São os inimigos, tem o formato de bolas com energia que se movem para esquerda ou direita, e tiram vida do Fritz quando os toca.
Espinhos 	São Objetos estáticos que se o Fritz tocar, o mesmo morre
Vida 	São Objetos que aumentam as vidas do Fritz
Barra de Life 	É uma barra que mostra a vida atual de Fritz
Componentes que formam os endereços. 	São componentes que servem para auxílio para o Fritz passar de subfase, são eles que norteiam o jogo. Podem ser nº da tabela de paginas, nº de frame, deslocamento, nº. segmento, nº da base e limite
Portas 	Elas ficam espalhadas em cada subfase com uma numeração que serve de endereço. Quando o Fritz entra na porta certa ele vai para o lugar correto, porém quando ocorre o contrario ele morre.
Portais 	São as saídas de cada subfases.

### 3.2.3.7 Definição dos Cenários

A tabela 5 descreve todos os cenários do jogo.

TABELA 5 – DEFINIÇÕES DOS CENÁRIOS

Cenários	Descrição
<p data-bbox="234 629 376 663">Introdução</p> 	<p data-bbox="694 629 1129 696">O cenário tem um fundo preto, com um computador e Fritz.</p>
<p data-bbox="234 960 313 994">Fase 1</p> 	<p data-bbox="694 960 1129 1312">Essa fase é dividida em 3 subfases que aumentam o nível de dificuldade. Ela aborda o assunto MV Paginada. O ambiente é bem agradável com um desenho de um circuito no <i>background</i> e vários elevadores e outros objetos do jogo. Além disso, tem uma barra de vidas, os endereços virtuais e físicos e um botão para o tutorial.</p>
<p data-bbox="234 1319 323 1352">Fase 2</p> 	<p data-bbox="694 1319 1129 1727">Essa fase é dividida em 2 subfases que aumentam o nível de dificuldade, só que muito mais difícil que a fase 1. Ela aborda o assunto MV Segmentada. O ambiente é bem agradável com um desenho de um circuito no <i>background</i> e vários elevadores e outros objetos do jogo. Além disso, tem uma barra de vida, os endereços virtuais, físicos e um botão para o tutorial.</p>

### 3.2.3.9 Gameplay

Essa etapa foi uma das mais interessantes, onde foram definidas as coordenadas e comandos do jogo, pois os comandos no teclado foram definidos para a movimentação do Fritz, de todo jogo. O jogo também teria situações que precisavam clicar em botões, então foi

utilizado o mouse, foram definidas também todas as regras do jogo do mais simples a mais complexa.

### 3.2.3.9 Cronograma

A tabela 6 aborda o cronograma de todo o planejamento que serviu para o desenvolvimento do protótipo. As etapas desse cronograma foram executadas rigorosamente, para que o trabalho saísse conforme o esperado.

TABELA 6 – CRONOGRAMA

<b>Tempo</b>	<b>Tarefa</b>
Do mês 8/13 a 10/13	Análise do jogo
Do mês 10/13 a 11/13	Desenho de todos os <i>sprites</i> e <i>backgroud</i> .
Do mês 11/13 ao mês 12/13	Desenvolvimento da introdução do jogo
Do mês 12/13 a 2/14	Desenvolvimento de toda a primeira fase
Do mês 2/14 ao mês 3/14	Desenvolvimento de toda a segunda fase
Do mês 3/14 ao mês 4/14	Reajustes finais do jogo
Do mês 4/14 a 5/14	Fase de testes

## 3.3 DESENVOLVIMENTO DO PROTÓTIPO

Para o desenvolvimento do protótipo foi utilizado a *engine Construct 2*, que é um motor de jogo que trabalha com a tecnologia HTML 5 para games em 2d em conjunto com *JavaScript* e *CSS*. Logo foram identificados todos os requisitos do jogo e algumas etapas iniciais do *game design* que, foram iniciadas o processo de desenvolvimento.

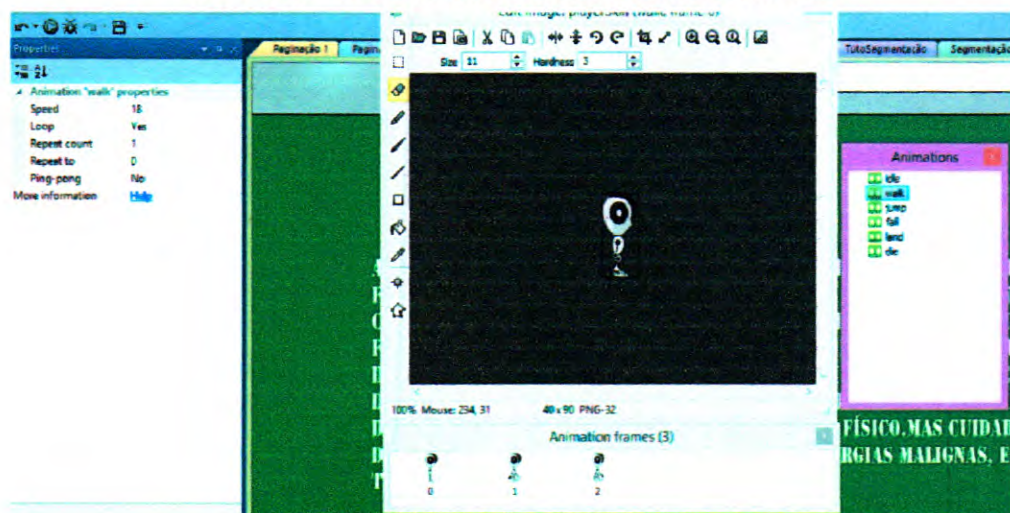
Depois que todos os objetos foram desenhados, iniciou-se o desenvolvimento da introdução do jogo e sua história.

### 3.3.1 Usando Construct 2

A figura 15 apresenta a criação de todos os frames do personagem principal, frames esses que são o personagem andando, pulando, parado, morrendo, e etc. Foram criados 28 *frames* só para o personagem principal.



