
Visão e Escopo



FACULDADE FARIAS BRITO

Bacharelado em Ciência da Computação

TCC I

PROJETO:

**UM OBJETO DE APRENDIZAGEM PARA O ENSINO DE SISTEMAS
OPERACIONAIS**

Nome do aluno: Reginaldo Silva Barros

Orientador: Sérgio Yunes

FORTALEZA-CE

2013



Histórico de Alterações

Data	Versão	Descrição	Autor
17/09/2013	1.0.0	Justificativa / Motivação	Reginaldo Barros
14/10/2013	1.0.0	Introdução	Reginaldo Barros
19/10/2013	1.0.0	Metodologia	Reginaldo Barros
21/10/2013	1.0.0	Objetivos	Reginaldo Barros
12/11/2013	1.0.0	Referencial Teórico	Reginaldo Barros
23/11/2013	1.0.0	Correções	Reginaldo Barros



Conteúdo

1. INTRODUÇÃO.....	4
2.MOTIVAÇÃO/JUSTIFICATIVA.....	5
3.OBJETIVOS.....	6
3.1OBJETIVOS ESPECÍFICOS.....	7
4.REVISÃO BIBLIOGRÁFICA.....	7
4.1OBJETO DE APRENDIZAGEM.....	7
4.1.1EXEMPLOS DE OBJETOS DE APRENDIZAGEM.....	8
CAIXA DE CÂMBIO 1.....	8
MUDANÇAS AMBIENTAIS GLOBAIS.....	9
SIMULADOR DE AUTÔMATOS.....	10
4.2SISTEMAS OPERACIONAIS: CONCEPÇÕES GERAIS.....	11
4.2.1PROCESSO.....	12
4.2.2ESPAÇO DE ENDEREÇAMENTO.....	12
4.2.3GERENCIAMENTO DE MEMÓRIA.....	13
4.2.3.1MEMÓRIA VIRTUAL.....	14
4.3OBJETOS DE APRENDIZAGEM EM SISTEMAS OPERACIONAIS.....	16
4.3.1OSIM	16
4.3.2SIMULARSO.....	17
4.3.3BRASILOS.....	18
4.3.4EPSOSIM.....	19
4.3.5CONCEITOS DE PROGRAMAÇÃO.....	21
4.3.5.1JAVASCRIPT.....	21
4.3.5.2SINGLE PAGE APPLICATION.....	22
5.METODOLOGIA.....	22
6.REFERÊNCIAS BIBLIOGRÁFICAS.....	23

1. INTRODUÇÃO

Segundo o MEC, Objetos de Aprendizagem (OAs) são recursos educacionais, em diversos formatos e linguagens, que tem por objetivo mediar e qualificar o processo de ensino-aprendizagem. Podemos, ainda, verificar em (FREIRE, 1977) que o processo de aprendizagem de algo por alguém, quer seja autonomamente ou com auxílio de outra pessoa, guarda relação direta com o grau de consciência – por parte de quem quer ensinar e também de quem quer aprender - do processo de aprendizagem. Desta forma, o uso de um OA que apoie o aluno no seu intuito de desenvolver o aprendizado, proporciona um enriquecimento do processo e gera uma melhor assimilação e conscientização do problema estudado.

Nesse contexto, diversos OAs foram construídos com o objetivo de contribuir para o aprendizado em todos os níveis da educação. Já na educação primária, o aluno aprende através de ferramentas lúdicas, brincadeiras e imagens, em complemento às experiências passíveis de serem realizados em sala de aula. Agora, com o uso da informática na educação, outros OAs puderam ser construídos de modo a fornecer um ótimo ambiente de interação, dinâmico e atrativo, transmitindo o conhecimento teórico de uma maneira menos formal.

O uso de software educativo no ensino de Computação tem recebido bastante atenção nos últimos anos, principalmente quando focam em temas de recorrente dificuldade dos alunos. Trabalhos como GraphOS (CAÑAS, 1987) e SOsim (MAIA, 2001), por exemplo, destacam-se por apresentar boas simulações de um sistema operacional. Todavia, enquanto esses trabalhos tratam com maior atenção o gerenciamento de processos, acabam deixando uma lacuna no que diz respeito à gerência de memória.

Ante o cenário apresentado, observa-se que há a necessidade de uma ferramenta que simule o gerenciamento de memória de forma a complementar as já existentes e oferecer uma maior riqueza de detalhes do assunto, porém, sem muitas configurações que pertençam a outro tema. Assim, este trabalho propõe a criação de um Objeto de Aprendizagem que contemple os detalhes da Memória Virtual de um sistema operacional, sem desconsiderar a gerência de processo no que for necessário para compreender o contexto da disciplina.

A organização deste trabalho dar-se-á em três capítulos. O capítulo I discorre sobre o conceito de objeto de aprendizagem e seu uso na educação e também mostra alguns exemplos de ferramentas existentes em diversas áreas. No capítulo II, serão expostos os conceitos de sistemas operacionais necessários para compreender o tema a ser tratado na construção de um objeto de aprendizagem. E no capítulo III, serão mostrados exemplos de objetos de aprendizagem da área de sistemas operacionais e conceitos referentes às tecnologias a serem utilizadas no desenvolvimento da ferramenta.

2. MOTIVAÇÃO/JUSTIFICATIVA

A disciplina de sistemas operacionais tem o propósito de explicar o funcionamento do software mais importante de um ambiente computacional – o sistema operacional - os problemas que devem ser por ele resolvidos e os desafios de sua implementação. Basicamente, o sistema operacional controla todos os componentes físicos e fornece abstração para que aplicações de usuários possam utilizá-los. Entender isso é importante porque o profissional de Tecnologia da Informação precisa saber lidar com problemas intrínsecos a esses sistemas e precisa ser capaz de desenvolver soluções que os atenuem.

Um dos objetivos importantes de um sistema operacional é permitir que vários programas de usuário sejam executados ao mesmo tempo – compartilhando recursos de *hardware*, tais como: Unidade Central de Processamento (CPU), memória e periféricos - de forma que uma aplicação não interfira na outra e nem se transforme num gargalo ao desempenho do sistema. Ou seja, a execução de vários programas deve ser transparente na perspectiva de cada processo.

