

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**FAGNER MARCELLO UGGIONI**

**A VIRTUALIZAÇÃO NO AUXÍLIO À ETAPA DE TESTE DE  
COMPATIBILIDADE DE SOFTWARES**

**CRICIÚMA, JULHO DE 2008**

**FAGNER MARCELLO UGGIONI**

**A VIRTUALIZAÇÃO NO AUXÍLIO À ETAPA DE TESTE DE  
COMPATIBILIDADE DE SOFTWARES**

Trabalho de Conclusão de Curso para a Obtenção  
do Grau de Bacharel em Ciência da Computação  
da Universidade do Extremo Sul Catarinense.

Orientador: Prof. MSc. Paulo João Martins

**CRICIÚMA, JULHO DE 2008**

Dedico este trabalho, a todos os que direta ou indiretamente colaboraram para sua conclusão, em especial há duas pessoas que vêm me acompanhando em cada passo de minha caminhada, estes são meus pais Aládio e Zuleide que estiveram sempre prontos para ajudar, orientar e acolher.

## **AGRADECIMENTOS**

A DEUS porque me deu o dom da vida e me abençoou com o saber.

A minha família, em especial meus pais Aládio e Zuleide, que me deram à vida, e me guiaram para o caminho do bem, chorando comigo em minhas dificuldades e me animando nas horas de desânimo, pois eles fizeram dos seus passos o meu passo quando eu queria desistir e sorriam com minhas vitórias.

A minha namorada Jadna Graci, pelo incentivo, companheirismo e muita paciência durante todo o tempo.

A empresa Market Automações Ltda, por ter oferecido condições para que eu pudesse concretizar o período acadêmico, em especial a Bernardo Efiting que como proprietário soube compreender a necessidade de algumas ausências.

Ao Prof. MSc. Paulo Martins, orientador desta pesquisa, pelo seu precioso tempo dedicado, pela sua orientação em vários momentos que não sabia qual rumo tomar e por ter acreditado na possibilidade de realização do mesmo não medindo esforços no momento de auxiliar para a conclusão deste trabalho.

E a todos os professores, funcionários e colegas da instituição UNESC, que de um modo geral contribuíram nessa caminhada, certo de que esta etapa final jamais se tornará um adeus, mais sim um até breve, se Deus quiser.

Não existe vento favorável para aquele que não sabe para onde vai.

(Arthur Schopenhauer)

## RESUMO

No dia a dia das organizações desenvolvedoras de softwares a etapa de teste de compatibilidade de software é uma constante, pois a evolução das tecnologias e dos sistemas operacionais se dá a passos largos. A quantidade de equipamentos e softwares necessários para a realização de testes de compatibilidade com as novas tecnologias se torna uma barreira para as organizações. Esta pesquisa emprega conceitos da Engenharia de Software, como a etapa de testes de compatibilidade de softwares, em conjunto com a virtualização, encontrando uma alternativa para as organizações realizarem os testes de compatibilidade de software nos mais variados ambientes computacionais. Esta adesão se dará com a demonstração de utilização de uma ferramenta de virtualização de ambientes computacionais, realizando testes com softwares em diferentes ambientes virtuais.

Além da utilização de virtualização em conjunto com a fase de teste de compatibilidade de software, esta pesquisa propõe um estudo de caso para manter softwares legados em funcionamento com o surgimento de novas tecnologias.

Palavras-chave: Engenharia de Software, Teste de Software, Virtualização, Software legado.

## ABSTRACT

On every day of developing organizations the stage of software compatibility test is a constant, because the evolution in technologies and operating systems takes wide steps. The amount of equipment and softwares needed to perform these compatibility tests with the new technology becomes a barrier for organizations. This research applies the concepts of Software Engineering, as the stage of tests of software compatibility, along with virtualization, finding an alternative for organizations to conduct the tests of software compatibility in a wide variety of computing environments. This alliance will happen with the demonstration of the use of a virtualization tool for computing environments, running tests on software in virtual some environments. In addition to this suffrage, this research suggests a case study to maintain legacy software in operation along with the incoming of new technologies.

Keywords: Software Engineering, Virtualization, software legacy, Software Test.

## LISTA DE ILUSTRAÇÕES

Figura 1 Ciclo de vida clássico de um software.....	20
Figura 2 Monitor de máquinas virtuais de tipo I.....	38
Figura 3 Monitor de máquinas virtuais de tipo II.....	38
Figura 4 Ambiente de teste virtualizado.....	45
Figura 5 Windows 98 virtualizado sobre o Windows XP.....	50
Figura 6 Erro no teste com ambiente Windows 98 virtualizado.....	50
Figura 7 Teste com ambiente Windows XP virtualizado.....	51
Figura 8 Erro no teste com ambiente Windows Vista virtualizado.....	52
Figura 9 Configurando memória do ambiente virtual Windows 98.....	53
Figura 10 Ambiente virtual Windows 98.....	53
Figura 11 Teste com ambiente Windows 98 virtualizado.....	54
Figura 12 Processos de instalação do Virtual PC.....	61
Figura 13 Processos de criação do ambiente virtual.....	62
Figura 14 Processos de instalação do ambiente virtual Windows 98.....	63
Figura 15 Processos de instalação e execução do SOFT_LEGADO.....	64
Figura 16 Execução do SOFT_LEGADO no ambiente Windows 98 Virtualizado...	64
Figura 17 Arquivos baixados do Virtual PC.....	71
Figura 18 Instalando Virtual PC – Passo 1.....	71
Figura 19 Instalando Virtual PC – Passo 2.....	72
Figura 20 Instalando Virtual PC – Passo 3.....	73
Figura 21 Instalando Virtual PC – Passo 4.....	73
Figura 22 Instalando Virtual PC – Passo 5.....	74
Figura 23 Instalando Virtual PC – Passo 6.....	74

Figura 24 Instalando Virtual PC – Passo 7.....	75
Figura 25 Criando Ambiente Virtual – Passo 1.....	76
Figura 26 Criando Ambiente Virtual – Passo 2.....	77
Figura 27 Criando Ambiente Virtual – Passo 3.....	77
Figura 28 Criando Ambiente Virtual – Passo 4.....	78
Figura 29 Criando Ambiente Virtual – Passo 5.....	78
Figura 30 Criando Ambiente Virtual – Passo 6.....	79
Figura 31 Criando Ambiente Virtual – Passo 7.....	79
Figura 32 Configurando Ambiente Virtual – Passo 1.....	80
Figura 33 Configurando Ambiente Virtual – Passo 2.....	81
Figura 34 Configurando Ambiente Virtual – Passo 3.....	81
Figura 35 Configurando Ambiente Virtual – Passo 4.....	82
Figura 36 Configurando Ambiente Virtual – Passo 5.....	82
Figura 37 Configurando Ambiente Virtual – Passo 6.....	83
Figura 38 Configurando Ambiente Virtual – Passo 7.....	83
Figura 39 Configurando Ambiente Virtual – Passo 8.....	84
Figura 40 Configurando Ambiente Virtual – Passo 9.....	84
Figura 41 Configurando Ambiente Virtual – Passo 10.....	85
Figura 42 Inicializando o ambiente virtual – Parte 1.....	85
Figura 43 Inicializando o ambiente virtual – Parte 2.....	86
Figura 44 Inicializando o ambiente virtual – Parte 3.....	87
Figura 45 Inicializando o ambiente virtual – Parte 4.....	87
Figura 46 Utilizando o ambiente virtual.....	88

## SUMÁRIO

1	INTRODUÇÃO .....	13
1.1	OBJETIVO GERAL .....	15
1.2	OBJETIVOS ESPECÍFICOS .....	15
1.3	JUSTIFICATIVA .....	16
1.4	ESTRUTURA DO TRABALHO .....	17
2	ENGENHARIA DE SOFTWARE .....	19
2.1	PROCESSO DE SOFTWARE .....	19
2.2	TESTE DE SOFTWARE .....	22
2.3	TÉCNICAS DE TESTE DE SOFTWARE .....	24
2.3.1	<b>Caixa-Branca .....</b>	<b>24</b>
2.3.2	<b>Caixa-Preta .....</b>	<b>25</b>
2.4	TIPOS DE TESTE DE SOFTWARE .....	26
2.4.1	<b>Teste de Conformidade .....</b>	<b>26</b>
2.4.2	<b>Teste Funcional.....</b>	<b>27</b>
2.4.3	<b>Teste de Desempenho .....</b>	<b>27</b>
2.4.4	<b>Teste de Regressão.....</b>	<b>27</b>
2.4.5	<b>Teste Smoke .....</b>	<b>28</b>
2.4.6	<b>Teste Stress.....</b>	<b>28</b>
2.4.7	<b>Teste Unitário .....</b>	<b>28</b>
2.4.8	<b>Teste de Compatibilidade .....</b>	<b>29</b>
2.5	EVOLUÇÃO DO SOFTWARE.....	29
2.6	SOFTWARE LEGADO.....	30
3	VIRTUALIZAÇÃO.....	33
3.1	TERMINOLOGIAS.....	34
3.1.1	<b>Hipervisor .....</b>	<b>34</b>
3.1.2	<b>Máquina Virtual.....</b>	<b>34</b>

3.1.3	<b>Paravirtualização.....</b>	<b>35</b>
3.1.4	<b>Virtualização de Aplicativos.....</b>	<b>35</b>
3.2	<b>MÁQUINAS VIRTUAIS.....</b>	<b>36</b>
3.2.1	<b>Tipo I.....</b>	<b>37</b>
3.2.2	<b>Tipo II.....</b>	<b>38</b>
3.3	<b>EXEMPLOS SIMULADORES.....</b>	<b>39</b>
3.3.1	<b>Xen.....</b>	<b>39</b>
3.3.2	<b>VMware.....</b>	<b>40</b>
3.3.3	<b>Microsoft Virtual PC 2007.....</b>	<b>41</b>
4	<b>A VIRTUALIZAÇÃO NO AUXÍLIO À ETAPA DE TESTE DE COMPATIBILIDADE DE SOFTWARE.....</b>	<b>42</b>
4.1	<b>AMBIENTE DE TESTES.....</b>	<b>44</b>
4.2	<b>TESTES.....</b>	<b>49</b>
4.2.1	<b>SOFT_VENDAS.....</b>	<b>49</b>
4.3	<b>RESULTADOS OBTIDOS.....</b>	<b>56</b>
4.4	<b>VANTAGENS DO USO DE MÁQUINAS VIRTUAIS.....</b>	<b>57</b>
4.5	<b>DESVANTAGENS DO USO DE MÁQUINAS VIRTUAIS.....</b>	<b>58</b>
4.6	<b>ESTUDO DE CASO.....</b>	<b>58</b>
4.6.1	<b>Terminologias.....</b>	<b>59</b>
4.6.2	<b>Histórico.....</b>	<b>59</b>
4.6.3	<b>Hardware e Software Utilizados.....</b>	<b>60</b>
4.6.4	<b>Visão Geral do Processo.....</b>	<b>60</b>
4.6.5	<b>Processo de Instalação do Monitor de Ambiente Virtual.....</b>	<b>61</b>
4.6.6	<b>Processo de Criação do Ambiente Virtual.....</b>	<b>62</b>
4.6.7	<b>Processo de Instalação do ambiente virtual Windows 98.....</b>	<b>63</b>
4.6.8	<b>Processo de Instalação e Execução do SOFT_LEGADO.....</b>	<b>63</b>
4.6.9	<b>Resultados Obtidos.....</b>	<b>65</b>
	<b>CONCLUSÃO.....</b>	<b>66</b>
	<b>REFERÊNCIAS.....</b>	<b>68</b>

BAIXANDO O VIRTUAL PC .....	70
<b>Recursos necessários.....</b>	<b>70</b>
<b>Instalando.....</b>	<b>71</b>
<b>Configurando o ambiente virtual.....</b>	<b>80</b>
<b>Utilizando um ambiente virtual.....</b>	<b>87</b>

## 1 INTRODUÇÃO

A fase de teste de software é importante para minimizar os possíveis erros nos softwares, bem como a fase de teste de compatibilidade com outros ambientes de execução é citada como essencial segundo Bartie (2005).