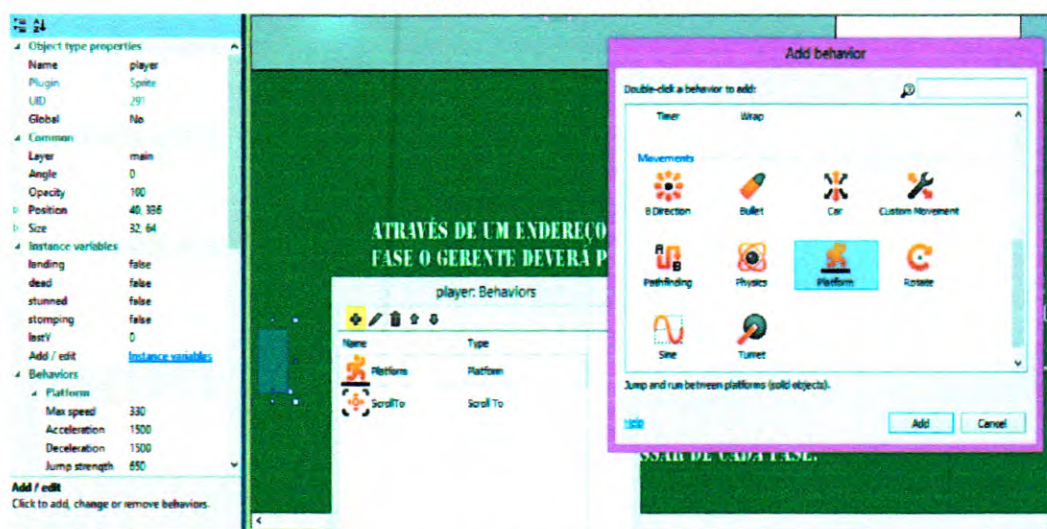
FIGURA15 – CRIANDO O PERSONAGEM PRINCIPAL



FONTE: DO AUTOR

A figura 16 aborda todos os atributos do personagem principal, como a gravidade que atua sobre ele, a largura, altura, e o tipo de comportamento do personagem. Definido o personagem sendo de plataforma, significa que ele vai ter pulo, então o mesmo terá propriedades das quatro coordenadas, ou seja, cima, baixo, direita e esquerda.

FIGURA16 – DEFININDO AS FUNÇÕES DO PERSONAGEM PRINCIPAL



FONTE: DO AUTOR

A listagem 1 apresenta uma parte do algoritmo gerado em *JavaScript* pela *engine*, que mostra como as propriedades direcionais são definidas. O personagem principal é movimentado de acordo com teclas, como são quatro direções, foram setadas quatro teclas.

## LISTAGEM 1 : FUNÇÃO KEYDOWN

```
//Esta função verifica as teclas pressionadas durante o jogo
```

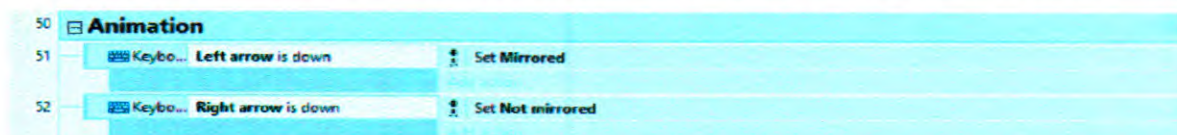
```
function KeyDown(evt){  
  switch (evt.keyCode) {  
    //seta para cima  
    case 38:  
      if (y - dy > 0){  
        y -= dy;  
      }  
      break;  
    //seta para baixo  
    case 40:  
      if (y + dy < HEIGHT){  
        y += dy;  
      }  
      break;  
    //seta para esquerda  
    case 37:  
      if (x - dx > 0){  
        x -= dx;  
      }  
      break;  
    //seta para direita  
    case 39:  
      if (x + dx < WIDTH){  
        x += dx;  
      }  
      break;  
  }  
}
```

FONTE: DO AUTOR

Para economizar os *sprite* durante o desenvolvimento, foram usadas algumas funções que deram grande ajuda. Por exemplo, quando o personagem vira à direita ou à esquerda, seria necessário construir mais alguns *sprites* para colocar na animação, assim

aumentando a carga de trabalho no desenvolvimento do game, mas a *engine* usada tem uma função chamada *mirrored*, que significa espelhamento, onde foi de grande ajuda. A figura 17 mostra como essa função foi usada, já na listagem 2 aborda como a função gerada.

**FIGURA 17 – USANDO A FUNÇÃO ESPELHO NOS LADOS DIREITO E ESQUERDO**



FONTE: DO AUTOR

**LISTAGEM 2: EFEITO ESPELHO**

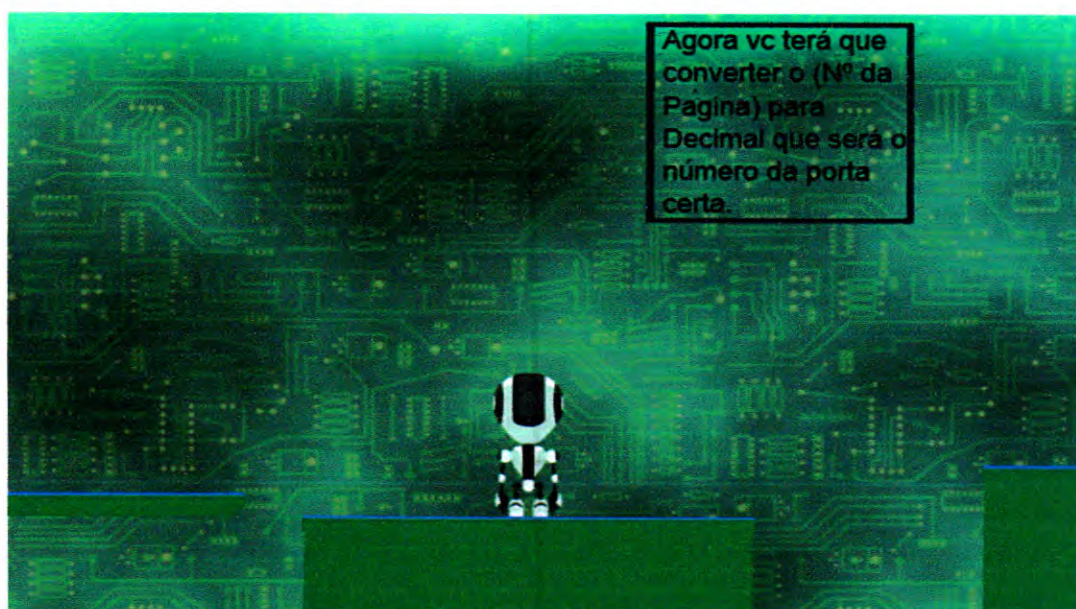
```
//efeito espelho
ctx.translate( largura + (x * 2), 0);
ctx.scale(-1, 1);
```

FONTE: DO AUTOR

### 3.3.2 Os testes para Validação

Depois que foi desenvolvido o protótipo em meados do mês de abril de 2014, o projeto foi para a fase de teste, que foi disponível para um pequeno grupo de alunos. Nos primeiros testes identificou-se a necessidade de mecanismos alta explicação no decorrer das fases. A figura 18 apresenta uma imagem do jogo que aparece um texto de auto-explicação.