Diante deste panorama, o ensino de sistemas operacionais passa por vários assuntos teóricos que explicam a resolução destes problemas. Dentre os temas que abordam a concepção desse objetivo, destaca-se o gerenciamento de memória, o qual inclui a Memória Virtual como uma solução ao problema da limitação da Memória de Acesso Randômico (RAM).

Por ser uma disciplina com enfoque na maioria das vezes teórico, o repasse desse conhecimento ao aluno, muitas vezes, o põe numa situação de dificuldade, quando este não

consegue a abstração necessária para compreendê-lo. Embora o professor possa utilizar palavras e exemplos do dia-a-dia, o aluno pode sentir-se distante do que lhe é mostrado e assim não desenvolver interesse no entendimento, conseqüentemente, assimilando pouco ou nenhum conhecimento. O uso de ilustrações é um recurso comum e ajuda. Porém, quando essas ilustrações são interativas, na forma de aplicativos, o usuário tende a manter melhor atenção naquilo que lhe é exposto, porque de certa forma, ele interage com aquela realidade e se põe a raciocinar para entender e utilizar o aplicativo. Desta forma, este trabalho objetiva desenvolver um Objeto de Aprendizagem que auxilie no ensino do assunto Memória Virtual, mais especificamente os algoritmos de substituição de páginas, que compõe o conceito base da paginação - forma comum de implementação de Memória Virtual.

O interesse no assunto parte da necessidade de entender, de forma aprofundada, como o computador trabalha e gerencia os seus recursos de forma inteligente. Possibilitando, dessa forma, sua utilização por usuários leigos e gerando um mecanismo de comunicação eficiente, na forma de cliques de mouse e uso do teclado. Essa experiência ajudará a resolver problemas de programas que possam vir a ser desenvolvidos e que muitas vezes passam por problemas similares ao de um sistema operacional que, em última instância, também é um programa.

Desta feita, o Objeto de Aprendizagem proposto pretende contribuir de duas formas: para os alunos, como ferramenta de simulação dos ambientes computacionais expostos nos livros textos, e para os professores como uma forma de aumentar sua produtividade na sala de aula. Isso tende a resultar numa melhor absorção do conteúdo e, conseqüentemente, melhor formação educacional.

3. OBJETIVOS

Construir um objeto de aprendizagem na forma de um simulador de gerenciamento de memória que demonstre através de recursos gráficos o funcionamento dos diversos algoritmos de substituição de páginas, e que permita, através de parametrizações, a simulação de diversos cenários por meio de parâmetros, tais como: número de processos na memória, política de alocação de páginas adotada, política de busca de páginas e o escopo da substituição de páginas.

3.1 OBJETIVOS ESPECÍFICOS

- a) Fazer uma pesquisa sobre a importância dos objetos de aprendizagem no processo de ensino aprendizagem;
- b) Fazer um levantamento dos objetos de aprendizagem existentes no âmbito da disciplina sistemas operacionais;
- c) Definir o escopo e os requisitos do software de simulação de gerenciamento de memória a ser desenvolvido de acordo com os principais aspectos da técnica de memória virtual;
- d) Criar interface web que permita configurar e visualizar o funcionamento das páginas de memória com animações gráficas;
- e) Codificar o objeto de aprendizagem do tipo ferramenta de simulação com base no resultado do item c.

4. REVISÃO BIBLIOGRÁFICA

O objetivo deste item é fornecer uma revisão da bibliografia que aborda os temas relacionados a esta pesquisa, dentre eles, o conceito e uso de Objetos de Aprendizagem, as ferramentas hoje existentes de auxílio ao ensino de sistemas operacionais e os conceitos utilizados na construção de um OA.

4.1 OBJETO DE APRENDIZAGEM

Segundo o Institute of Electrical and Eletronics Engineers (IEEE, 2000), o objeto de aprendizagem é definido como qualquer entidade, que pode ser utilizada, reutilizada ou referenciada durante o aprendizado apoiado por computador. Ele pode conter simples elementos como um texto ou um vídeo. Ou ainda, ele pode ser um hipertexto, um curso ou até mesmo uma animação com áudio e recursos mais complexos. De acordo com Singh (2001), um Objeto de Aprendizagem deve ser estruturado e dividido em três partes bem definidas, pois essa estrutura diferencia o recurso de outras tecnologias aplicadas à Educação e possibilita a produção de conhecimento. O objetivo informa ao aluno o que pode ser aprendido através do uso do objeto; o conteúdo instrucional prevê as ferramentas necessárias

para que, no final, o objetivo possa ser atingido; a prática (*feedback*) é a oportunidade do aluno de registrar a sua experiência com o objeto e confirmar se houve conhecimento.

Várias características norteiam a produção de um OA, como acessibilidade, autoconsistência, durabilidade e reusabilidade (entre outros). Sendo esta última a mais importante, pois permite seu uso em diferentes ambientes de aprendizagem. Desta forma, vários repositórios são criados para centralizar e organizar as ferramentas e permitir sua busca por grau de instrução, disciplina e nível de dificuldade. No Brasil, o BIOE – Banco Internacional de Objetos Educacionais (disponível em <http://objetoseducacionais2.mec.gov.br>), criado pela Secretaria de Educação a Distância (SEED/MEC), é um exemplo prático disso.

4.1.1 EXEMPLOS DE OBJETOS DE APRENDIZAGEM

A seguir apresentamos alguns objetos de aprendizagem disponíveis para outras áreas do ensino e que foram disponibilizadas na internet.

CAIXA DE CÂMBIO 1

O Objeto de Aprendizagem Caixa de câmbio 1 tem como objetivo fazer o estudante compreender e visualizar como é o funcionamento de um carro e de sua caixa de câmbio. Trata-se de uma animação/simulação desenvolvida em Flash que explica como é possível um carro andar, devido às marchas, motor, embreagem e outras peças. Em seguida, pode-se ver como todas essas peças estão organizadas dentro de um carro.