As organizações encontram dificuldades em realizar os mais diversos testes de software antes do mesmo de ser liberado para o usuário final. Muitos testes de software, como o teste de compatibilidade, demandam das organizações disponibilidade de diferentes equipamentos e sistemas operacionais para que os testes possam ser realizados. Desta forma, as organizações desenvolvedoras tentam a cada dia encontrar alternativas para suas equipes de testes de softwares no momento de realizarem os diversos testes em diferentes ambientes computacionais.

Porém, não somente os softwares novos devem ser testados. Os chamados softwares legados, que são softwares cuja tecnologia tal qual foram desenvolvidos encontram-se obsoletas, Bennett (1995), também necessitam ser testados e mantidos em funcionamento com as novas tecnologias e com novas versões de sistemas operacionais tais quais estão sendo executados.

A migração desses softwares para novas plataformas, como hardware e software, pode e geralmente apresentam inconsistências quando não realizados os devidos testes.

Para reduzir os problemas de testes de compatibilidade de software esta pesquisa utiliza a virtualização em conjunto com teste de compatibilidade de software como alternativa no momento da realização dos testes.

Para o problema dos softwares legados, esta pesquisa apresenta um estudo de caso para utilização de softwares legados empregando virtualização de sistemas

operacionais, mantendo assim os softwares legados em execução em conjunto com novas tecnologias.

A virtualização é uma técnica da qual permite que um processador funcione idêntico a vários processadores trabalhando em paralelo, possibilitando que vários sistemas operacionais sejam executados ao mesmo tempo em um mesmo computador, afirma Nanda (2005).

A técnica de virtualização não é uma técnica inovadora, existem no mercado atualmente algumas ferramentas que oferecem opções para simulação de ambientes computacionais, como se pode citar: Virtual PC, VMware e Xen.

Apesar de não ser inovadora a técnica de virtualização provê suporte para que inúmeros ambientes computacionais sejam executados ao mesmo tempo em um único computador físico, o que para esta pesquisa é essencial, pois a utilização de vários ambientes computacionais pode proporcionar maior agilidade e economia no momento de realizar testes de compatibilidade de software.

## 1.1 OBJETIVO GERAL

Utilizar virtualização em conjunto com teste de compatibilidade de software a fim de indicar uma alternativa aos desenvolvedores e equipes de teste de software no momento de realizarem testes em diferentes ambientes computacionais.

## 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos desta pesquisa são:

- a) descrever e documentar as etapas em que a virtualização pode auxiliar durante as fases de teste de software;
- b) instalação, configuração e utilização de uma ferramenta de virtualização;
- c) comparação entre duas ferramentas de virtualização;
- d) realizar testes de compatibilidade de software utilizando virtualização;
- e) realizar um estudo de caso com softwares legados, a partir de uma ferramenta de virtualização;

### 1.3 JUSTIFICATIVA

Trayahú (2006) afirma que mais de 90% dos sistemas são liberados com algum defeito. Comprovando estas estatísticas várias revistas, livros e a própria internet relatam inúmeras falhas em softwares.

Buscando diminuir esta estatística, a realização de testes de software traz vantagens, dentre elas pode-se citar a confiabilidade do usuário final, alta qualidade do produto desenvolvido, entre outros, e assim reduzindo diversos custos tanto para a organização desenvolvedora quanto para o cliente.

Além dos testes em softwares novos, a realização de testes em softwares que já estão em produção há algum tempo, também são importantes. Os chamados softwares legados, cuja tecnologia tal qual foram desenvolvidos encontra-se ultrapassada (BENNETT, 1995) também necessitam de testes. Testes estes que podem e devem ser realizados no momento de migrarem para uma versão mais atualizada deste software ou mesmo no momento da migração de plataformas, como hardware e sistemas operacionais ao qual estão sendo executados.

Porém, a realização de testes de compatibilidade demanda das organizações disponibilidade de diferentes equipamentos e sistemas operacionais para que os testes possam ser realizados.

Buscando facilitar a fase de testes de compatibilidade de software, a utilização de virtualização pode proporcionar aos desenvolvedores e equipes de testes uma alternativa que os auxiliem no momento de testarem um software.

Nanda (2005) afirma que a virtualização traz como benefício a capacidade que vários sistemas operacionais sejam executados ao mesmo tempo em um mesmo computador.

Existem atualmente no mercado algumas ferramentas que oferecem opções para simulação de ambientes computacionais, como se pode citar: Virtual PC, VMware e Xen.

Esta pesquisa propõe aplicar a técnica de virtualização em conjunto com a fase de teste de compatibilidade gerando assim uma alternativa prática e livre de custos para as organizações no momento de realizarem os testes de compatibilidade de software.

Porém, além dos testes dos softwares novos manter softwares legados muitas vezes pode ser trabalhoso e produzir um custo. Alguns destes softwares já não possuem mais suporte das organizações desenvolvedoras para atualizações e mesmo assim podem ser de fundamental importância para a organização que o utiliza. Com o surgimento de novas tecnologias, tal como uma versão mais atualizada de um determinado Sistema Operacional, este considerado software legado, tenderá a não estar compatível.

Desta forma, esta pesquisa apresenta através de um estudo de caso uma alternativa para as organizações a fim de manter softwares legados em funcionamento em conjunto com as novas tecnologias.

#### 1.4 ESTRUTURA DO TRABALHO

A estrutura desta pesquisa está separada em quatro capítulos para melhor compreensão. Um capítulo apresenta uma seção introdutória e os objetivos desta pesquisa. No capítulo segundo é apresentado um conceito sobre engenharia de software com foco sobre testes de software e softwares legados. O capítulo seguinte faz uma explanação sobre virtualização, máquinas virtuais e terminologias utilizadas. O capítulo

quatro é apresentado o trabalho desenvolvido, no qual vincula a técnica de virtualização com a fase de teste de compatibilidade de software apresentando uma alternativa para equipes de testes de compatibilidade de software, bem como um estudo de caso para manter softwares legados em funcionamento em conjunto com novas tecnologias.

## 2 ENGENHARIA DE SOFTWARE

Por meio da utilização de tecnologias e práticas da ciência da computação, gerência de projetos e outras áreas, a engenharia de software (SOMMERVILLE, 2003) tem por objetivo a organização, produtividade e qualidade do software a ser desenvolvido, utilizando linguagens de programação, banco de dados, bibliotecas, qualidade de software, entre outras.

Além de utilizar modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, programar e manter sistemas, a engenharia de software deve oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento.

Para que se possa gerenciar o desenvolvimento do software como um todo, a engenharia de software apresenta o processo de software como uma prática a ser adotada.

### 2.1 PROCESSO DE SOFTWARE

Segundo Pressman (2002) o processo de software tem como base uma seqüência coerente de práticas que tem por objetivo o desenvolvimento ou evolução de sistemas de software. Estas práticas englobam as atividades de especificação, projeto, implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos.

Essa seqüência de prática é denominada modelo de processo.

Um modelo de processo ou considerado por alguns autores como ciclo de vida, pode ser visto como uma representação, ou abstração dos objetos e atividades envolvidas no processo de software, ou seja, é o imo do processo, pois define as fases e atividades a serem seguidas. Além disso, o modelo de processo oferece uma forma mais abrangente e fácil de representar o gerenciamento e conseqüentemente o progresso do projeto, afirma Pressman (2002).

Modelo Seqüencial ou Cascata, Espiral e Prototipação são alguns exemplos de modelos de processo de software.

Para demonstrar o processo de software esta pesquisa utiliza-se do modelo cascata que é considerado por Leach (2000) o modelo mais popular no processo de software.

No modelo Cascata, cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase. O software é desenvolvido em um longo processo e entregue ao final deste. O modelo sugere laços de retorno que permitem realimentar fases anteriores, mas em geral, o modelo cascata é considerado um modelo linear (LEACH, 2000). A Figura 1 fornece uma descrição visual do modelo.

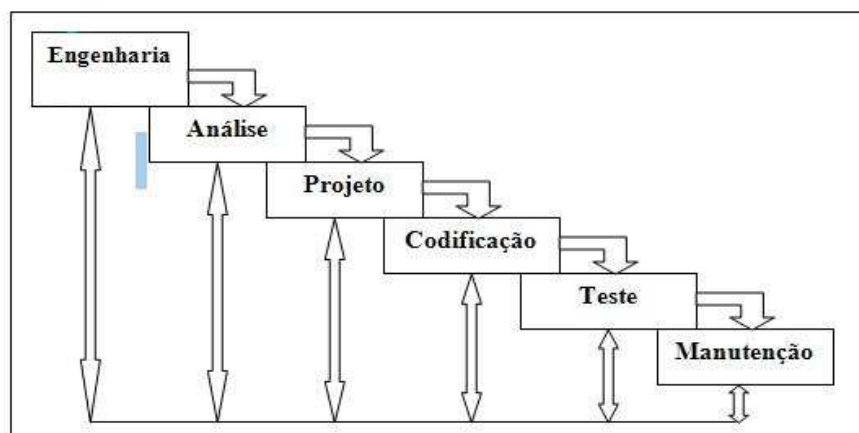


Figura 1 Ciclo de vida clássico de um software  
Fonte: Pressman, R (2002)

Conforme Pressman (2002) as etapas são as seguintes:

- a) análise e engenharia de sistemas:** é nesta etapa que estão envolvidos a coleta de requisitos em nível do sistema, pequena quantidade de projeto e análise de alto nível;
- b) análise de requisitos de software:** o processo de coleta dos requisitos é intensificado e concentrado especificamente no software. Os requisitos são documentados e revistos com o cliente. Para tornar concluída esta fase é necessário satisfazer os seguintes requisitos:
- satisfazer os objetivos da organização,
  - especificações aceitas pelos usuários,
  - especificações devem ser táticas e economicamente viáveis,
  - lógica do processamento bem definida;
- c) projeto:** é a tradução dos requisitos do software para um conjunto de representações que podem ser avaliadas quanto à qualidade, antes que a codificação se inicie. Concentra-se em quatro atributos do programa:
- estrutura de Dados,
  - arquitetura de Software,
  - detalhes Procedimentais,
  - caracterização de Interfaces;
- d) codificação:** é a tradução das representações do projeto para uma linguagem “artificial” resultando em instruções executáveis pelo computador, onde engloba a revisão das especificações dos programas, desenvolvimento da lógica dos programas, codificação em si, construção das tabelas do banco de dados, se necessário, testes dos programas e elaboração dos manuais de operação.

**e) testes:** concentra-se nos aspectos lógicos internos do software, garantindo que todas as instruções tenham sido testadas nos aspectos funcionais externos, para descobrir erros e garantir que a entrada definida produza resultados que concordem com os esperados. Começa quando os componentes do sistema, já testados individualmente, são reunidos para testes e aceitação como sistema. Tem-se nessa etapa o treinamento para implantação, testes do sistema, revisão dos procedimentos operacionais e conversão do sistema.

**f) manutenção:** o software poderá sofrer mudanças depois que for entregue ao cliente. Algumas causas dessas mudanças podem ser: erros, adaptação do software para acomodar mudanças em seu ambiente externo e exigência do cliente para acréscimos funcionais e de desempenho.

Dentre as etapas apresentadas anteriormente, a fase de teste destaca-se de forma fundamental no combate aos problemas que possam ocorrer de um modo geral.

## 2.2 TESTE DE SOFTWARE

Uma das maiores dificuldades enfrentadas até hoje são as falhas nos aplicativos, fazendo com que a fase de testes torne-se de essencial importância.

Brandão (2007) em matéria publicada na revista TI Master, utiliza-se de um exemplo para formalizar a importância dos testes, segue:

Se uma fábrica de pneus não submete seus produtos a uma bateria de testes, imagine como aumentaria o número de acidentes de carro. Os automóveis também passam por várias verificações antes de chegarem ao comprador. Textos acadêmicos e literários precisam passar pelas mãos de um revisor antes de serem publicados. Por que seria diferente com softwares?