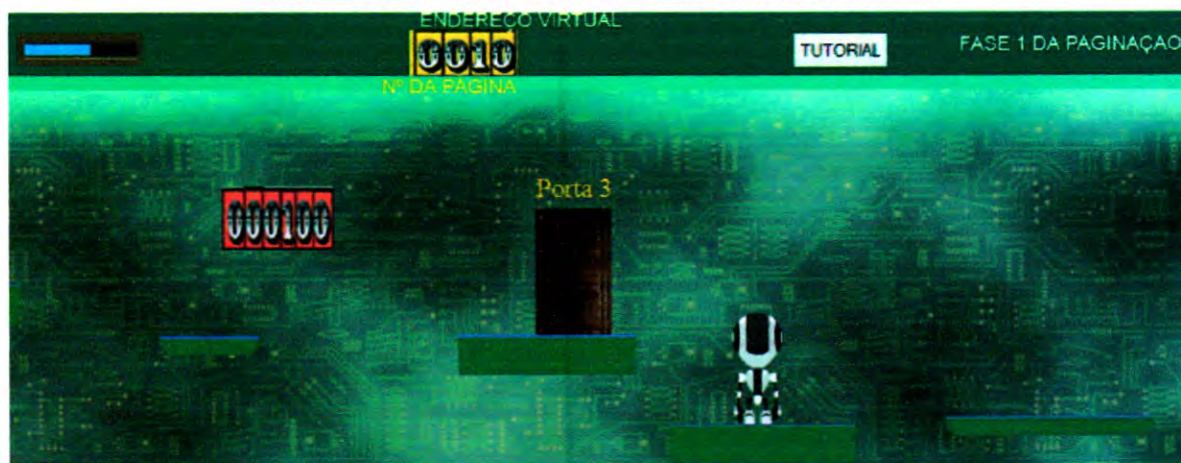
**FIGURA18 – MECANISMO DE AUTO-EXPLICAÇÃO DURANTE O JOGO**



FONTE DO AUTOR

Já na figura 19 é mostrado um botão que foi colocado em todas as fases para que se possa visualizar o tutorial, ou seja, todas as informações da fase. Antes só tinha um tutorial durante o início do jogo, ou seja, o jogador só visualizava uma vez, assim dificultando o seu desempenho nas fases.

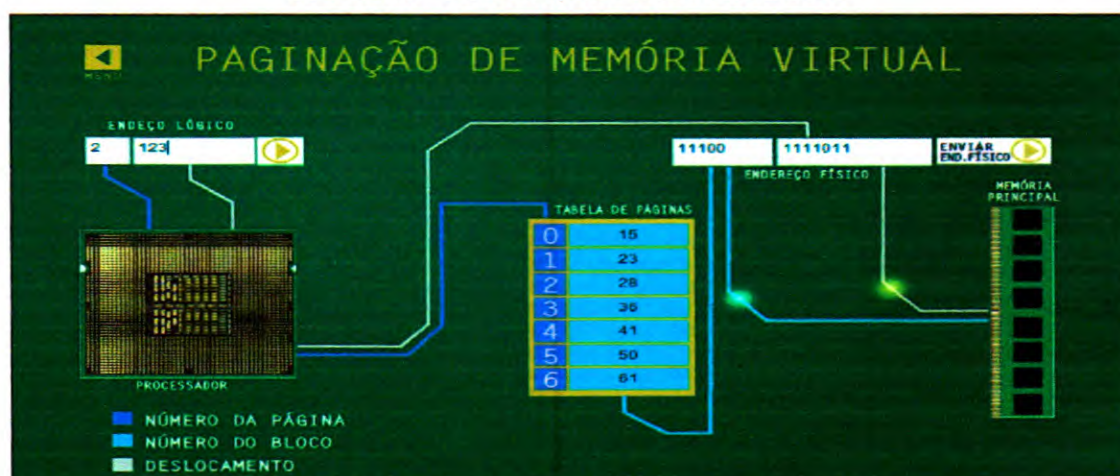
**FIGURA 19 – AS FASES COM O BOTÃO DO TUTORIAL**



**FONTE: DO AUTOR**

Durante a fase de testes também foi acoplado duas simulações para facilitar o entendimento das técnicas, que são MV paginada e MV segmentada. Com essa simulação o jogo facilita ainda mais, não só, a missão de zerar, mas também o entendimento claro e sucinto das técnicas. A figura 20 apresenta a imagem de uma das simulações durante o intervalo das fases. Essa imagem trata da técnica de MV por paginação, onde são carregados endereços lógicos, e depois são convertidos em endereços físicos.