Conforme figura 1, é possível verificar a interface do aplicativo e a facilidade de controle para avançar e retroceder nas lições.



Figura 1 – Demonstração das peças que permitem o funcionamento de um carro. Disponível em <<http://objetoseducacionais2.mec.gov.br/handle/mec/8245>>, acessada em 11/11/2013.

MUDANÇAS AMBIENTAIS GLOBAIS

Outro exemplo é este Objeto de Aprendizagem que objetiva sensibilizar professores e alunos sobre os impactos do aquecimento global nas atividades humanas e ecossistema e que medidas são necessárias para diminuir esses efeitos.

A animação (Figura 2) em Flash permite ativar/desativar tanto o áudio como a legenda. Também é possível avançar facilmente no conteúdo do vídeo. É direcionado praticamente a todos os níveis de ensino, conforme está descrito no endereço <http://objetoseducacionais2.mec.gov.br/handle/mec/1978>.



Figura 2 – Explicação sobre as mudanças ambientais globais. Disponível em: <http://objetoseducacionais2.mec.gov.br/handle/mec/1978>, acessada em 12/11/2013.

Muitas outras ferramentas de simulação e apoio ao ensino-aprendizagem, como estas, podem ser facilmente encontradas no repositório de objetos de aprendizagem do MEC, site <http://objetoseducacionais2.mec.gov.br/>.

SIMULADOR DE AUTÔMATOS

O Simulador de Autômatos é um programa simulador que permite a criação, simulação e conversão de modelos formais, desenvolvido com o objetivo de auxiliar o aprendizado de Linguagens Formais e Autômatos (JUNIOR, on-line). Trata-se de uma ferramenta que pode ser baixada e executada no computador e que, ao abrir, permite o usuário desenhar um modelo através de figuras já conhecidas dos alunos da disciplina. Ou seja, a mesma linguagem utilizada pelo professor em seus exemplos. Na figura 3, podemos ver um exemplo simples da construção de um autômato.

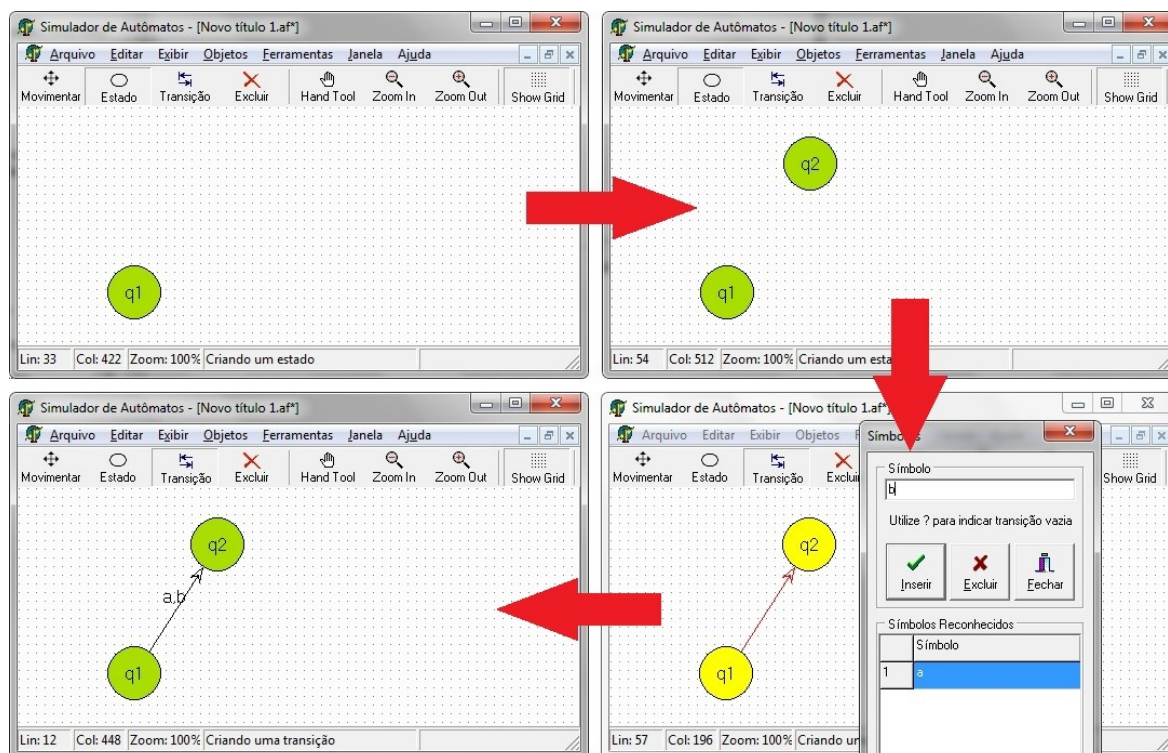


Figura 3: Exemplo de criação de um autômato simples. Disponível em <
<http://www.simuladordeautomatos.com/imagens>>, acesso em 24/11/2013.

4.2 SISTEMAS OPERACIONAIS: CONCEPÇÕES GERAIS

De acordo com Deitel (2005), sistema operacional é um software que habilita as aplicações a interagir com o hardware de um computador. Sendo assim, o programador passa a escrever programas não para um hardware específico, mas para uma abstração computacional que é fornecida pelo sistema operacional. Detalhes intrínsecos a uma determinada arquitetura pode ser ignorada e o trabalho de desenvolvimento passa a ser otimizado para o problema a ser solucionado pelo programa.

Um programa de computador (*software*), por sua vez, é uma sequência de comandos que instruem a máquina a fazer algo, e estes comandos estão escritos em uma linguagem capaz de ser entendida pela máquina (VILARIM, 2004). Isso significa que tanto o sistema operacional quanto as aplicações de usuários são programas e, portanto, são linhas de códigos que ordenam o computador a realizar tarefas, a diferença está no papel de cada um.

Alguns conceitos e abstrações são necessários para compreender como o sistema operacional funciona. A seguir, apresentamos a definição de elementos fundamentais nessa compreensão.