A fase de teste é um processo que envolve ações que vão do levantamento de requisitos até a execução do teste propriamente dito. São estes que garantem, ou tentam garantir, o funcionamento correto do software.

O tamanho do projeto a ser desenvolvido e a quantidade de pessoas envolvidas no processo aumentam ainda mais a complexidade. Assim, a presença de falhas é inevitável.

Segundo Pressman (1995) o objetivo desta fase é encontrar defeitos nos softwares, para que estes possam ser corrigidos pela equipe desenvolvedora, ou seja, esta fase é utilizada como um processo da engenharia de software para encontrar defeitos.

Na fase de testes de software é criada uma série de testes que tem a intenção de encontrar possíveis falhas no seu desenvolvimento, e um bom caso é aquele que tem uma elevada probabilidade de revelar um erro ainda não descoberto.

Para que esta atividade seja bem conduzida, ela deverá descobrir possíveis erros no software. Como benefício secundário, ela demonstra que as funções de software estão trabalhando de acordo com as especificações, e que os requisitos de desempenho aparentemente foram cumpridos. Além disso, os dados compilados durante a atividade de testes é levada a efeito e proporcionam uma boa indicação da confiabilidade no software e alguma indicação da qualidade deste, como um todo (PRESSMAN, 1995).

Dizer que o software falhou, significa basicamente que não está de acordo com o esperado pelo usuário, ou com a sua especificação inicial.

Testar softwares implica em subdividir os testes em algumas categorias para que se possa abordar os softwares de maneiras diferentes. Assim, cercado-se todos os lados fica mais fácil encontrar possíveis falhas.

Tais categorias são chamadas de técnicas de testes de software.

## 2.3 TÉCNICAS DE TESTE DE SOFTWARE

Atualmente existem muitas maneiras de se testar um software, porém o objetivo principal destas técnicas é o mesmo: encontrar falhas no software (PRESSMAN, 2002).

Dentre as técnicas mais conhecidas pode-se citar o Teste da Caixa-Branca e Teste de Caixa-Preta

### 2.3.1 Caixa-Branca

Técnica que avalia o comportamento interno do componente de software. Trabalha diretamente sobre o código-fonte do componente de software para avaliar aspectos tais como: teste de condição, teste de fluxo de dados, teste de ciclos e teste de caminhos lógicos. Segundo Trayahú (2006) os aspectos avaliados nesta técnica de teste dependem da complexidade e da tecnologia que determinam a construção do componente de software, cabendo, portanto avaliação de mais aspectos do que os citados anteriormente.

Esta categoria de teste é desenvolvida analisando-se o código fonte e elaborando-se casos de teste que cubram todas as possibilidades do componente de software. Dessa maneira, todas as variações originadas por estruturas de condições são testadas. Também se enquadram nessa técnica testes manuais ou testes efetuados com

apoio de ferramentas para verificação de aderência a boas práticas de codificação reconhecidas pelo mercado de software.

Segundo Sommerville (2003) esta técnica é recomendada para as fases de Teste da Unidade e Teste da Integração, cuja responsabilidade principal fica a cargo dos desenvolvedores do software, que por sua vez conhecem bem o código-fonte produzido.

### **2.3.2 Caixa-Preta**

Técnica em que o componente de software a ser testado é abordado como se fosse uma caixa-preta, ou seja, não se considera o comportamento interno do mesmo. Segundo Pressman (2002) dados de entrada são fornecidos, o teste é executado e o resultado obtido é comparado a um resultado esperado previamente conhecido. Haverá sucesso no teste se o resultado obtido for igual ao resultado esperado.

De forma geral, para se descobrir qual categoria de teste aplicar é necessário levantar o foco do software. Alguns fatores como: se o sistema é visto em partes, com diferentes linguagens aplicadas ou se tudo é integrado e apenas uma linguagem foi adotada ajudam ou atrapalham, a decidir qual categoria de teste integrar. Outro fator fundamental também é se a equipe de testes tem o conhecimento suficiente pra realizar uma categoria de teste caixa branca, afirma Estevam (2007).

Segundo Estevam (2007) grande parte das empresas brasileiras ainda utilizam os próprios desenvolvedores para testarem os próprios softwares. Desta forma, pode-se afirmar que a maior parte dos testes realizados por aí fora são caixa-branca.

Desta forma uma desvantagem é que dificilmente o programador acha o próprio erro, devido a fator tempo, preocupação em terminar a codificação e também

porque principalmente o desenvolvedor, envolto em seu raciocínio no momento do desenvolvimento, simplesmente julgará certos comportamentos da aplicação como sendo corretos, sendo que às vezes podem não ser.

Enfim, a fase de teste é uma fase de extrema importância, desde um simples teste até mesmo o mais complexo dos sistemas, é necessário testar.

Para facilitar um pouco esta tarefa, existem tipos de testes que podem ser aplicados no momento da realização de testes.

## 2.4 TIPOS DE TESTE DE SOFTWARE

Testar um software implica em subdividir os testes em algumas categorias para que se possa abordar os software de maneiras diferentes. Para isso se utiliza os tipos de testes de software. Eles formam uma cadeia para que todos os testes possam ser realizados afim de não deixar que nenhum erro ou falha no software possa passar despercebido.

### 2.4.1 Teste de Conformidade

O teste de conformidade, segundo afirma Estevam (2007), verifica a implementação de acordo com padrões da indústria. Produzido para ter certeza do comportamento que suporte portabilidade, interoperabilidade ou compatibilidade de padrões definidos.

### **2.4.2 Teste Funcional**

O teste funcional valida um aplicação em conformidade com as especificações de modo que atenda as funções requeridas. É dividido num passo a passo de modo que cada funcionalidade é testada e tem sua funcionalidade aprovada ou reprovada utilizando de dados normais e anormais como entrada. Pode envolver teste da interface, API, gerenciamento de BD, segurança, instalação, redes, etc. Pode ser feito uma automação ou mesmo manual, usando metodologias de caixa branca ou caixa preta, afirma Estevam (2007).

### **2.4.3 Teste de Desempenho**

O teste de desempenho é utilizado para compreender escalabilidade. Este tipo de teste identifica onde residem os gargalos dos sistemas, se é na própria aplicação ou na infra-estrutura em que ela está implementada, afirma Estevam (2007). Geralmente envolve teste de automação, facilitando a simulação de uma série de cenários como normal, picos e condições de carregamentos extraordinários.

### **2.4.4 Teste de Regressão**

Estevam (2007) afirma que o teste de regressão é similar em escopo ao teste funcional. Um teste de regressão permite uma validação repetitiva e consistente de cada lançamento de uma versão do produto testado. Tais testes, reportam defeitos que já foram corrigidos porém, por algum motivo, sofreram problemas de qualidade no novo

processo de manutenção. Geralmente é um teste manual, mas se usa muito automação para reduzir o tempo de teste em uma plataforma já testada anteriormente.

#### **2.4.5 Teste Smoke**

O teste smoke é um teste bem superficial, de maneira exploratória, das funções básicas de um pedaço da aplicação sem se ater a detalhes. Originalmente, esse teste foi herdado de testes de hardware onde se ligava uma função qualquer e se a peça não pegasse fogo, o teste era positivo (Estevam, 2007).

#### **2.4.6 Teste Stress**

O teste de stress conduz uma avaliação de um sistema ou um componente além ou aquém dos limites que suporta ou não para determinar quando falha ou não, quando um pico é atingido. É semelhante ao Teste de desempenho, porém os testes são submetidos a uma carga bem maior de dados (Estevam, 2007).

#### **2.4.7 Teste Unitário**

O teste unitário é funcional e confiável num ambiente de desenvolvimento. Avalia o comportamento de componentes de um produto para certificar que os comportamentos de saída estão de acordo com a integração do sistema e especificações (Estevam, 2007)

### 2.4.8 Teste de Compatibilidade

O teste de compatibilidade é utilizado para assegurar a compatibilidade entre sistemas ou aplicações. Um exemplo simples é o teste para assegurar compatibilidade de um sistema web que rode em diferentes browsers, em diferentes sistemas operacionais e diferentes plataformas de hardware. Pode ser feito manualmente.

Para fins de estudo desta pesquisa, o enfoque principal foi à fase de teste de compatibilidade de software.

Porém, os testes não devem ser realizados apenas nos novos softwares ou em novas versões. Até mesmo os softwares que já estão em produção devem receber testes, principalmente de teste de compatibilidade e teste funcional, devido a velocidade com que novas tecnologias são apresentadas (Estevam, 2007).

## 2.5 EVOLUÇÃO DO SOFTWARE

Nos dias atuais são notórias as mudanças nos software, e muitos são desenvolvidos para refletir os comportamentos do mundo real. No entanto, tem-se a necessidade de que o mesmo acompanhe as mudanças de requisitos impostas pelo ambiente no qual está inserido. Uma vez que esse acompanhamento não seja feito, ou seja falho, pode acarretar em perda de qualidade por parte do software ou até mesmo o fim da sua vida útil.

Das desvantagens causadas pelo envelhecimento natural, podemos citar a perda de desempenho devido a modificações não adequadas na sua estrutura interna, número crescente de novos erros devidos a alterações indevidas no código e perda de

usuários devido à falta de meios para concorrer com versões mais recentes de sistemas semelhantes.

Não se tem como evitar o envelhecimento, o que se pode tentar é entender o que está causando isto e assim tomar medidas para diminuir seus efeitos.

O envelhecimento do software muitas vezes se dá através do próprio envelhecimento das tecnologias aplicadas no seu desenvolvimento, fazendo com que essas tecnologias tornem-se ultrapassadas, o que se chama software legado.

## 2.6 SOFTWARE LEGADO

A definição de software legado se dá através da evolução das tecnologias aplicadas tanto no desenvolvimento quanto na utilização desses softwares. Essa evolução dá-se a uma velocidade muitas vezes incapaz de ser acompanhada, fazendo com que as técnicas utilizadas no software tornem-se obsoletas, (BENNETT, 1995), como por exemplo, Cobol ou Fortran.

A migração e/ou alteração desse tipo de software gera desafios técnicos, por exemplo: verificar efeitos colaterais de uma alteração, e não-técnicos, por ex.: estimar o custo de alteração a todos os envolvidos com manutenção de software.

Ao mesmo tempo em que ele traz incorporado o acúmulo de anos de experiência e refinamento, traz também todos os vícios e defeitos vigentes na época de seu desenvolvimento, mesmo que naquela época, o que hoje é chamado de vício e defeito, fosse à melhor indicação para o desenvolvimento de software (BENNETT, 1995). Por exemplo, o tamanho do programa, que devido à limitação imposta pelo custo do hardware, tinha que ser reduzido, atualmente não é crucial (devido ao barateamento dos componentes de hardware).

Houve ainda a época em que se prezava a eficiência, em detrimento da clareza e estruturação do programa. Adicionem-se a isso as sucessivas manutenções e conseqüentes degenerações sofridas pelo software. Sabe-se hoje que todos esses fatores estão diretamente relacionados com a dificuldade de entendimento de qualquer software legado.

Muitos softwares legados são críticos para os negócios das organizações que os possuem. Eles têm embutidas informações dos negócios e procedimentos, que podem não estar documentados.

As dificuldades devem ser enfrentadas, pois qualquer software que não evoluir continuamente tornar-se-á menos útil no mundo real Bennett (1995).

Segundo Bennett (1995), os maiores problemas em manter software legado são:

- a) Desestruturação e dificuldade de entendimento do código, muitas vezes porque o software foi desenvolvido antes da introdução dos métodos de programação;
- b) Programadores que não participaram do desenvolvimento de um produto de software sentem dificuldade em entender e mapear a funcionalidade para o código fonte;
- c) Documentação desatualizada, não auxiliando em nada a equipe de manutenção;
- d) Dificuldade de prever as conseqüências de efeitos colaterais;
- e) Dificuldade de administrar múltiplas alterações concorrentes.

O problema de manutenção de sistemas legados é complexo porque, geralmente, esses sistemas não são simples programas, desenvolvidos e mantidos.