**FIGURA20 – SIMULAÇÃO DA MV PAGINADA**



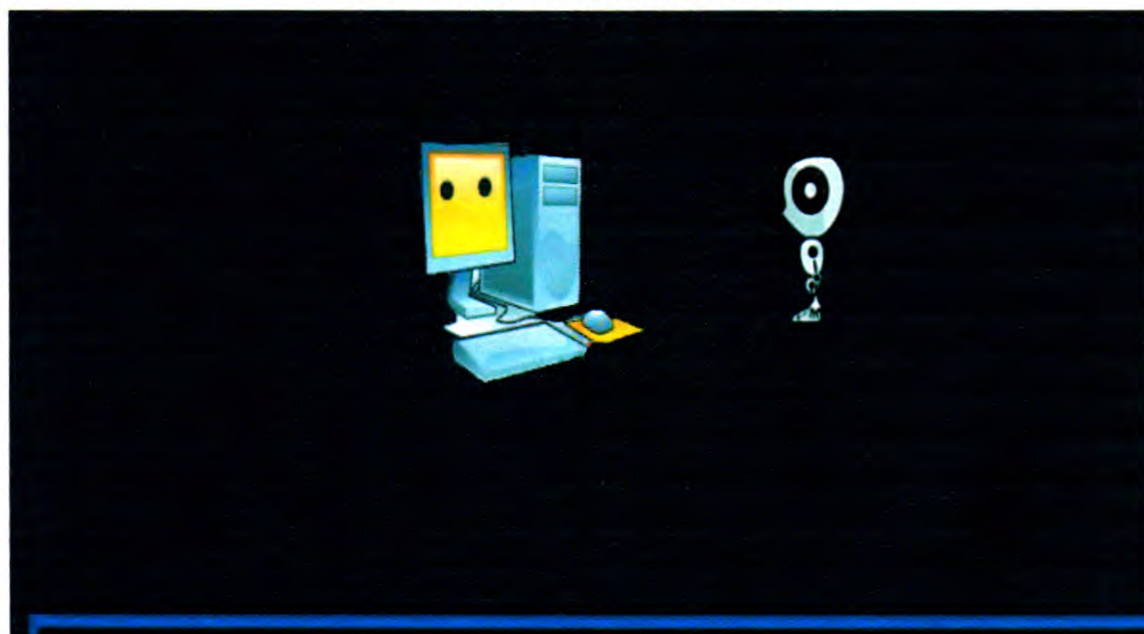
**FONTE: DO AUTOR**

#### 4 PROTÓTIPO DESENVOLVIDO

O protótipo desenvolvido se classifica na área de jogos educativos, que aborda o assunto MV. O jogo visa demonstrar o funcionamento da MV, ou seja, será visto todas as funcionalidades das MV Paginada e Segmentada. Foi proposto como uma solução para os problemas que norteiam este assunto, servindo como recurso didático para facilitar o aprendizado das técnicas mencionadas. O protótipo desenvolvido se destaca muito, além de ser um jogo educativo que é um dos motivos para estimular os alunos a entenderem o tópico.

O jogo se baseia em um enredo para que o jogador tente se envolver e entender as funcionalidades da técnica MV. O nome do personagem principal do jogo é Fritz, que é uma espécie de robô humanoide de cor cinza com preto. O protótipo se baseia em um diálogo na sua introdução entre o SO e o Fritz, onde o SO não está mais conseguindo gerenciar a memória virtual, então o mesmo pede ajuda ao personagem principal para ser o novo gerente. A figura 21 apresenta a imagem do diálogo na introdução.

FIGURA 21: INTRODUÇÃO DO JOGO

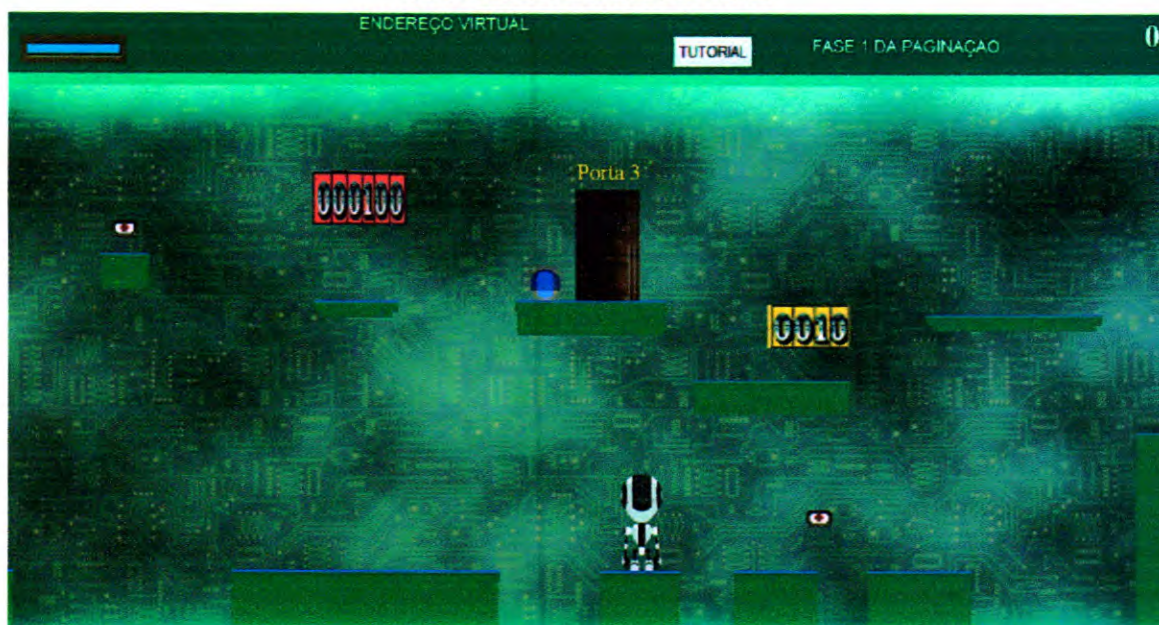


FONTE DO AUTOR

Depois do diálogo, o Fritz entra no SO para tentar executar a missão, que é dividida em duas fases uma para MV Paginada e outra MV Segmentada. A primeira fase é dividida em três subfases, cuja dificuldade vai aumentando conforme o progresso do jogador. Já a segunda fase é dividida em duas subfases do mesmo modo que as outras com nível de

dificuldade elevado sequencialmente. Na figura 22 está a primeira subfase da primeira fase que aborda o tema MV Paginada.