4.2.1 PROCESSO

Define-se um processo como um programa em execução (TANENBAUM, 2009), ou seja, um programa de computador carregado na memória que compartilha a Unidade Central de Processamento (CPU) com outros processos e realiza o seu trabalho. A cada processo é atribuído um espaço de endereçamento, que corresponde a um intervalo de posições na memória no qual o processo pode ler e escrever. Nesse espaço está contido o programa executável, os dados do programa e sua pilha. Além disso, há outras informações associadas a cada processo, como, por exemplo, uma lista dos arquivos abertos pelo mesmo.

“Em muitos sistemas operacionais, todas as informações relativas a um processo – que não sejam o conteúdo de seu próprio espaço de endereçamento – são armazenadas em uma tabela do sistema operacional denominada **tabela de processos**, que é um arranjo (ou lista encadeada) de estruturas, uma para cada processo existente” (TANENBAUM, 2009. Pag. 23).

4.2.2 ESPAÇO DE ENDEREÇAMENTO

Em sistemas operacionais modernos, múltiplos programas podem compartilhar a memória paralelamente. Para que um processo não interfira na execução de outro, é necessário um mecanismo de proteção, uma forma de impedir que um processo invada o espaço de endereços de outro processo. Muito embora essa proteção seja implementada no hardware, ela é controlada pelo sistema operacional.

No gerenciamento do espaço de endereçamento, cada processo possui um conjunto de endereços que pode utilizar, geralmente iniciando do 0 até um valor máximo. No caso mais simples, a quantidade máxima de espaço de endereçamento que um processo tem é menor que a memória principal (TANENBAUM, 2009). Porém, quando o programa precisa de um espaço de endereçamento superior à própria memória, uma técnica chamada de Memória Virtual é utilizada para resolver o problema da escassez de memória. Discutiremos

posteriormente os detalhes dessa técnica, uma vez que é na sua implementação/funcionamento que se baseia este trabalho.

4.2.3 GERENCIAMENTO DE MEMÓRIA

O gerenciamento de memória é o componente do sistema operacional que gerencia a memória primária, também chamada de Memória de Acesso Randômico (RAM). Seu papel é alocar memória para os processos criados e desalocar no seu término, bem como manter o controle de uso da memória. Devido ao custo muito alto de sua fabricação, foi criada uma hierarquia de memórias que é formada por memórias cache muito rápidas e a memória principal, ambas voláteis, uma vez que os seus dados são perdidos quando falta energia. O sistema operacional trata de abstrair esses recursos e fornecer aos programas um modelo mais simplificado.

Quando não há uma abstração da memória, geralmente cada programa é executado por vez, devido ao simples fato de que cada programa iria assumir que é dono de toda a memória disponível e facilmente corromperia os dados um do outro.

Nesse modelo de trabalho, a organização da memória poderia se dar de várias maneiras, conforme figura 9. Na opção a, o programa do usuário fica na parte superior da memória e o sistema operacional na inferior. Já na opção b, o sistema operacional fica na parte superior em Memória de Somente Leitura (ROM) e o programa logo abaixo. Na opção c, apenas os drivers ficam em Memória de Somente Leitura (ROM), enquanto o sistema operacional fica na parte inferior e no meio, o programa de usuário.

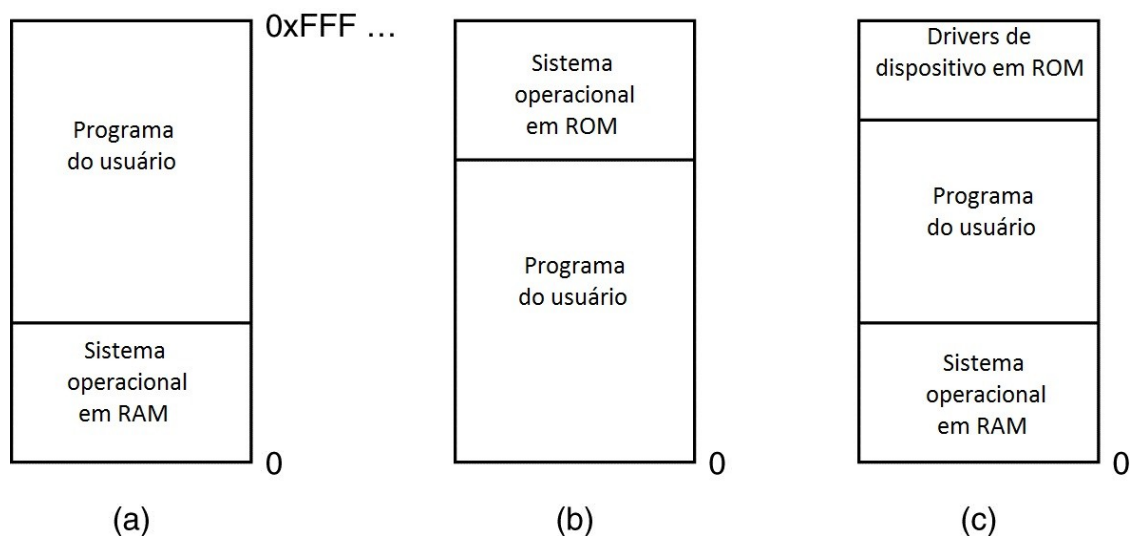


Figura 9: Organização da memória (TANENBAUM, 2009).

Embora esses modelos possam ter sido usados no passado, a necessidade de executar múltiplos programas simultaneamente nos sistemas operacionais modernos torna-os muito dispendiosos e difícil de gerenciar. Portanto, tornou-se necessário criar uma abstração de memória chamada espaço de endereçamento, definida anteriormente.

4.2.3.1 MEMÓRIA VIRTUAL

Segundo Tanenbaum (2009), a ideia por trás da memória virtual é que cada programa tem seu próprio espaço de endereçamento, que é dividido em blocos chamados páginas. Assim, antes do programa ser carregado na memória, ele é dividido em páginas, e somente algumas páginas são carregadas. Quando um endereço referenciado pelo programa está em uma página que não foi carregada na memória, o sistema operacional é responsável por carregar a página necessária e executar novamente a instrução.