Muitos deles são compostos de outros diferentes programas que, de alguma forma, compartilham dados. Também ocorre que esses programas foram desenvolvidos por diferentes pessoas ao longo dos anos, as quais não estão mais na organização.

Com o objetivo de minimizar os problemas gerados por manutenções difíceis e, algumas vezes, degenerativas da estrutura do sistema, muitas organizações tem buscado alternativas para mantê-los funcionando mesmo com a necessidade de evolução das tecnologias empregadas.

Outros por sua vez, reescrevem todo o software legado em linguagens novas ou até mesmo transplantando a base de informações do software legado para bancos de dados atuais.

Porém, como citado anteriormente, muitos são os fatores que podem tornar essa troca um problema. Para este problema esta pesquisa apresenta a virtualização como saída para a realização de testes em diferentes ambientes computacionais, os ambientes virtuais podem simular o ambiente real da organização para que se possam realizar os devidos testes sem a necessidade de intervir no processo de vida atual da organização.

### 3 VIRTUALIZAÇÃO

A técnica de virtualização tem por objetivo utilizar-se de recursos computacionais a fim de apresentar ambientes computacionais diferenciados.

A virtualização utiliza-se de metodologias como particionamento ou agregação de hardware e software, simulação parcial ou completa, compartilhamento de tempo, dentre outros (KIYANCLAR, 2005).

Desta forma a virtualização fornece uma camada de abstração entre sistemas de hardware de computador e o software que roda nestes sistemas. Ao proporcionar uma visão lógica dos recursos de computação, em vez de uma visão física, as soluções de virtualização possibilitam dentre que se executem múltiplos sistemas operacionais simultaneamente em uma única máquina física (KIYANCLAR, 2005).

A técnica de virtualização já era utilizada em servidores mainframes. Porém com o aumento da variedade dos tipos de hardwares e sistemas operacionais, esta técnica está voltando a ser utilizada e revigorada.

É através da virtualização que é possível a execução de diversas aplicações desenvolvidas primeiramente para um específico hardware ou sistema operacional, em outros ambientes computacionais.

Os exemplos mais comuns de softwares de virtualização de ambientes computacionais, que são Virtual PC, VmWare, Xen que permitem que os usuários executem réplicas múltiplas de sistemas operacionais como Windows e Linux hospedados em um sistema anfitrião.

Para compreender melhor como funciona a virtualização, têm-se dois modos em sua execução: administrador e não-administrador.

O modo administrador, geralmente é executado apenas no monitor da máquina virtual, e o modo não administrador é executado nos sistemas operacionais hospedeiros.

Algumas nomenclaturas são utilizadas dentro da virtualização.

### 3.1 TERMINOLOGIAS

Para que se possa compreender um pouco mais sobre o processo de virtualização, tem-se a necessidade de compreender melhor sobre algumas terminologias utilizadas neste meio.

#### 3.1.1 Hipervisor

Hipervisor é o componente mais básico da virtualização. É o software que desvincula o sistema operacional e os aplicativos de seus recursos físicos. Um hipervisor tem seu próprio *kernel* e é instalado diretamente no hardware. Ele é, quase literalmente, inserido entre o hardware e o Sistema Operacional.

#### 3.1.2 Máquina Virtual

Uma máquina virtual é um ambiente operacional auto-suficiente, ou seja, um software que funciona com, mas é independente de, um sistema operacional local.

Em outras palavras, é uma implementação em software de uma CPU, independente de plataforma, que executa código compilado. As máquinas virtuais têm

de ser desenvolvidas especificamente para os sistemas operacionais nos quais rodam. As tecnologias de virtualização também são chamadas de “software dinâmico de máquina virtual”.

### **3.1.3 Paravirtualização**

Paravirtualização é um tipo de virtualização em que o sistema operacional inteiro roda sobre o hipervisor e se comunica com ele diretamente, resultando em melhor desempenho. Porém, os *kernels* do sistema operacional e do hipervisor precisam ser modificados para acomodar esta interação estreita.

Um sistema operacional Linux paravirtualizado, por exemplo, é otimizado especificamente para rodar em um ambiente virtual. A virtualização total, em comparação, apresenta uma camada abstrata que intercepta todas as chamadas para recursos físicos.

A paravirtualização se apóia em um subconjunto virtualizado da arquitetura x86. Avanços recentes em chips da Intel e da AMD estão ajudando a suportar esquemas de virtualização que não requerem sistemas operacionais modificados.

### **3.1.4 Virtualização de Aplicativos**

A virtualização na camada de aplicativos isola os programas do hardware e do sistema operacional, encapsulando-os como objetos móveis independentes que podem ser deslocados sem afetar outros sistemas. As tecnologias de virtualização de

aplicativos reduzem as alterações relacionadas a aplicativos no sistema operacional e diminuem os desafios de compatibilidade com outros programas.

Para utilização da técnica de virtualização esta pesquisa utiliza-se de máquina virtual. Para tanto, é necessário uma melhor compreensão sobre o assunto.

### 3.2 MÁQUINAS VIRTUAIS

O conceito de máquinas virtuais surgiu paralelamente com a técnica de virtualização. Tal conceito foi desenvolvido e vem sendo utilizado desde os anos 60, quando a IBM a fim de centralizar os sistemas de computador utilizou no ambiente chamado VM/370.

Neste caso, cada máquina virtual simulou outra máquina oferecendo aos usuários a ilusão de estar acessando diretamente outra máquina fisicamente (GOLDBERG, 1974).

Segundo Goldberg (1974) uma máquina virtual é definida como “uma duplicata eficiente e isolada de uma máquina real”.

Enquanto a máquina real utiliza-se de uma camada de software de baixo nível para fornecer acesso aos vários recursos do hardware, em uma máquina virtual o emulador implementa todas as instruções realizadas pela máquina real em um ambiente abstrato, possibilitando executar um aplicativo de uma plataforma em outra.

Uma máquina virtual é um ambiente criado por um monitor de máquinas virtuais. O monitor de máquinas virtuais pode criar uma ou mais máquinas virtuais sobre uma única máquina real (GOLDBERG, 1974).

Enquanto um emulador fornece uma camada de abstração completa entre o sistema em execução e o hardware, um monitor de máquina virtual abstrai o hardware subjacente e controla uma ou mais máquinas virtuais.

Cada máquina virtual fornece facilidades para uma aplicação ou um “sistema convidado” que acredita estar executando sobre um ambiente normal de acesso físico ao hardware.

Existem basicamente dois tipos de construção de máquinas virtuais: o tipo I onde o monitor de máquinas virtuais é implementado entre o hardware e os sistemas convidados, e o tipo II onde o monitor é implementado como um processo de um sistema operacional real subjacente, denominado sistema anfitrião (GOLDBERG, 1974).

### **3.2.1 Tipo I**

Na abordagem da construção de máquina virtual do tipo I, o monitor de máquina virtual é colocado diretamente sobre o hardware, e as máquinas virtuais são criadas na camada logo acima (GOLDBERG, 1974).

O monitor de máquina virtual, Figura 2, é executado com maior prioridade e os sistemas convidados são executados no modo usuário. Desta forma o monitor de máquina virtual pode emular e interceptar todas as operações dos sistemas convidados que acessam ou manipulam os recursos de hardware.

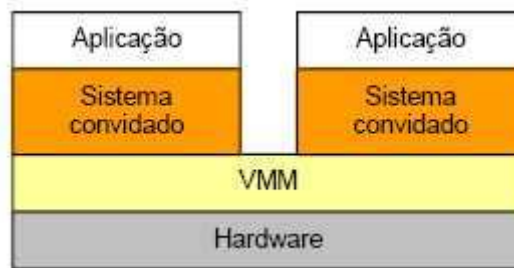


Figura 2 Monitor de máquinas virtuais de tipo I  
Fonte: Goldberg, R. (1974)

### 3.2.2 Tipo II

Na abordagem de construção da máquina virtual do tipo II, Figura 3, é adicionada uma camada acima do sistema operacional denominado hospedeiro, chamando assim de máquina virtual hospedada (GOLDBERG, 1974).

Desta forma a máquina virtual é instalada e executada com uma aplicação típica, além de utilizar os recursos de hardware através de drives de dispositivos do sistema operacional.

Como o sistema operacional convidado e o ambiente de execução na máquina virtual são idênticos ao da máquina real, é possível usar os softwares já construídos para a máquina real dentro das máquinas virtuais. Essa transparência evita ter de construir novas aplicações ou adaptar as já existentes.

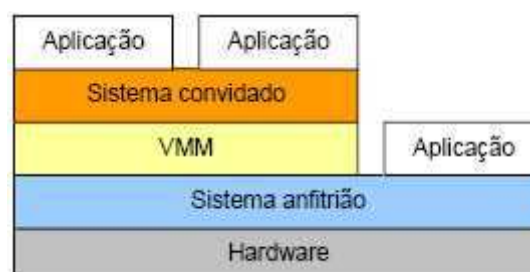


Figura 3 Monitor de máquinas virtuais de tipo II  
Fonte: Goldberg, R. (1974)

Porém, a utilização de um emulador traz como desvantagem a perda de eficiência em traduzir as instruções da máquina real.

### 3.3 EXEMPLOS SIMULADORES

No mercado atualmente existem muitos simuladores de ambientes virtuais. Alguns deles são específicos para uma arquitetura, ou até mesmo específico para simular um determinado sistema operacional.

Abaixo segue alguns exemplos dos simuladores de ambientes virtuais mais comuns encontrados.

#### 3.3.1 Xen

O monitor de máquinas virtuais Xen é uma ferramenta de código aberto desenvolvida pela Universidade de Cambridge, baseada na abordagem clássica de implementação de virtualização, onde o monitor de máquina virtual possui o nível mais alto de privilégio (BARHAM et al. 2003).

A virtualização clássica implementada utilizando a técnica de paravirtualização requer pequenas alterações nos sistemas operacionais virtualizados, já que a interface oferecida pelo monitor de máquina virtual difere em alguns pontos da interface real do computador. Efetuando-se estas alterações, um sistema operacional adaptado permite que os processos e serviços internos executem sem nenhuma alteração em seus códigos, sem impor uma sobrecarga limitante que comprometa o desempenho final das aplicações (MENON et al. 2005).

Atualmente, sistemas operacionais como Linux, NetBSD, FreeBSD, e Windows XP possuem implementações adaptadas para execução sobre a interface virtual implementada por Xen.

Além de gerenciar o acesso aos recursos de *hardware* existentes, o monitor de máquina virtual Xen implementa através do *Domain0* um conjunto de chamadas de sistema que permitem migrar um sistema operacional entre computadores. A independência de *hardware* obtida pelo monitor Xen permite que máquinas virtuais sejam encapsuladas e migradas entre computadores com arquitetura virtualizada. Este processo de migração pode ser efetuado sem a interrupção dos serviços em execução sobre o sistema operacional virtualizado (BARHAM et al. 2003).

A migração de máquinas virtuais surge como ferramenta auxiliar para administradores de redes e servidores, já que um sistema operacional pode migrar sem interromper os serviços em execução, permitindo a manutenção do *hardware* do computador hospedeiro quando necessário. Porém, esta ferramenta somente pode ser utilizada em ambientes que utilizam um sistema de compartilhamento de arquivos e diretórios, já que a implementação atual do mecanismo de *live migration* não possui suporte para migração do sistema de arquivos local.

### 3.3.2 VMware

O monitor de máquinas virtuais VMware é um produto desenvolvido pela empresa VMware Inc., utilizando a técnica de virtualização hospedada. Para tal, o sistema é dividido em VMApp, que é a máquina virtual que executa no espaço de usuário do sistema operacional hospedeiro e VMDriver, que é um software que executa junto ao sistema operacional hospedeiro e processa as requisições feitas pelo VMApp (Sugerman et al. 2001).

Utilizando essa arquitetura, VMware pode ser utilizado para executar a maioria dos sistemas operacionais disponíveis no mercado, sem necessitar de alteração do código dos mesmos.