**FIGURA 22: PRIMEIRA FASE**



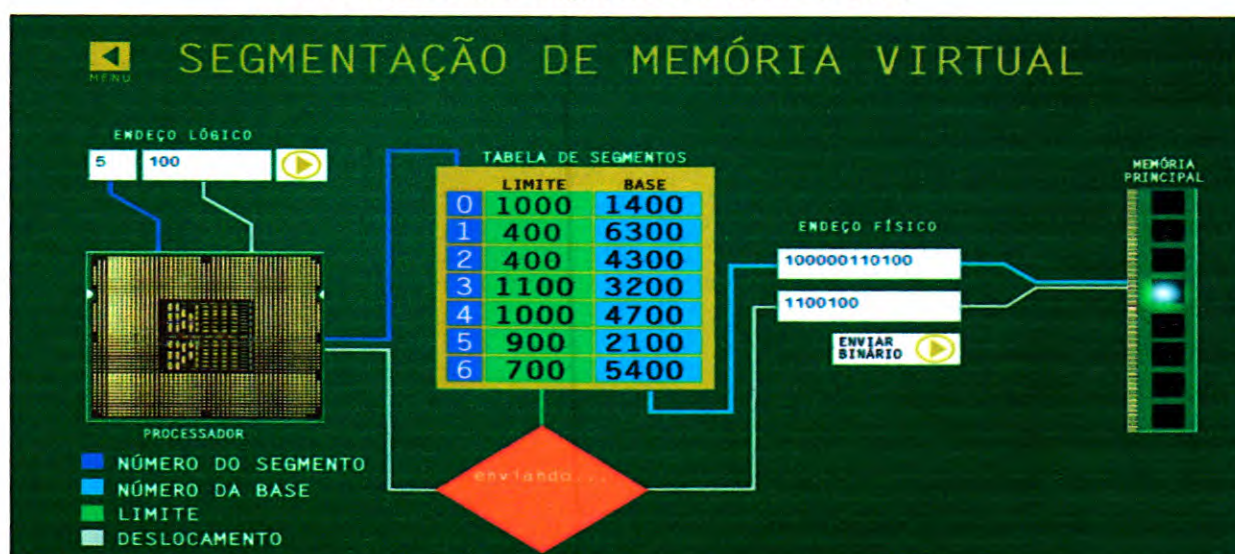
**FONTE: DO AUTOR**

Para o jogador passar da primeira fase tem que tentar identificar todos os objetos do jogo, pois alguns têm informações que o ajudam a concluir as missões. O objetivo da primeira fase é pegar o deslocamento que é um retângulo vermelho e está do lado direito de uma porta, e pegar o número da tabela de páginas que é um retângulo amarelo e está à esquerda da porta na figura 22, ainda na imagem da figura 22 tem o *life* que é a barra azul no canto direito superior da tela. Abaixo do nome endereço virtual no centro superior da imagem, ficará a combinação do deslocamento com o numero da tabela de páginas.

A primeira fase tem várias portas com numerações decimais, mas o Fritz tem que entrar na porta certa, onde é a que o levará para o lugar que está o endereço do *frame*, e forma o endereço físico. O objetivo principal da fase é depois que o endereço virtual foi formado, ou seja, a combinação do deslocamento com o número da tabela de páginas é tentar achar a porta certa através da informação do número da tabela de páginas, que está em binário. Para achar a porta certa, o jogador terá que saber converter de binário para decimal, pois se o mesmo entrar na porta errada morrerá, mas se caso entre na porta certa, achará o endereço de *frame* que será substituído com o número da tabela de paginas que faz uma nova combinação com o deslocamento, e forma o endereço físico, feito isso ficará visível um portal de saída, onde o jogador passará de subfase.

Durante as fases existem vários obstáculos, plataformas e elevadores, onde o jogador tem que ser muito astuto, além de raciocínio rápido. As fases possuem espinhos que tiram vidas, buracos, onde se o jogador cair perde a vida, e inimigos em forma de energia, que se os mesmos tocarem tirará vida do Fritz. Existe também um tempo em cada fase de 5 minutos, no qual se durante a subfase o tempo chegar em 5 minutos e o jogador não tiver cumprido a tarefa, o mesmo perderá. Ainda nas subfases tem retângulos brancos espalhados nas fases que são pequenas vidas que se o Fritz pegar aumentará o *life*.

FIGURA 23 : SUMULAÇÃO DA MV SEGMENTADA

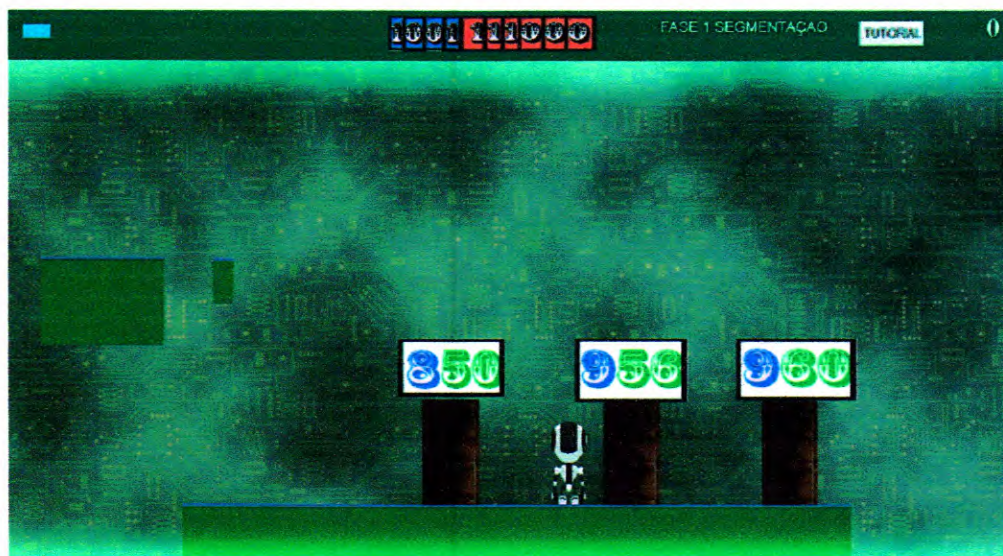


FONTE: DO AUTOR

Na segunda fase aborda-se o assunto MV Segmentada, que trata do mesmo objetivo da primeira fase que é formar o endereço físico, só que a técnica é um pouco diferente. Nesta fase a meta é formar o endereço de segmento, que é formado com o deslocamento e o número do segmento, todos representados por números binários. Já as portas da fase MV Segmentada são um pouco de diferença da fase MV Paginada, pois o endereço delas é formado por dois blocos decimais, um bloco representa o limite e o outro representa a referência do segmento. Depois que o endereço de segmento é formado, o jogador tem que procurar a porta certa, de tal forma, que todos os valores dos endereços devem ser convertidos de decimal para binário. Dessa maneira o número do segmento tem que ser o mesmo da referência do segmento da porta, e o deslocamento tem que ser menor que o limite desse segmento. Dessa forma a porta certa o levará para onde está o número da base que é outro componente que quando é junto com o deslocamento, forma o endereço físico, e assim o jogador passa de fase.