A implementação mais comum dessa abstração da memória, na qual cada programa dispõe de uma memória conceitualmente infinita, é a paginação. Nela, os endereços referenciados pelos programas são denominados endereços virtuais e constituem o espaço de endereçamento virtual (TANENBAUM, 2009). Um hardware denominado Unidade de Gerenciamento de Memória (MMU) é responsável por traduzir endereços virtuais para endereços que existem na memória física.

Cada página de endereço virtual pode ser carregada numa região da memória chamada moldura de página (*page frame*). O mapeamento é gerenciado pela Unidade de Gerenciamento de Memória (MMU) através de uma tabela de páginas. Sempre que a Unidade Central de Processamento (CPU) precisa acessar um endereço de memória, ela passa o endereço virtual para a Unidade de Gerenciamento de Memória (MMU) que usa a tabela de páginas para traduzir para o endereço real da memória. Se o endereço referenciado não estiver carregado em nenhuma moldura (*frame*), o controle é passado para o sistema operacional, para que ele possa carregar a página que contém o endereço. Esse evento é chamado de falta de página (*page fault*).

A tabela de páginas possui várias entradas com as informações necessárias para realizar a tradução de endereços virtuais em endereços físicos. Na figura 10 podemos ver a estrutura de uma entrada da tabela. Além das informações de mapeamento em si, também é utilizado bits para informar se a página foi referenciada e/ou modificada, determinar os tipos de acesso permitido, se o cache está habilitado e se a página está atualmente carregada na memória.

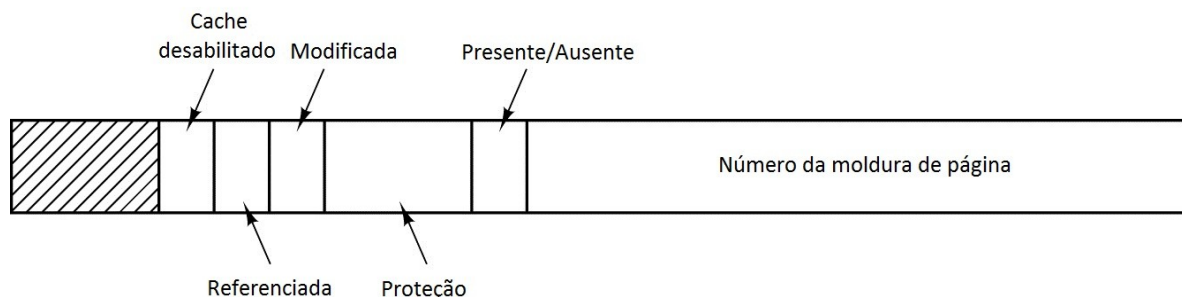


Figura 10: Entrada da tabela de páginas (TANENBAUM, 2009).

Quando uma página referenciada não está carregada na memória, conforme já dito anteriormente, o controle é passado para o sistema operacional que precisa decidir qual página deve sair da memória e ceder o espaço para a nova página. Se a página a ser removida tiver sido modificada enquanto estava na memória, ela deverá ser reescrita no disco com o propósito de atualizar a cópia virtual lá existente (TANENBAUM, 2009).

A escolha da página a ser retirada da memória pode ser crucial para o desempenho do sistema, por isso, existem vários algoritmos que podem ser utilizados para que essa escolha não seja aleatória. Quando avaliamos uma estratégia de substituição de página normalmente a

comparamos com a estratégia de substituição de página ótima, a qual determina que, para obtermos desempenho ótimo, devemos substituir a página que não será referenciada novamente no futuro mais distante possível (DEITEL, 2005).

4.3 OBJETOS DE APRENDIZAGEM EM SISTEMAS OPERACIONAIS

Dentro da área de sistemas operacionais, muitos objetos de aprendizagem do tipo ferramentas de simulação foram desenvolvidos para demonstrar o funcionamento de determinadas características de um sistema operacional. A seguir, serão apresentados alguns deles.

4.3.1 SOSIM

O SOsim é um simulador de sistemas operacionais que foi desenvolvido por Luiz Paulo Maia. Ele permite que o professor apresente os conceitos e mecanismos de um sistema operacional multiprogramável e/ou multitarefa, como Unix, OpenVMS e Windows (MAIA, 2001). Os principais temas abordados pelo simulador são Gerência de Processos, Gerência de Processador e Gerência de Memória Virtual. Apesar de ser uma ferramenta bem completa, podemos verificar que a Gerência de Memória não explora todos os detalhes e sua visualização não é tão intuitiva já que os quadrados que representam as páginas são bem pequenos e isso pode ficar confuso quando estamos simulando diversos processos.

A ferramenta foi desenvolvida com o intuito de ser gratuita e está disponível na página <http://www.training.com.br/sosim/>.

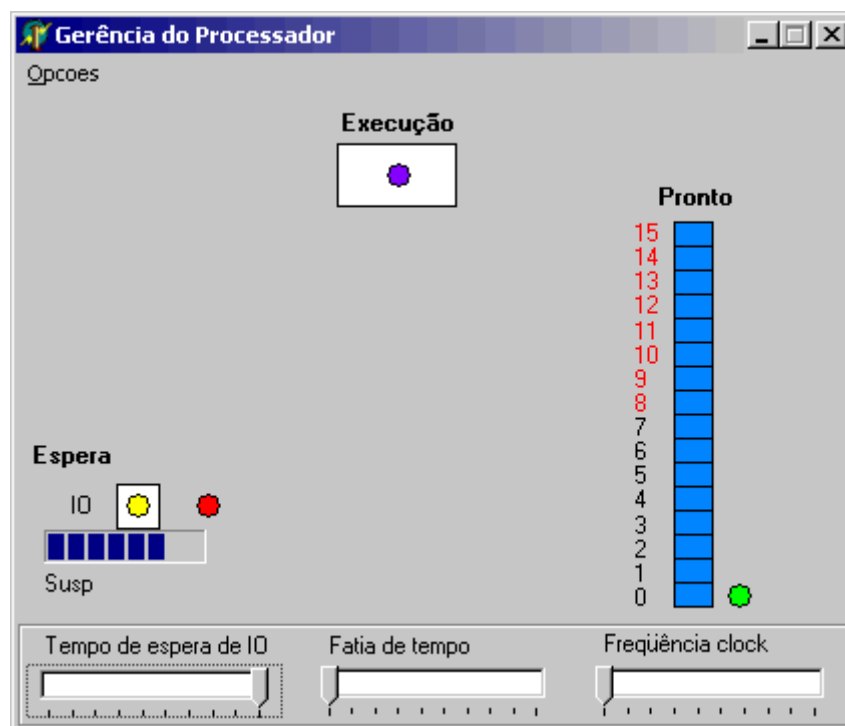


Figura 4 – Gerência de Processador no simulador SOsim. Disponível em <<http://www.training.com.br/sosim>>, acesso em 16/11/2013.