A migração de uma máquina virtual para outro sistema hospedeiro é possível, porém com interrupção momentânea do sistema. Para tal, é feita uma cópia da memória utilizada pela máquina virtual para o novo hospedeiro, bem como o estado atual da máquina virtual e o arquivo com as configurações do Basic Input Output System (BIOS) virtual.

O sistema de arquivos da máquina virtual precisa estar armazenado em algum sistema de armazenamento compartilhado, como por exemplo, um sistema de armazenamento conectado à rede (Sugerman et al. 2001).

### **3.3.3 Microsoft Virtual PC 2007**

O Microsoft Virtual PC 2007, também conhecido como VirtualPC antes de ser adquirido da Connectix pela Microsoft é uma máquina virtual muito parecida com o VMware. Pode ser instalado e configurado na maioria dos sistemas operacionais baseados em x86, sem a necessidade de drives customizados.

O Virtual PC tem suporte para até quatro adaptadores de rede, com suporte até 4gb de memória.

Sua configuração foi baseada na linguagem XML para facilitar a cópia da estação virtual. Uma característica do Virtual PC é a função que permite aos usuários desfazer qualquer mudança no disco virtual, é o chamado *Undo Disks*.

É possível salvar os arquivos de cada máquina individual por completo, pois o mesmo salva estas configurações em arquivos separadamente.

## **4 A VIRTUALIZAÇÃO NO AUXÍLIO À ETAPA DE TESTE DE COMPATIBILIDADE DE SOFTWARE**

A realização da prática de teste de software muitas vezes torna-se um problema para as organizações principalmente quando se fala em testes de compatibilidade de software. Tal prática tem por objetivo realizar testes de software em ambientes com características distintos entre eles, permitindo assim encontrar falhas que só acontecem em determinadas versões de um sistema operacional, ou com a utilização de uma determinada DLL, ou ainda em conjunto com um determinado software ou hardware.

A utilização desta técnica de teste demanda das organizações disponibilidade de diferentes equipamentos e sistemas operacionais para que os testes possam ser realizados.

Utilizar vários computadores com diferentes hardwares e sistemas operacionais, além de ocasionar um gasto maior para manter em funcionamento estes equipamentos, também prejudica a agilidade dos testes. A necessidade de configuração de cada ambiente demanda tempo e algumas vezes pessoas capacitadas para a sua realização.

Além disto, ficar inicializando vários ambientes computacionais por várias vezes ao longo dos testes pode ocasionar um declínio na curva de tempo da fase de testes.

Outros pontos negativos podem ser citados com a utilização de muitos equipamentos para a realização de testes, como disponibilidade de espaço físico para armazenar os equipamento e consumo elevado de energia.

Outro ponto negativo está no momento de configurar um determinado hardware para que se possam realizar os testes de acordo com a variada gama de possibilidades atuais.

A simples configuração da capacidade de memória RAM de cada computador pode se tornar um empecilho no momento de realizar os testes de compatibilidade. Um determinado software pode ter sua compatibilidade comprometida se executado em um computador com 128 megabytes, enquanto que se executado em um computador com um gigabyte pode estar compatível.

Neste caso, a fase de teste de compatibilidade pode causar além de um atraso pela necessidade de abertura do computador e retirada ou colocação de memórias, como também a necessidade de intervenção técnica para efetuar tal manutenção.

Desta forma, as organizações desenvolvedoras tentam a cada dia encontrar alternativas para suas equipes desenvolvedoras e equipes de testes de softwares no momento de realizarem os diversos testes em diferentes ambientes computacionais.

A utilização da técnica de virtualização auxilia nesta prática, simplificando o processo e reduzido custos.

As tecnologias de virtualização permitem o uso de múltiplos ambientes computacionais em um mesmo hardware. Alguns softwares de virtualização, como Virtual PC, VmWare e Xen, permitem a mobilidade e a possibilidade de rápida duplicação, armazenamento e configuração de recursos.

A única ressalva que se faz é em relação aos testes de desempenho, que na sua maioria devem ser realizados em máquinas não hospedeiras de máquinas virtuais, exceto quando software desenvolvido será utilizado somente em máquinas virtuais.

Para demonstrar como a virtualização pode auxiliar neste processo de teste, foi criado um ambiente de testes específico para teste nesta pesquisa.

#### 4.1 AMBIENTE DE TESTES

Para tornar ágeis os processos de testes de software, muitas vezes é necessário oferecer maior liberdade de configuração às equipes de testes.

As tecnologias de virtualização auxiliam com a possibilidade de se ter como sistema hospedeiro as versões mais recentes de sistemas operacionais que lhe oferecem melhor desempenho de trabalho, enquanto versões anteriores de sistemas operacionais podem estar instaladas em máquinas virtuais hospedadas, não interferindo em nada no ambiente original.

Isso permite que se possam incluir no processo de testes, os testes iniciais em configurações que estão disponíveis aos usuários, para os quais os processos de atualização de hardware e software são feitos com ciclos de vida mais longos, minimizando os efeitos de incompatibilidade entre versões de sistemas operacionais e *browsers* normalmente encontrados nas organizações.

A configuração das estações de trabalho, das equipes de testes e dos desenvolvedores em pequenos testes iniciais, pode ser customizada de acordo as necessidades dos testes.

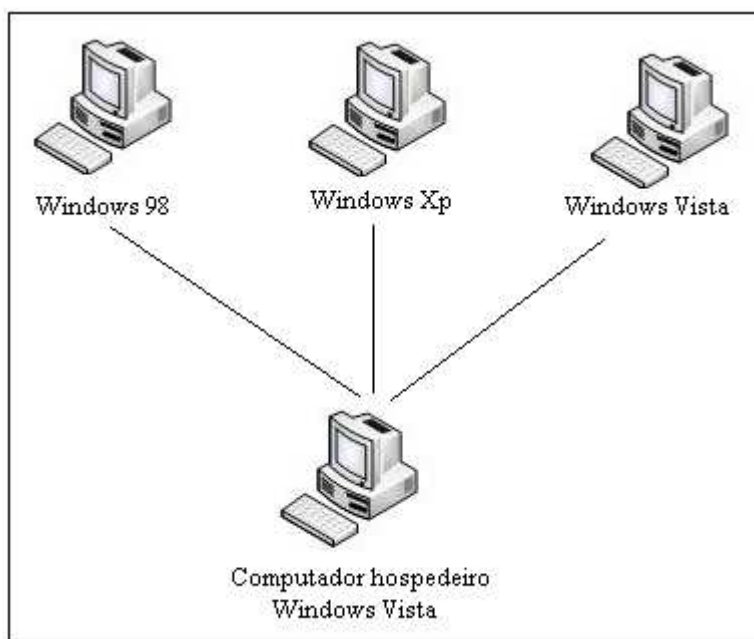


Figura 4 Ambiente de teste virtualizado

A Figura 4 representa um exemplo de um computador hospedeiro utilizando-se do sistema operacional Windows Vista, e ao mesmo tempo virtualizado três ambientes completos para efeitos de testes, Windows 98, Windows XP Service Pack 2 e Window Vista.

Estas máquinas virtuais devem ser preferencialmente clones de máquinas virtuais pré-configuradas com a configuração inicial desejada para testes. Este processo de clonagem é facilitado pela possibilidade de cópia de matrizes dos arquivos que contém todo o empacotamento necessário para a utilização de uma máquina virtual cuja manutenção fica a cargo de uma equipe responsável pela qualidade do processo de testes.

Estas matrizes devem ser protegidas em um meio de repositório das equipes de testes, pois caso seja danificado uma máquina virtual, pode-se fazer nova cópia dos arquivos necessários para a máquina hospedeira e reconfigurar o sistema hóspede de acordo com suas necessidades de testes.

Além do ambiente, para demonstrar a realização de testes utilizando virtualização foi necessário um software de simulação de ambientes computacionais.

Tal software utiliza-se da técnica de virtualização para criar vários ambientes computacionais completos dentro de um único computador físico.

Neste caso foi utilizado o Virtual PC da Microsoft. A escolha deste deu-se pelo fato do autor desta pesquisa já conhecer e estar desenvolvendo este estudo há algum tempo.

A Tabela 1 mostra uma breve comparação entre os dois mais conhecidos simuladores do mercado, dos quais é possível observar que os mesmos estão em patamares equivalentes de funcionalidade.

Tabela 1. Comparativo entre simuladores

<b>Descrição</b>	<b>Virtual Pc</b>	<b>VMWare</b>
Tamanho arquivo para download	30.4 mb	275.8 mb
Facilidade na instalação	✓	✓
<b>Instalação de ambientes</b>		
Utiliza imagens ISO	✓	✓
Possibilidade de configuração BIOS e Setup	✓	✓
<b>Configurações Ambientes</b>		
Aceita configuração memória	✓	✓
Aceita configuração configurar HD	✓	✓
Aceita configuração de CD-ROM/DVD-ROM	✓	✓
Aceita configuração Driver Diskete	✓	✓
Aceita configuração portas COM	✓	✓
Aceita configuração portas LPT	✓	✓
Aceita configuração rede	✓	✓
Aceita configuração som	✓	✓
Aceita configuração Mouse	✓	✓
Aceita configuração pastas de trocas de arquivos	✓	✓
Aceita configuração monitor	✓	✓
Inicialização de vários ambientes simultâneos	✓	✓
<b>Sistemas Operacionais aceitos como ambientes virtuais</b>		
Windows	✓	✓
Linux	✓	✓
Mac	✓	✓
Solaris		✓
FreeBSD		✓
<b>Sistemas Operacionais no qual podem ser instalados</b>		
Windows	✓	✓
Linux		✓
Mac		✓

O software Virtual PC é disponibilizado gratuitamente pela Microsoft.

O Apêndice A trata especificamente sobre como baixar, instalar, configurar e criar máquinas virtuais utilizando o Virtual PC.

Após a escolha, instalação, configuração e instalação das máquinas virtuais, é necessário definir quais testes utilizar.

A utilização da técnica de caixa branca inviabilizaria a execução desta pesquisa pelo fator da necessidade de conhecimentos sobre rotinas utilizadas no software a ser testado.

A técnica de caixa preta é utilizada de forma a fornecer os dados de entrada ao software a ser testado a fim de executar o teste e comparar o resultado obtido com o resultado esperado, esse último, previamente conhecido. Desta forma, haverá sucesso no teste se o resultado obtido for igual ao resultado esperado.

A utilização da técnica de caixa-preta dá-se em conjunto com a fase de teste de compatibilidade, desta forma é possível verificar se o software a ser testado está realizando as mesmas funções nos mais variados ambientes computacional.

A realização dos testes nesta pesquisa é demonstrada em diferentes sistemas operacionais utilizando virtualização, a fim de demonstrar que é possível utilizar virtualização para testar compatibilidade de software.

A escolha dos sistemas operacionais aconteceu de forma simples. Utilizar versões diferentes de sistemas operacionais.

Desta forma, os sistemas operacionais escolhidos foram Windows 98, pois segundo a própria Microsoft esta versão não tem mais suporte a partir de 11 de julho de 2006. Windows XP, é o Sistema Operacional sucessor do Windows 98 e Windows Vista.

A disponibilização dos ambientes com estes sistemas operacionais deve-se a necessidade de instalação e configuração dos mesmos, estes procedimentos estão detalhados no Apêndice A.

O hardware utilizado nesta pesquisa foi um processador Intel Pentium D 3.0 GHz, 1MB *cache*, placa de rede Intel Pro/100 *Ethernet*, memória total 4GB e disco rígido Samsung SP080ZN, 80GB, IDE.

A utilização de vários ambientes computacionais simultâneos depende especificamente da capacidade de processamento do computador hospedeiro.

A cada novo ambiente virtual em execução os recursos do computador hospedeiro são divididos com o ambiente virtualizado.

Foram escolhidos dois softwares. O primeiro será denominado SOFT\_VENDAS apenas para nomenclatura desta pesquisa. A escolha desde deu-se pelo fato da necessidade da realização de testes de compatibilidade nas três versões de sistemas operacionais citados anteriormente, Windows 98, Windows XP e Windows Vista.