Na figura 24, as placas das portas são as referências dos segmentos representados na cor roxa, e ao limite de cor verde. A fase de MV Segmentada tem um grau de dificuldade bem elevado com relação à fase MV Paginada, pois existem situações em que algumas portas têm a mesma referência de segmento, nesse caso caberá ao jogador descobrir a porta certa.

**FIGURA 24 : FASE 2 DA MV SEGMENTADA**



**FONTE: DO AUTOR**



## 5. TESTES E RESULTADOS

Após a etapa de desenvolvimento foi realizado um teste para validar o protótipo, ou seja, verificar a utilização da ferramenta pelo público alvo em questão. Esta etapa consistiu na aplicação de um questionário misto envolvendo questões com resposta qualitativas, dicotômicas e subjetivas. O questionário encontra-se no apêndice A

As questões qualitativas tinham como alternativas os valores de 1 a 5, sendo 1 péssimo, 2 ruim, 3 regular, 4 bom e 5 ótimo, que foram aplicadas nas 3 primeiras perguntas do questionário.

A primeira questão abordava sobre o grau de utilização da ferramenta como recurso didático para o assunto MV. A segunda pergunta buscava saber sobre o nível de organização do conteúdo, fácil de entender e jogar. A terceira questão desse título tratava sobre a interface do jogo, ou seja, se tinha uma interface amigável e atraente.

Em seguida foram apresentadas quatro questões dicotômicas, ou seja, questões com apenas duas respostas. Tais questões que estão relacionadas à compreensão do assunto, a falta de algum assunto na temática, à facilidade de uso e se o aluno conhecia algum outro jogo educativo com a mesma abordagem.

Por fim, a última questão é subjetiva, na qual procura levantar sugestões sobre a próxima versão jogo.

Os avaliadores foram dez discentes da UESPI, campus Parnaíba do curso Bacharelado em Ciência da computação, todos estes já passaram pela disciplina.

A partir dos resultados apresentados no gráfico da figura 25 estima-se a primeira pergunta do questionário aplicado aos estudantes da disciplina, onde a pergunta era para saber qual a opinião sobre a utilidade do jogo. Uma pessoa marcou bom e 9 pessoas marcaram como ótimo.

O gráfico da figura 26 trata a questão da organização do jogo, onde 6 pessoas marcaram como bom, e 4 pessoas marcaram como ótimo a temática em questão.

Na figura 27 aborda a terceira pergunta que fala sobre a interface do jogo, questionando se é amigável. Nessa, somente 1 pessoa marcou como regular, 3 marcaram como bom, e 6 assinalaram como ótima.

Assim termina a sequência de questões qualitativas, ou seja, que tinham vários níveis para a avaliação do SE. Mas foi perceptível os resultados positivos nessas primeiras questões, pois não foram obtidos nenhum resultados abaixo de regular.

**FIGURA 25 : GRAFICO DA QUESTÃO SOBRE A UTILIDADE DO JOGO**



FONTE: DO AUTOR.

**FIGURA 26 : GRAFICO DA QUESTÃO SOBRE A ORGANIZAÇÃO DO JOGO**

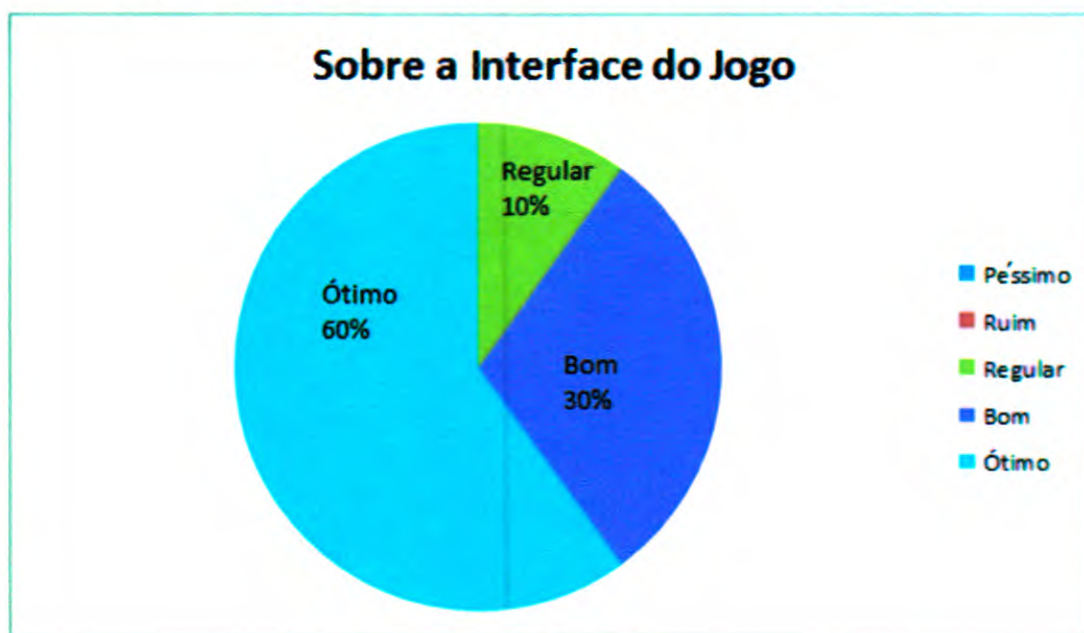


FONTE:DO AUTOR

A questão 4 do teste, perguntou ao aluno, se foi compreendido o assunto mencionado no jogo. Os resultados mostraram que todos os alunos marcaram sim, dando um resultado de cem por cento positivos. Na Questão 5 a pergunta era para saber se estava faltando algum assunto de MV no jogo, cem por cento marcaram não. Na questão 6

perguntava se o jogo facilitava o ensino de MV, cem por cento marcaram sim. Já na questão 7 perguntava se o aluno conhecia algum outro jogo com a mesma abordagem, cem por cento marcaram não.

**FIGURA 27 : GRAFICO DA QUESTÃO SOBRE A INTERFACE DO JOGO**



**FONTE:DO AUTOR**

Na última pergunta que é a de número 8, foi pedido sugestões para a segunda versão do jogo. Foram obtidas excelentes recomendações dentre elas, destacam-se: para a pergunta como melhorar a interface do jogo, as respostas são: colocar outros personagens principais para que o usuário possa escolher e colocar um ranking online no jogo para que os usuários possam competir. Outra recomendação é colocar mais dificuldade nas fases e deixar o jogo multiplayer. Já a última sugestão é inserir mais assuntos em relação gerência de memória.