4.3.2 SIMULARSO

O Simulador de Recursos de sistemas operacionais (SimulaRSO) é um aplicativo que simula o comportamento dos principais recursos que são implementados em um sistema operacional moderno (RODRIGUES, PEREIRA, 2011). Permite visualizar graficamente o Escalonamento de Processos, o Escalonamento de Disco e a Paginação de Memória. Não requer nenhum tipo de instalação, podendo ser visualizado direto no browser sem quaisquer requerimentos de instalação. No entanto, o computador precisa estar conectado à internet. É importante frisar que nos testes realizados, o site mostrou-se lento para processar as requisições. Está disponível em <http://www.simula-rso.appspot.com/>.

Simulação de Escalonamento de Processos

Algoritmo: SJF

Processo	Burst CPU	Tempo de Espera	Tempo de Resposta	Turn Around
1	10 ms	0 ms	10 ms	10 ms
2	10 ms	10 ms	20 ms	20 ms
3	10 ms	20 ms	30 ms	30 ms
4	10 ms	30 ms	40 ms	40 ms
5	10 ms	40 ms	50 ms	50 ms
Total de processos	Tempo de Execução	Tempo de Espera Médio	Tempo de Resposta Médio	Turn Around Médio
5	50 ms	20.0 ms	30.0 ms	30.0 ms

Simulação Gráfica: SJF

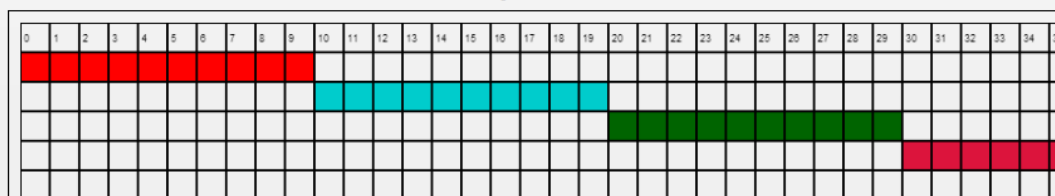


Figura 5 – SimulaRSO: Simulação de Escalonamento de Processos. Disponível em <<http://www.simula-rso.appspot.com>>, acessado em 16/11/2013.

4.3.3 BRASILOS

Segundo (SILVA, FILHO, QUEIROZ, MENEZES, 2011), a ferramenta BrasilOS se apresenta essencial para compreender gerenciamento de processos, pois considera várias características do processo, como: tamanho, tipo, prioridade, tempo de CPU, tempo que permanece bloqueado e tempo de alternância entre processos. Desse modo, ele se detém exclusivamente na gerência de processos, que é um dos temas mais importantes no estudo de sistemas operacionais.

Conforme podemos ver na Figura 6, a interface é bem intuitiva e colorida, facilitando a identificação dos diferentes processos. Também possui botões de ajuda para explicar a teoria por trás do simulador.

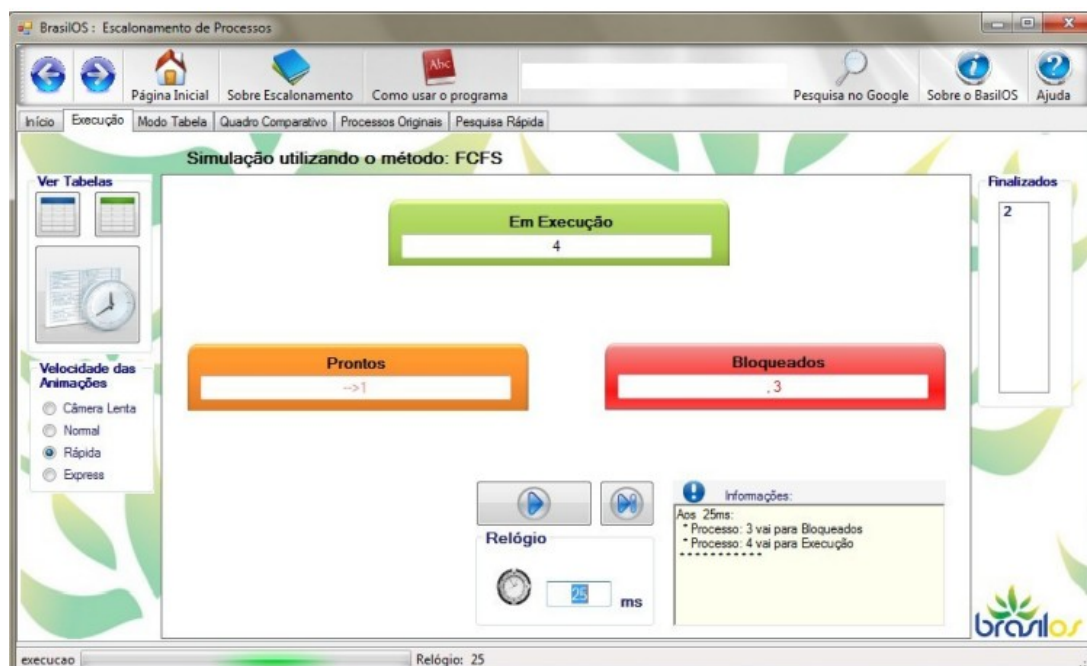


Figura 6 – BrasilSO: Execução de Simulação em Modo Gráfico. Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2011/sexoestec/art2049.pdf>>, acessado em 10/10/2013.