O segundo software será denominado de SOFT\_LEGADO apenas para nomenclatura desta pesquisa. A escolha deste deu-se por tratar-se de um software considerado software legado, pois o mesmo utilizou técnicas de programação hoje consideradas ultrapassadas. A utilização do SOFT\_LEGADO se dará no estudo de caso apresentado no capítulo de estudo de caso.

A seguir é demonstrada a realização dos testes em conjunto com a técnica de virtualização.

## 4.2 TESTES

Depois de concluído a fase inicial dos testes, que compreende na escolha e configuração do ambiente virtual, escolha dos softwares a serem testados e técnicas de testes, têm-se então os testes propriamente ditos.

### 4.2.1 SOFT\_VENDAS

Seguindo a metodologia descrita, a realização de testes com o software denominado SOFT\_VENDAS, aconteceu nos ambiente Windows 98, XP e Windows Vista.

O procedimento aconteceu de forma a inicializar o passando como parâmetros de entrada o caminho de uma base de dados tal qual o sistema deverá se conectar, o usuário e a senha.

Desta forma, os testes inicialmente serão realizados sobre os sistemas operacionais Windows 98, Windows XP e Windows Vista, todos estes são ambientes virtualizados demonstrando que é possível unir virtualização e teste de compatibilidade de software.

A Figura 5 demonstra que o teste foi realizado dentro do ambiente virtual Windows 98, sendo que o sistema operacional do computador físico é Windows XP.

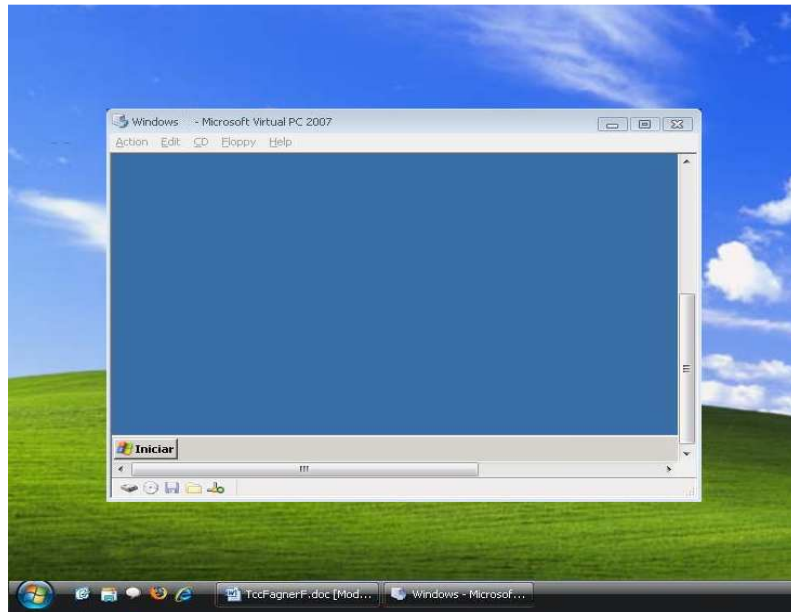


Figura 5 Windows 98 virtualizado sobre o Windows XP

A Figura 6 demonstra o erro encontrado com o ambiente virtual Windows 98. Desta forma, é possível comprovar que o SOFT\_VENDAS não é compatível com Windows 98.



Figura 6 Erro no teste com ambiente Windows 98 virtualizado

Além de demonstrar o erro encontrado no SOFT\_VENDAS, é possível comprovar que a virtualização neste caso auxiliou na realização dos testes, pois foi

possível utilizar um ambiente virtual e comprovar que o software tem incompatibilidades nesta versão de Sistema Operacional.

O mesmo teste foi realizado sobre o ambiente Windows XP. Conforme mostra a Figura 7, o teste neste ambiente virtual não apresentou erro, sendo que o software de uma forma inicial é compatível com o sistema operacional Windows XP.



Figura 7 Teste com ambiente Windows XP virtualizado

Para finalizar esta parte de testes o SOFT\_VENDAS foi submetido à testes sobre o ambiente virtualizado Windows Vista.

A Figura 8 demonstra uma mensagem de erro ao se inicializar o software com os mesmos parâmetros de entrada dos ambientes Windows 98 e Windows XP.

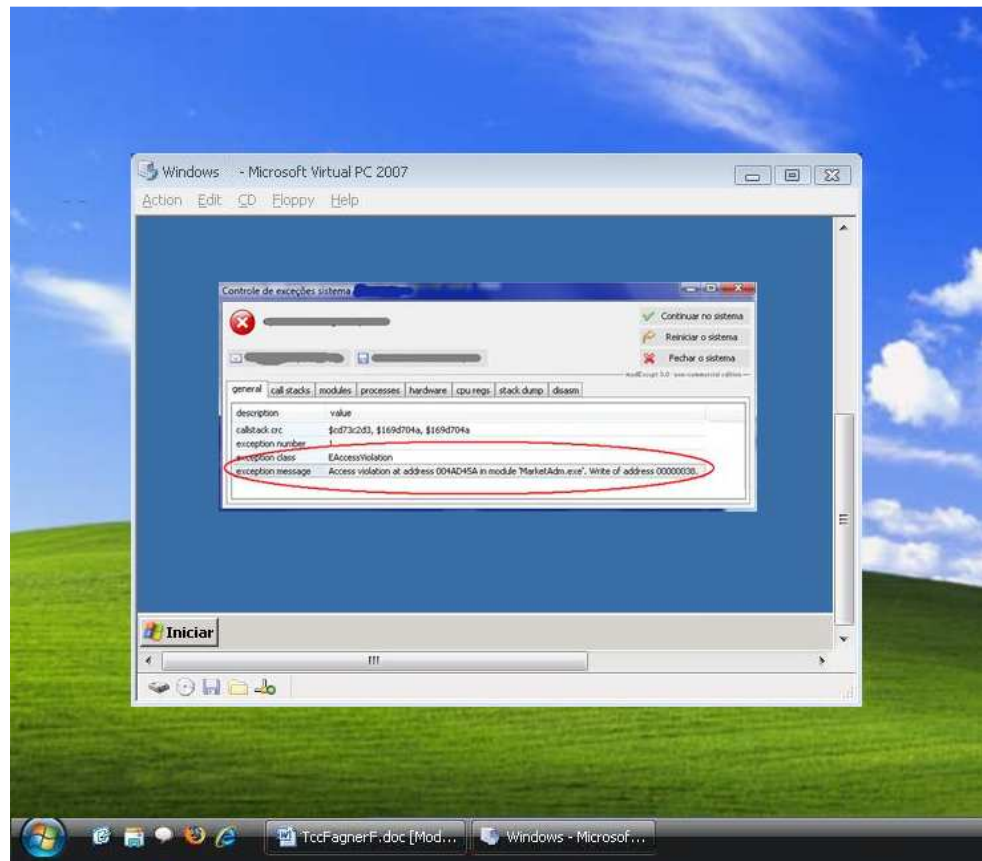


Figura 8 Erro no teste com ambiente Windows Vista virtualizado

O erro apresentado é “*Access violation*”.

Nestes casos de erros antes mesmo que a versão fosse liberada para o cliente final poder-se-ia ser testada em três ambientes virtuais, sem a necessidade de se reiniciar computador para carregar outro sistema operacional, sem precisar trocar de computador. O teste foi realizado com apenas alguns cliques do mouse, pois o Virtual PC da Microsoft cria ambientes virtuais completos dentro de uma única janela do sistema operacional hospedeiro.

Além do teste de compatibilidade de sistema operacional, a utilização de virtualização também permite validar a compatibilidade do software para diferentes configurações de hardware.

O exemplo a seguir demonstra que é possível de forma rápida e prática alterar a quantidade de memória de um ambiente virtual, a fim de se testar a compatibilidade do software utilizando uma determinada quantidade de memória.

A Figura 9 mostra a configuração da capacidade de memória do ambiente virtual Windows 98.

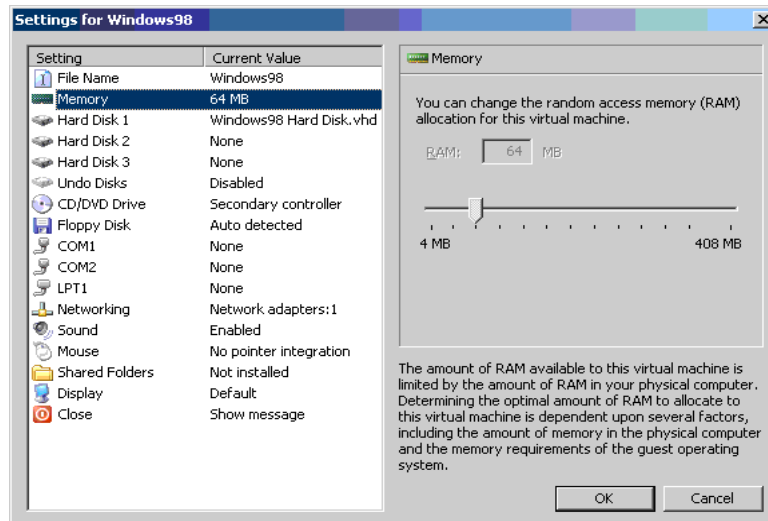


Figura 9 Configurando memória do ambiente virtual Windows 98  
Fonte: MICROSOFT (2007)

Depois de configurado a memória do ambiente virtual, caso o ambiente esteja em execução, é necessário a sua reinicialização para que as configurações sejam aplicadas. Depois da reinicialização é possível observar por meio da Figura 10 que o ambiente virtual já assumiu a configuração de memória selecionada anteriormente.

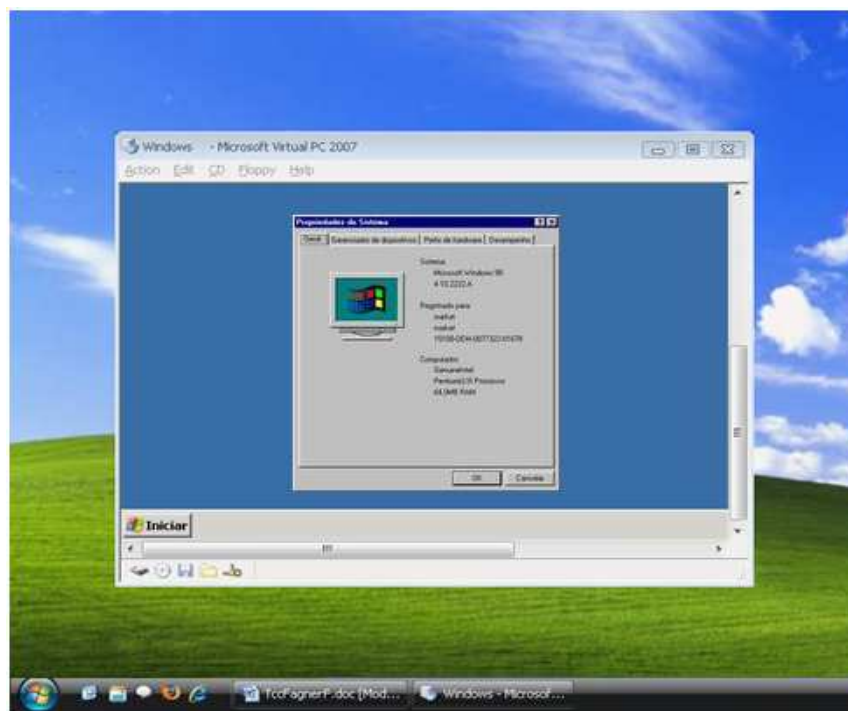


Figura 10 Ambiente virtual Windows 98  
Fonte: Microsoft (2007)

Desta forma é possível efetuar o teste de compatibilidade no SOFT\_VENDAS sobre o ambiente Windows 98 utilizando 64 megabytes de memória RAM, conforme demonstra Figura 11.