Em vista dos resultados alcançados pode-se perceber que a ferramenta teve um retorno positivo quando testada por alunos. Assim sendo perceptível sua contribuição para disciplina de SO.

## 6 CONCLUSÃO

O presente trabalho demonstrou uma solução para minimizar as dificuldades encontradas no ensino do tópico MV. A proposta consiste no desenvolvimento de um protótipo de um jogo educativo, que foi motivado pela escassez de recursos didáticos similares, o que torna relevante por se tratar de uma contribuição inédita, além do mais o grau de complexidade do conteúdo do assunto, que tende a dificultar a compreensão clara ou precisa da teoria da temática.

Apesar de não ter se obtido sucesso nas pesquisas sobre jogos educativos de memória virtual, foram encontrados outros softwares educativos que abordam o mesmo tópico, sendo estes nas categorias de simuladores e tutoriais. Estas ferramentas encontradas foram de extrema importância para o desenvolvimento deste trabalho, que ajudaram a suprir as deficiências encontradas.

Todas as dificuldades no assunto foram buscadas minuciosamente, pensada como resolvê-las colocando no jogo. Mas mesmo algumas ferramentas tendo esses assuntos complexos como conteúdo, ainda deixavam a desejar.

O protótipo do jogo foi desenvolvido com a *engine Construct 2*, que usa a linguagem HTML 5. Essa *engine* trabalha com jogos 2d. A escolha dessa ferramenta se deu por causa do grande crescimento da linguagem HTML 5, que além de ser leve, tem vários recursos que ajudam muito no desenvolvimento. Foi utilizada também, a metodologia *Game Design*, que foi usada para fazer o planejamento de todo o projeto.

Através dessas ferramentas foi possível criar um jogo estilo plataforma (2d), que possibilita a representação do conteúdo de maneira gráfica permitindo uma visualização mais atraente de memória virtual, minimizando o problema das abstrações. Além do mais, é portátil, sendo possível execução em diferentes plataformas.

Durante o desenvolvimento do protótipo foram realizados diversos testes funcionais que serviram para encontrar e corrigir alguns erros. Além do mais ainda nesta fase alguns alunos também colaboraram no desenvolvimento do jogo, através de sugestões que foram acatadas e contribuíram para uma melhor qualidade da ferramenta.

Após a etapa de desenvolvimento concluída da primeira versão do protótipo, foi realizado o teste de validação da ferramenta, que teve como objetivo verificar se os objetivos aqui propostos foram alcançados. Para tanto foi aplicado um questionário entre alguns alunos do Curso de Computação da UESPI campus Parnaíba.

Os resultados obtidos nesta fase foram satisfatórios permitindo confirmar a eficiência do jogo. Além do mais os avaliadores apresentaram algumas sugestões que serão analisadas e poderão ser inseridas nas próximas versões da ferramenta como proposta de trabalhos futuros.

Por fim pode-se concluir que esta é uma ferramenta importante para o auxílio do ensino de assunto MV, uma vez que proporciona facilidades na compreensão do assunto de forma atraente e motivadora.

## 6.1 TRABALHOS FUTUROS

Ficam como propostas algumas melhorias tais como colocar recursos de áudio durante os mecanismos auto-explicativos e acrescentar mais fases.

Além dessas recomendações, sugere-se um projeto envolvendo o mesmo assunto, MV paginada e segmentada utilizando realidade virtual. Podem-se usar tecnologias atuais como óculos de realidade virtual, para fazer com que o jogador seja o protagonista real do jogo, e que possa se envolver de forma direta. Existem diversas recomendações para aplicar nesse jogo, pode-se até mesmo desenvolver um jogo em 3d para melhorar a jogabilidade, e fica como sugestão de desenvolvimento a *engine Blender*.

## REFERÊNCIAS

ALMEIDA, Fernando José; VALENTE, José Armando. **Visão Analítica da Informática na Educação no Brasil: A Questão da Formação do Professor**. 2005 Disponível em: <<http://www.proinfo.gov.br>>. Acesso em: 12.05.2014

ARTIGO SQL MAGAZINE. **PROJETO DE SOFTWARE UTILIZANDO UML**. Disponível em : <<http://www.devmedia.com.br/artigo-sql-magazine-13-projeto-de-software-utilizando-uml/5640>>

AZEVEDO, EDUARDO et al. **Desenvolvimento de Jogos 3D e Aplicações em Realidade Virtual**. Rio de Janeiro: Elsevier, 2005. 319 p.

COUTINHO, MURILO. **Virtual Memory Simulation**. Disponível em: <<http://www.isi.edu/mass/MuriloHomePage/Java/midterm402/midterm2.html>>. Acessado em: 12.05.2014

CURTI, M. M. **Conceitos e Tecnologias no Desenvolvimento de Jogos Eletrônicos**. 2006. 84 p. Trabalho de Conclusão do Curso (Sistemas de Informação) – UNIFEV, Centro Universitário de Votuporanga, Votuporanga, 2006.

DENNING, Peter (1997). **Before Memory Was Virtual**. In the Beginning: Recollections of Software Pioneers. Disponível em : <<http://cs.gmu.edu/cne/pjd/PUBS/bvm.pdf>> acessado em: 13.05.2014.

EMILY YURIKO KIOKI.; PATRICK PIAZZA SANTIAGO. **Simulador de Sistema Operacional Genérico (SSOG)**. Revista INICIA, Santa Rita do Sapucaí, MG, n. 8, p. 41-48, 2008.

F. B. MACHADO AND L. P. MAIA, **Operating Systems Architecture**. 3rd ed., Brazil, LTC, 2002.

GADELHA, Renê N. S. et al. **OS Simulator: Um Simulador de Sistema de Arquivos para Apoiar o Ensino/Aprendizagem de Sistemas Operacionais**. In: XXI Simpósio Brasileiro de Informática na Educação, João Pessoa, PB. 2010.

GARMPIS, ARISTOGIANNIS. **Design and Development of a Web-based Interactive Software Tool for Teaching Operating Systems**. Journal of Information Technology Education Volume 10, 2011.