Este simulador também permite a comparação automática dos resultados da simulação de diferentes algoritmos, permitindo o usuário entender a eficiência de um e a ineficiência de outro algoritmo diante do grupo de processos criados. Foi desenvolvido utilizando a linguagem C# e trata-se de uma aplicação genuinamente *Desktop*.

4.3.4 EPSOSIM

O EPSOsim é um simulador de gerenciamento de processos, permite a configuração detalhada do sistema. Conforme visualizado na figura 7, é possível determinar o tempo que cada processo poderá utilizar o processador (*timeslice*) e o tempo que o sistema esperará por tarefas de acesso a dispositivos. Já na figura 8, é possível definir o tipo de processo que será simulado, uma cor para melhor identificá-lo e seu tempo de processador. A partir de então, o simulador exibe graficamente o funcionamento dos algoritmos de escalonamento.

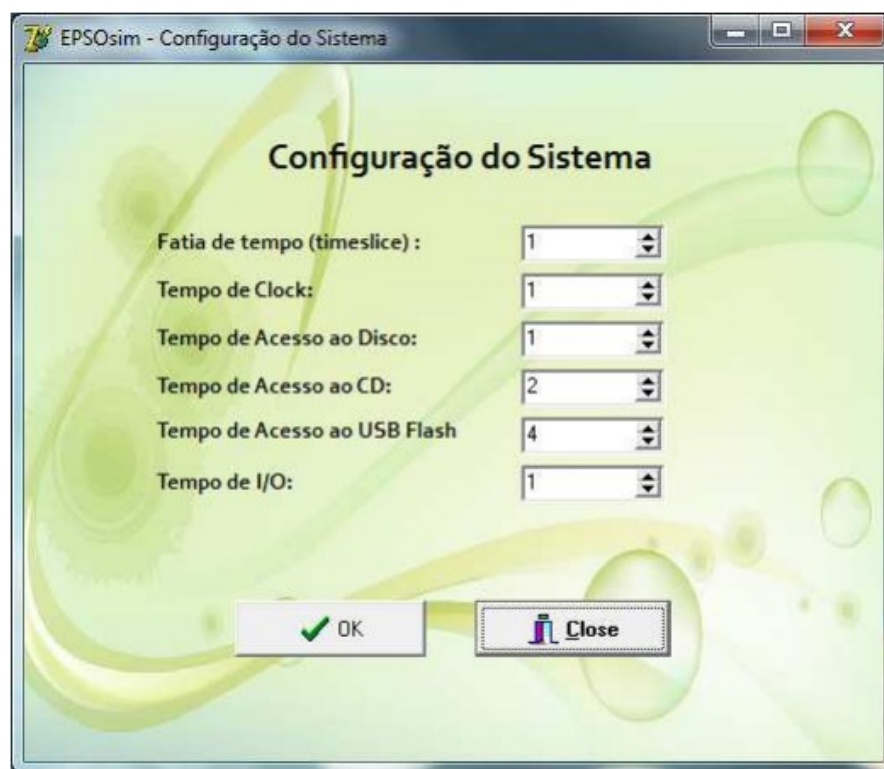


Figura 7 – EPSOsim: Configurações do Sistema. Disponível em: <<http://www.ffb.edu.br/sites/default/files/tcc-20111-joel-oliveira-carvalho.pdf>>, acesso em 16/11/2013.



Figura 8 – EPSOsim: Janela de Criação de um Processo. Disponível em: <<http://www.ffb.edu.br/sites/default/files/tcc-20111-joel-oliveira-carvalho.pdf>>, Acesso em 16/11/2013.

O simulador possui oito algoritmos de escalonamento e permite a visualização do Diagrama de Grantt, o qual é uma das partes fundamentais do mesmo, segundo Carvalho (2011).

4.3.5 CONCEITOS DE PROGRAMAÇÃO

As linguagens de programação permeiam o mundo da computação e se destacam por suas facilidades frente a um determinado tipo de problema. No mundo da internet, por exemplo, várias ferramentas e linguagens se juntam para trazer ao usuário uma experiência de navegação cada vez melhor. Nesse contexto, pretende-se usar ferramentas e padrões estabelecidos para o desenvolvimento do objeto de aprendizagem. Portanto, faremos uma descrição das tecnologias e métodos relacionados.

4.3.5.1 JAVASCRIPT

O *Javascript* é uma linguagem de programação que foi criada por Brendan Eich e lançada em 1995 como componente do navegador de internet Netscape. Dentre suas características está o suporte a orientação a objetos, a programação funcional, a tipagem fraca, linguagem dinâmica e funções de primeira classe.

Com o avanço das tecnologias de internet e a demanda por aplicações web robustas e responsivas, a linguagem ganhou força e hoje é amplamente suportada em diversas plataformas, seja no computador, em tablets ou nos telefones inteligentes (*smartphone*). Segundo Crockford (2008), o *javascript* é uma linguagem importante porque é a de navegadores de internet.

O grande diferencial do *Javascript* é que é uma linguagem que roda predominantemente no lado do cliente (*client-side*), isto é, dentro do navegador do usuário. Embora existam outras tecnologias capaz de executar programas dentro de uma página web - como os applets Java e o Flash - o fato do interpretador está incorporado ao navegador torna esse modelo mais independente e seguro.

Com o uso do *Javascript* as aplicações web evoluíram e hoje é possível aproximar a usabilidade de uma página na internet com aquela que o usuário está acostumado em seu computador – aplicações desktop.

4.3.5.2 SINGLE PAGE APPLICATION

No ambiente tradicional da WEB, o navegador do cliente solicita a um servidor remoto o conteúdo do site escolhido e mostra na tela o resultado de sua interpretação do texto recebido. Esse texto, basicamente, está escrito usando a Linguagem de Marcação de Hipertexto (HTML), a qual o navegador reconhece e é capaz de interpretar suas diretivas.