Figura 11 Teste com ambiente Windows 98 virtualizado

Apesar da tela inicial do SOFT\_VENDAS estar sendo exibida, o teste com o mesmo apresentou uma anomalia, o mesmo ficou travado por aproximadamente 5 minutos. Acredita-se que esta anomalia seja causada devido ao fato do mesmo utilizar-se de muitas bibliotecas de comunicação com equipamentos de terceiros, como balança, impressora fiscal, desta forma há a necessidade de uma maior capacidade de memória para a sua execução.

Aproveitando o ensejo, pelo fato do SOFT\_VENDAS apresentar a necessidade de comunicação com uma grande quantidade de biblioteca de terceiros, a realização dos testes de compatibilidade de software neste caso torna-se ainda mais

necessário, e a utilização da virtualização apresenta uma possibilidade maior de melhoria no processo de testes.

Supondo que o SOFT\_VENDAS necessite fazer comunicação com dois bibliotecas de fabricantes diferentes.

Cada biblioteca tem uma versão específica para cada sistema operacional, ou seja, neste caso duas bibliotecas versus três sistemas operacionais neste caso de testes, dariam seis combinações de testes de compatibilidade.

Supondo que cada teste levaria cerca de 5 minutos para configuração e inicialização do sistema operacional desejado, tem-se 30 minutos de testes iniciais.

Com a utilização do ambiente virtual, estas configurações poderiam cair em aproximadamente 75% do tempo. Pois o ambiente já estaria configurado previamente de acordo com a necessidade diária de cada caso. Bastariam apenas alguns cliques para que o ambiente virtual estivesse disponível no próprio computador do desenvolvedor ou da equipe de teste.

Supondo ainda que, para cada versão de sistema operacional, tivesse a necessidade de se testar o software com as chamadas subversões, ou ainda pacotes de atualizações, como é o caso do Windows XP Service Pack 1, 2 e 3.

Nestes casos, o tempo de configuração, inicialização e realização dos testes subiriam em patamares algumas vezes insuportáveis.

A utilização da virtualização para testes de compatibilidade nestes casos adentraria como uma alternativa prática e livre de custos.

### 4.3 RESULTADOS OBTIDOS

Por meio da união do teste de compatibilidade de software com a virtualização, como demonstrado no capítulo anterior, é possível realizar de compatibilidade de software nos mais diferentes ambientes computacionais.

Configurar vários ambientes computacionais utilizando-se ferramentas de virtualização é uma tarefa relativamente fácil, a fim de realizar a maior quantidade de testes de compatibilidade de software, buscando assim minimizar os possíveis erros que poderia ocorrer quando o software estivesse em produção.

A união da técnica de virtualização com teste de compatibilidade de software resultou ainda na agilidade com que os testes podem ser realizados, ao invés de se utilizar um computador com diversos ambientes computacionais instalados e ficar efetuando logouts e logins nestes ambientes acarretando um detrimento no tempo, a utilização de uma ferramenta para virtualização torna o processo mais rápido, sendo possível acessar diversos ambientes computacionais com apenas alguns cliques.

Além da possibilidade de se testar o software em diversos sistemas operacionais, quase que simultaneamente, a virtualização ainda permite a rápida configuração de periféricos, como HardDisk e Memória, além de outros, de forma a facilitar no momento de testar a compatibilidade do software com diferentes configurações de hardware.

O capítulo anterior demonstra a redução da capacidade de memória RAM do ambiente virtual, a fim de validar a compatibilidade do software testado com o mínimo de memória possível.

Além dos resultados obtidos nesta pesquisa a virtualização ainda traz como advindo algumas vantagens na sua utilização.

#### 4.4 VANTAGENS DO USO DE MÁQUINAS VIRTUAIS

Dentre as vantagens que se pode obter através da utilização de máquinas virtuais, Blunden (2002) cita:

- a) Aperfeiçoamento e testes de novos sistemas operacionais;
- b) Executar diferentes sistemas operacionais sobre o mesmo hardware, simultaneamente;
- c) Simular configurações e situações diferentes do mundo real, como por exemplo, mais memória disponível, outros dispositivos de E/S;
- d) Simular alterações e falhas no hardware para testes ou re-configuração de um sistema operacional, provendo confiabilidade e escalabilidade para as aplicações;
- e) Garantir a portabilidade das aplicações legadas (que executariam sobre uma VM simulando o sistema operacional original);
- f) Desenvolvimento de novas aplicações para diversas plataformas, garantindo a portabilidade destas aplicações;
- g) Diminuir custos com hardware;
- h) Sustentação de softwares legados implementando um subconjunto de funcionalidades na camada do monitor de máquina virtual, as aplicações com legado poderiam ser suportadas, provendo a eficiência que um sistema anfitrião oferece.
- i) Portabilidade, desenvolvimento de novas aplicações para diversas plataformas, garantindo a portabilidade destas aplicações.

Porém, a virtualização não traz somente vantagens na sua utilização.

#### 4.5 DESVANTAGENS DO USO DE MÁQUINAS VIRTUAIS

Dentre as desvantagens com a utilização de máquina virtual a principal se destaca no custo adicional de execução dos processos na máquina virtual em comparação com a máquina real.

Porém, pesquisas recentes têm obtido a redução desse custo a patamares abaixo de 20%, graças, sobretudo a ajustes no código do sistema anfitrião (DUNLAP et al. 2003). Esse problema não existe em ambiente cujo hardware suporta o conceito de virtualização, como é o caso de mainframes.

Além da realização de testes de compatibilidade de software, esta pesquisa realiza um estudo de caso para apresentar uma alternativa para manter em funcionamento softwares legados com as novas tecnologias.

#### 4.6 ESTUDO DE CASO

O estudo de caso apresentado neste capítulo tem por objetivo mostrar como a virtualização possibilitou a resolução de um problema que se baseia praticamente na seguinte premissa: incompatibilidade entre softwares e sistemas operacionais.

Este estudo de caso utilizou o SOFT\_LEGADO como base para a realização de testes em softwares legados utilizando-se virtualização para solucionar o problema.

#### **4.6.1 Terminologias**

A organização do ramo mercadologista será denominada como ORGMER, pois a mesma não autorizou a divulgação de seu nome.

Da mesma forma, os softwares de controle de gestão envolvidos neste estudo de caso são de registro exclusivo das organizações desenvolvedoras, e para fim de estudo de caso serão denominados:

**SOFT\_LEGADO:** é o softwares de gestão que está sendo utilizado há 8 anos na organização ORGMER.

Em contato com as organizações envolvidas nesta problemática, ORGMER e a proprietária do SOFT\_LEGADO, as mesmas autorizaram a realização do estudo e a sua divulgação nesta pesquisa, desde que os mesmo não tivessem os nomes divulgados para não afetar a imagem do produto e nem da organização.

#### **4.6.2 Histórico**

Em uma organização do ramo mercadologista denominada ORGMER, situada na região de Criciúma-SC, a utilização do software de controle de gestão, o chamado SOFT\_LEGADO, é de extrema importância. É com a utilização deste software que a organização gerencia a maior parte dos processos de sua vida diária

O SOFT\_LEGADO esta sendo utilizado pela organização desde há oito anos e está sendo executados em quatro computadores, todos com sistema operacional Windows 98.

Porém com a necessidade de expansão da ORGMER foram adquiridos dez novos computadores com o sistema operacional Windows XP.

O problema ocorreu no momento de se efetuar a instalação do SOFT\_LEGADO nestes novos computadores. O software foi projetado apenas para ser executado em versões de Windows 98 ou abaixo.

A ORGMER ficou enlaçada no seu planejamento de expansão.

Mediante ao problema descrito e após uma análise detalhada de todo os procedimentos da organização, a solução encontrada foi a virtualização de sistema operacional, de forma a atender rapidamente e com um custo reduzido á problemática citada.

A virtualização ocorrerá de forma que o computador possa executar e manterem-se em funcionamento o SOFT\_LEGADO, podendo atender as solicitações dos usuários o com maior agilidade possível.

#### **4.6.3 Hardware e Software Utilizados**

O hardware utilizado nesta pesquisa foram processadores Intel Pentium D 3.0 GHz, 512MB *cachê*, placa de rede Intel Pro/100 *Ethernet*, memória total 512mb e disco rígido Samsung SP080ZN, 80GB, IDE.

#### **4.6.4 Visão Geral do Processo**

De forma a criar um roteiro para a implantação da solução encontrada, esta secção tem por objetivo mostrar uma visão macro de todo o processo. Este processo será dividido em quatro partes:

- a) Processo de instalação do monitor de ambiente virtual;
- b) Processo de criação do ambiente virtual;
- c) Processo de instalação do ambiente virtual Windows 98;
- d) Processo de instalação e execução do SOFT\_LEGADO.

#### 4.6.5 Processo de Instalação do Monitor de Ambiente Virtual

O processo de instalação do monitor de ambiente virtual consiste em instalar o software Microsoft Virtual PC, detalhado no Apêndice A sobre o ambiente Windows XP.

O software Microsoft Virtual PC atualmente é disponibilizado gratuitamente pela própria Microsoft e sua instalação é simples.

Em alguns passos, como demonstrado na Figura 12 pode-se ter o ambiente instalado pronto para ser usado.

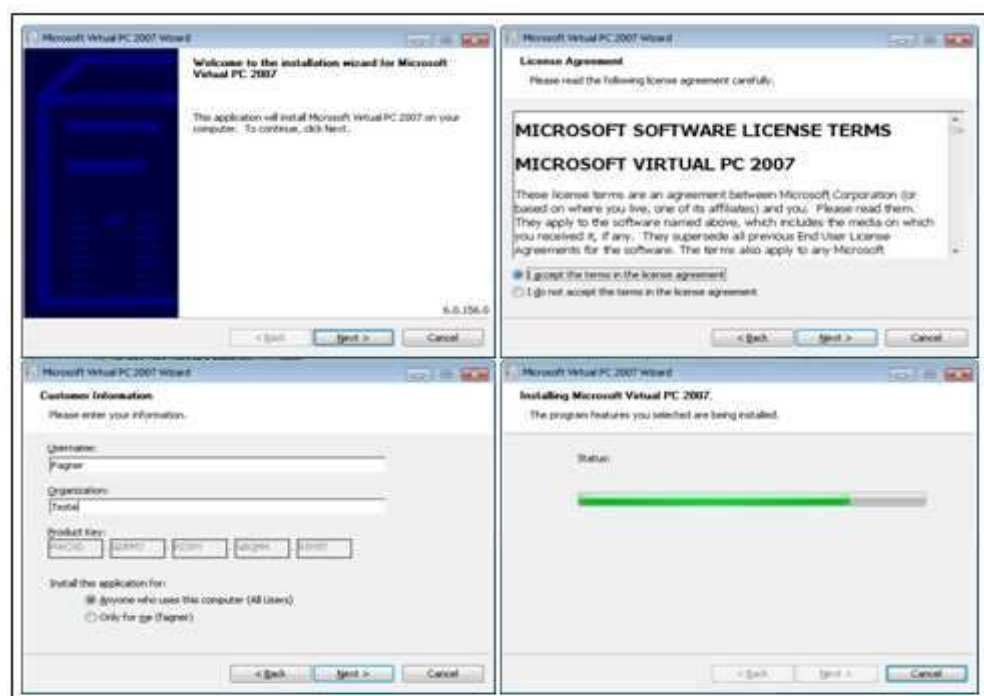


Figura 12 Processos de instalação do Virtual PC  
Fonte: Microsoft (2007)

#### 4.6.6 Processo de Criação do Ambiente Virtual

Através do processo de criação do ambiente virtual é possível criar um novo ambiente para hospedar o sistema operacional Windows 98, tal qual o SOFT\_LEGADO necessita para ser executado. Tal processo consiste em configurar alternativas como: Memória, espaço em disco, Drivers Cd/Dvd, Portas paralelas, portas seriais, dentre outros processos.

O processo de criação de um ambiente virtual completo é detalhado no Apêndice A.

A Figura 13 demonstra alguns processos da criação do ambiente virtual.