JAVASCRIPT.ABOUT. Disponível em: <<http://javascript.about.com/od/reference/p/javascript.htm>> Acessado em : 01.06.2014

JOHNSON, LAURENCE F., et al **O RELATÓRIO HORIZON 2010: A K12 Edição** Austin, TX.: The New Media Consortium, 2010 < <http://wp.nmc.org/horizon-k12-2010/chapters/game-baseado-em-aprendizagem/#0> >

KOMOSINSKI, Leandro José. **Um novo significado para a educação tecnológica fundamentado na informática como artefato mediador da aprendizagem**. 2000. 146 f. Tese de Doutorado (Doutor em Engenharia da Produção) - Programa de Pós- Graduação em

Engenharia de Produção da Universidade Federal de Santa Catarina. Universidade Federal de Santa Catarina, Florianópolis.

LOPES, Aline Barbosa<sup>1</sup>; QUEIROZ, Glauber Vanesco Forni; SACCA, Juliana. **SIM MEMORY**. Revista Científica de Psicologia, 2008.

LOPES, A. R.; SOUZA, D. A. ; CARVALHO, J. R. B. ; SILVA, W. O. ; SOUSA, V. L. P. **SIME: Memory Simulator for the Teaching of Operating Systems**. In: SIIE 2012: Simposio Internacional de Informática Educativa, 2012, Andorra La Vella. SHE 2012 Actas del XIV Simposio Internacional de Informática Educativa. Andorra la Vella: La Salle Open University, 2012. v. 14°. p. 283-287.

MAIA, P. L. **SOSim: Simulador para o Ensino de Sistemas Operacionais**. Dissertação (Mestrado em Computação) – Universidade Federal do Rio de Janeiro, NCE/UFRJ, 2001. Disponível em: <<http://www.training.com.br/sosim>>. Acessado em: 11.05.2014

MORAIS, R. **SOFTWARE EDUCACIONAL: A IMPORTÂNCIA DE SUA AVALIAÇÃO E DO SEU USO NAS SALAS DE AULA**. 2003. Disponível em :< <http://www.flf.edu.br/revista-flf/monografias-computacao/monografia-rommel-xenofonte.pdf>> Acesso em : 13.05.2014.

MORITZ, GUSTAVO. **Simulador de Gerência de Memória - Simulador De Mecanismos De Gerência De Memória Real E Virtual**. Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau, 2004.

ONTKO, RAY; REEDER, ALEXANDER. **MOSS – Modern Operating System Simulators**. Disponível em:< <http://www.ontko.com/moss/>>. Acessado em: 11.05.2014

REIS, F. P.; JÚNIOR, P. A. P.; COSTA, H. A. X. **TBC-SO/ WEB: Um Software Educacional para o Ensino de Políticas de Escalonamento de Processos e de Alocação de Memória em Sistemas Operacionais**. In: XX Simpósio Brasileiro de Informática na Educação, Florianópolis, SC. 2009.

SCIRRA. Disponível em:< <https://www.scirra.com/construct2>> acessado em: 15.05.2014.

SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas operacionais: conceitos e aplicações**. Rio de Janeiro: Campus, 2004.

SILBERSCHATZ, GALVIN GAGNE. **Fundamentos de Sistemas Operacionais**. [S.l.:s.n.], 2002.

**SITE OFICIAL DA UML**. Disponível em: <<http://www.uml.org/>> Acessado em: 15.05.2014.

SOFFA, M.M.; ALCÂNTARA, P.R de C. **O Uso Do Software Educativo: Reflexões Da Prática Docente Na Sala Informatizada**. Disponível em: <[www.pucpr.Br/eventos/educere/educere 2008 /anais/pdf/335\\_357.pdf](http://www.pucpr.Br/eventos/educere/educere%202008/anais/pdf/335_357.pdf)> acesso em 13.05.2014

PERUCIA, A. S. et al. **Desenvolvimento de Jogos Eletrônicos: Teoria e Prática**.

São Paulo: Novatec, 2005. 320 p.

TYSON, Jeff. **Como a 3DO cria videogames**. 2007. Disponível em: <<http://informatica.hsw.uol.com.br/como-a-3do-cria-videogames.htm>>. Acesso em: 13.05.2014

VILLA, O., SCHAUMONT, P., VERBAUWHEDE, I., MONCHIERO, M., AND PALERMO, G. **Fast Dynamic Memory Integration in Co-Simulation Frameworks for Multiprocessor System on-Chip**, In Proceedings of the Conference on Design, Automation and Test in Europe - Volume 2 (March 07 - 11, 2005).

WIDMER, S, WODNIOK,D, GOESELE ,M,WEBER,N. **Fast Dynamic Memory - Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05), 2013** . Disponível em:<[http://www.gris.informatik.tu-darmstadt.de/~mgoesele/download/Widmer\\_2013\\_FDM.pdf](http://www.gris.informatik.tu-darmstadt.de/~mgoesele/download/Widmer_2013_FDM.pdf)> Acessado em: 11.05.2014.

**W3SCHOOL/HTML5**. Disponível em: <[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)> Acessado em 15.05.2014.

**W3SCHOOL/CANVAS**. Disponível em: <[http://www.w3schools.com/html/html5\\_canvas.asp](http://www.w3schools.com/html/html5_canvas.asp)> Acessado em: 01.06.2014.

**W3SCHOOL/CSS3**. Disponível em: < [http://www.w3schools.com/css/css3\\_intro.asp](http://www.w3schools.com/css/css3_intro.asp)> Acessado em: 05.06.2014.



**APÊNDICE A- Questionário de avaliação do protótipo****Questionário de Avaliação do Protótipo do Jogo NOME:**

1) Que nota você daria sobre a utilidade do Jogo?

1       2       3       4       5

2) Que nota você daria sobre a organização do jogo (fácil de entender e jogar)?

1       2       3       4       5

3) Dê a nota sobre a interface do Jogo (se é amigável)?

1       2       3       4       5

4) Você aprendeu / esclareceu uma dúvida / entendeu melhor o funcionamento de algum assunto abordado pelo Jogo?

Sim       Não

5) Você acha que falta algum assunto no Jogo em relação à Memória Virtual? Qual?

Sim       Não

---

---

---

6) Você acha que o Jogo facilita a aprendizagem de Memória Virtual?

Sim       Não

7) Você conhece outro *Jogo* Educativo de Memória Virtual? Qual?

Sim       Não

---

---

---

8) Que sugestões você daria para a segunda versão do Jogo?

---

---

---