Para navegar entre as partes do mesmo sítio (*site*), o usuário clica em links que ordena o navegador a fazer uma nova requisição ao servidor remoto. Neste momento, o navegador deixa de exibir a página atual e aguarda o retorno de sua nova solicitação para poder mostrar na tela.

Com o crescimento da complexidade das aplicações WEB, tecnologias como *javascript* passaram a ser usadas para deixar a página mais dinâmica e interativa. O avanço dessa tecnologia fez surgir um novo modelo de desenvolvimento de aplicação WEB, o *Single Page Application* (SPA) que, segundo (MIKOWSKI, POWEL, 2013), entrega uma aplicação desktop no browser. Ou seja, é possível criar uma aplicação WEB com uma alta experiência de usuário – sem esperas ao clicar num *link* e sem ter a página limpa enquanto uma ação de clique retorna do servidor.

Ainda segundo o mesmo autor, as vantagens de um SPA são: a não necessidade de um *plugin* instalado no navegador, a leveza da aplicação, a diminuição de complexidade devido ao uso de uma única linguagem e a maior fluidez e interatividade da página. Com base nisso, desenvolver um objeto de aprendizagem que usufrua dessas vantagens, o torna mais atrativo para o professor, por poder executar o simulador sem esperas de requisições e para o aluno que se diverte com a usabilidade.

5. METODOLOGIA

Para viabilizar a realização deste trabalho será feito uma pesquisa bibliográfica para definição e identificação de Objetos de Aprendizagem existentes que colaboram para qualquer área do ensino. Em seguida, será dado um foco maior nos Objetos de Aprendizagem existentes na área de sistemas operacionais. Para tanto, será utilizado, principalmente, ferramentas de pesquisa na internet, como o Google, Google Acadêmico, para seleção de

artigos, livros, teses, etc., e o site webeduc do MEC. O acervo físico e virtual (site da ACM, IEEE e outros) da biblioteca da Faculdade Farias Brito também será considerado para esclarecer os conceitos de sistemas operacionais que serão abordados.

Em um segundo momento, será feita uma análise comparativa dos principais Objetos de Aprendizagem encontrados que envolvem o ensino de sistemas operacionais, apontando seus pontos positivos e negativos, a partir de uma análise descritivo-explicativa. A partir desses pontos, será definido a arquitetura e o escopo do objeto a ser proposto de modo que esse venha a complementá-los.

Na sequência, será definido o ambiente de desenvolvimento a ser utilizado que, a priori, pretende-se que seja WEB, utilizando a Linguagem de Programação *Javascript* de forma a aumentar o poder de alcance da ferramenta. Isto é, qualquer pessoa poderá utilizá-la em qualquer lugar do mundo sem qualquer processo de instalação.

Por fim, o trabalho de desenvolvimento seguirá o paradigma de Orientação a Objetos, que é amplamente utilizado pelo mercado, pela sua manutenibilidade e capacidade de expansão.

6. REFERÊNCIAS BIBLIOGRÁFICAS

CAÑAS, D. A. **Graphos: a graphic operating system**. ACM SIGCSE, Bulletin, Volume 19 (1) 201 – 205, 1987.

CARVALHO, J. O. **EPSOSIM - SIMULADOR DE ESCALONAMENTO DE PROCESSOS EM SISTEMAS OPERACIONAIS**. Monografia, FFB, 2011. Disponível em: < <http://www.ffb.edu.br/sites/default/files/tcc-20111-joel-oliveira-carvalho.pdf>> Acesso em 16 de Novembro de 2013.

CROCKFORD, Douglas, em: **O melhor do Javascript**. Rio de Janeiro: Editora Alta Books. 2008.

DEITEL, H. M, em: **sistemas operacionais**. 3. Ed. São Paulo: Editora Pearson Prentice Hall, 2005.

FREIRE, P., **Extensão ou comunicação? O Mundo Hoje**. Editora Paz e Terra, Rio de Janeiro, v. 24, 10ª edição.

IEEE Learning Technology Standards Committee (LTSC) (2000) “**Draft Standard for Learning Object Metadata**”, Institute of Electrical and Electronics Engineers, Inc. LTSC. (2000). Learning technology standards committee website. Disponível em: <http://ltsc.ieee.org/> Acesso em: 20 de julho de 2007.

JUNIOR, N. G. R. **Simulador de Autômatos**. Universidade de Uberaba, on-line. Disponível em < <http://www.simuladordeautomatos.com> > acesso em 24 de Novembro de 2013.

MAIA, L. P. **SOsim: Simulador para o Ensino de sistemas operacionais**. Tese de Mestrado, NCE/UFRJ, mar. 2001. Disponível em: <<http://www.training.com.br/sosim>> Acesso em 16 de Novembro de 2013.

MIKOWSKY, MICHAEL S. POWELL, JOSH C, em: **Single Page Application: Javascript end to end**. Editora Manning. 2013.

RODRIGUES, A. A, PEREIRA, C. R. **SimulaRSO: Simulador de Recursos de sistemas operacionais**. Universidade Católica de Santos, 2011. Disponível em: <<http://www.simula-rso.appspot.com>> Acesso em 16 de Novembro de 2013.

SILVA, M. A. M, FILHO, R. G. D, QUIROZ, E. F, SILVA, H. C, MENEZES, J. W. M. **DESENVOLVIMENTO E VALIDAÇÃO DE UM OBJETO DE APRENDIZAGEM PARA ENSINO DE SISTEMAS OPERACIONAIS: O CASO BRASILOS**. COBENGE, 2011. Disponível em: <



<http://www.abenge.org.br/CobengeAnteriores/2011/sextoestec/art2049.pdf> > Acesso em 10 de Outubro de 2013.

SINGN, Harvi. **Introduction to Learning Objects**. 2001. Disponível em: www.elearningforum.com/july2001/singh.ppt.2001. Acesso em: 5 jul. 2009.

TANENBAUM, Andrew, em: **sistemas operacionais Modernos**. 3. ed. São Paulo: Editora Pearson Prentice Hall. 2009.

VILARIM, G. **Algoritmos: Programação Para Iniciantes**. 2. Ed. Rio de Janeiro: Editora Ciência Moderna Ltda. 2004.