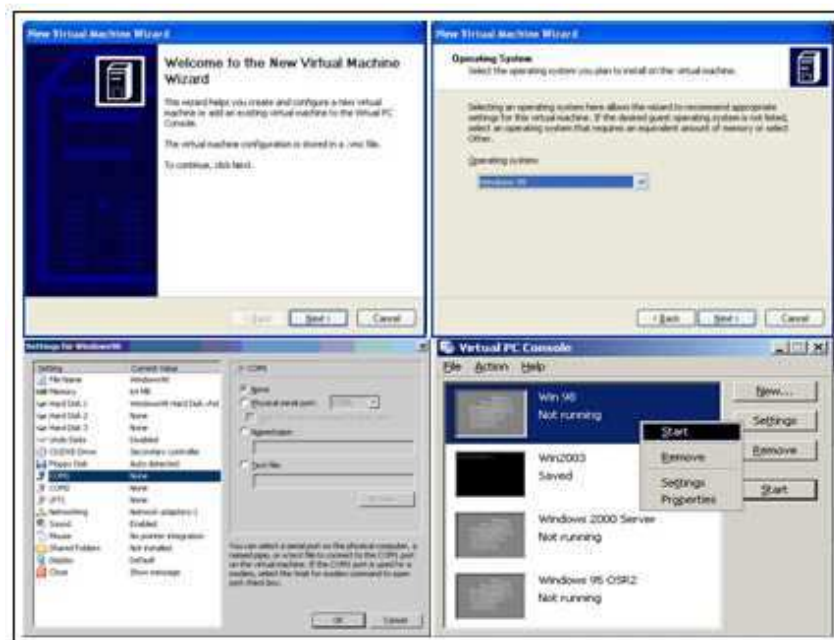


Figura 13 Processos de criação do ambiente virtual  
Fonte: Microsoft (2007)

#### 4.6.7 Processo de Instalação do ambiente virtual Windows 98

Neste processo são realizados todos os passos cuja instalação do sistema operacional em um computador físico requer.

A Figura 14 mostra alguns processos de instalação do sistema operacional.

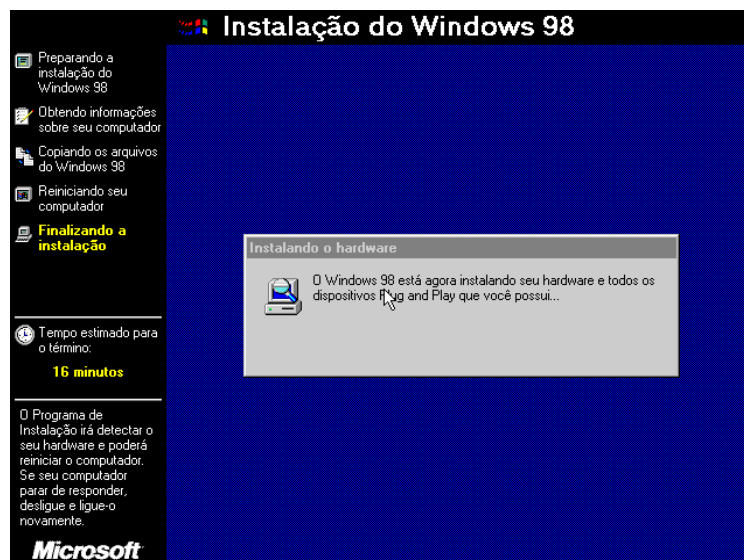


Figura 14 Processos de instalação do ambiente virtual Windows 98  
Fonte: Microsoft (2007)

Além da instalação do sistema operacional, foram realizados alguns processos de otimização a fim de desativar serviços que não seriam necessários para a execução do SOFT\_LEGADO e também para que o processo de *start* do sistema operacional acontecesse de forma mais ágil.

#### 4.6.8 Processo de Instalação e Execução do SOFT\_LEGADO

Este processo consiste em instalar o SOFT\_LEGADO a fim de comprovar a viabilidade de utilização de sistemas virtualizados.

A Figura 15 mostra alguns processos de instalação do SOFT\_LEGADO ocorrendo dentro do ambiente virtualizado Windows 98.



Figura 15 Processos de instalação e execução do SOFT\_LEGADO

A Figura 16 demonstra a utilização do SOFT\_LEGADO dentro do ambiente virtualizado Windows 98. Desta forma é possível manter o SOFT\_LEGADO em funcionamento com as novas tecnologias, como é o caso do Windows XP ou até mesmo Windows Vista.

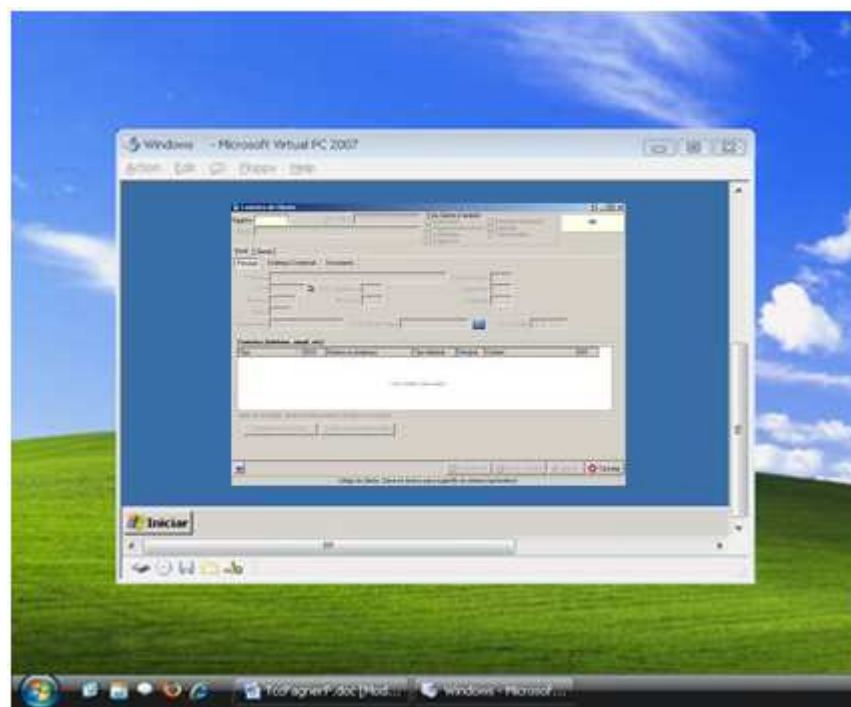


Figura 16 Execução do SOFT\_LEGADO no ambiente Windows 98 Virtualizado

#### **4.6.9 Resultados Obtidos**

O processo de instalação desta solução começou a ser feito em abril de 2008 e atualmente existem aproximadamente dez computadores utilizando esta solução.

Até o presente momento a utilização desta solução de virtualização não apresentou nenhum problema considerado de alto risco para o funcionamento dos dois sistemas no mesmo computador. Pelo contrário, a solução se mostrou estável e completamente funcional não apresentando, até o presente momento, nenhum tipo de anomalia. Sendo que a utilização do SOFT\_LEGADO dá-se de forma igualitária dentro do ambiente virtual quanto no computador que possui sistema operacional Windows 98.

## CONCLUSÃO

Evitar falhas nos software é muito difícil, principalmente com a evolução da tecnologia nos dias atuais. Manter os softwares atualizados com a as novas tecnologias muitas vezes gera um alto custo para as organizações.

Os erros nos softwares são comuns e ao mesmo tempo problemáticos, principalmente nos dias atuais tais quais os ambientes computacionais evoluem constantemente. A cada dia que passa se encontram novos hardwares e softwares utilizando novas tecnologias das quais, os softwares podem não estar apto a confrontá-los.

Desta forma, a fase de teste de compatibilidade de software é de suma importância.

A utilização desta técnica de teste de compatibilidade de software demanda das organizações disponibilidade de diferentes equipamentos e sistemas operacionais para que os testes possam ser realizados.

A técnica de teste de compatibilidade de software foi utilizada em conjunto com a virtualização de ambientes computacionais e resultado obtido foi uma alternativa para as equipes desenvolvedoras e equipes de testes de softwares no momento de realizarem os testes.

Utilizando tal alternativa é possível realizar os testes no software desejado nos mais variados ambientes computacionais, podendo até mesmo ser realizado em um ambiente similar ao qual será executado quando finalizado uma versão. Com a utilização desta técnica também é possível verificar alguns conflitos entre versões de software, principalmente quando se refere a versões de sistemas operacionais diferentes.

Outra alternativa apresentada nesta pesquisa é a utilização da técnica de virtual para as organizações manter em funcionamento os softwares legados com o surgimento de novas tecnologias, através de um estudo de caso.

Os softwares legados estão presentes em várias organizações e muitas vezes representam problemas quase insustentáveis.

Para trabalhos futuros é indicada a explanação de outras técnicas de teste de software em conjunto com a virtualização de ambientes computacionais, bem como o estudo das ferramentas de casos de testes em virtualização de ambientes computacionais.

## REFERÊNCIAS

BARHAM, P.; DRAGOVIC, B.; FRASER, K. HAND, S; HARRIS, T; HO, A; NEUGEBAUER, R; PRATT, I; WARFIEL, A. **Xen and the art of virtualization. In Proceedings of the Nineteenth ACM symposium on Operating systems principles (SOSP), Bolton Landing.** Ny: Acm, 2003.

BARTIE, Alexandre. **Categorias de Testes de Software.** Disponível em: <[http://imasters.uol.com.br/artigo/3648/des\\_de\\_software/categorias\\_de\\_testes\\_de\\_software](http://imasters.uol.com.br/artigo/3648/des_de_software/categorias_de_testes_de_software)>. Acesso em: 02 jun. 2008.

BENNETT, K.H. **Legacy Systems: coping with success.** Los Alamitos: Ieee Computer Society Press, 1995. v.12.

BLUNDEN, B. **Virtual Machine Design and Implementation in C/C++ (Paperback).** Texas: Wordware Publishing, 2002.

BRANDÃO, A. **Tudo pronto? Vamos testar!** Disponível em: <[http://www.timaster.com.br/revista/materias/main\\_materia.asp?codigo=1201](http://www.timaster.com.br/revista/materias/main_materia.asp?codigo=1201)>. Acesso em: 02 jun. 2008.

DUNLAP, G.; KING, S.; CINAR, S.; BASRAI, M.; CHEN, P.. **ReVirt: Enabling Intrusion Analysis through.** Disponível em: <<http://www.eecs.umich.edu/virtual/papers/dunlap02.pdf>>. Acesso em: 02 jun. 2008.

ESTEVAM, Alex. **Qualidade de Software.** Disponível em: <<http://www.abramti.org.br/modules/news/article.php?storyid=34>>. Acesso em: 16 jun. 2008.

GOLDBERG, R. **Survey of Virtual Machine Research.** IEEE Computer Magazine, Vol.7, 1974.

LEACH, Ronald J.. **Introduction to Software Engineering.** Disponível em: <<http://books.google.com/books?hl=pt-R&lr=&id=XW8PIP7sCOkC&oi=fnd&pg=RA1-PA1&dq=LEACH,+Ronald+J.+Introduction+to+Software+Engineering&ots=6w0stY40Jl&sig=zBBhkQE7WQdoeC6isCTkQgrq15g#PRA1-PA53,M1>>. Acesso em: 16 jun. 2008

KIYANCLAR, Nadir. **A Survey of Virtualization Techniques Focusing on Secure On-Demand Cluster Computing.** Disponível em: <<http://arxiv.org/abs/cs/0511010v1>>. Acesso em: 05 maio 2008.

MENON, A., SANTOS, J. R., TURNER, Y., JANAKIRAMAN, G. J., and ZWAENPOEL, W. **Diagnosing performance overheads in the xen virtual machine**

**environment. In VEE '05: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments.** New York: Acm, 2005.

NANDA, Susanta. **A Survey on Virtualization.** Disponível em:  
<<http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf>>. Acesso em: 16 jun. 2008.

PRESSMAN, Roger S. **Engenharia de Software.** São Paulo: São Paulo: Makron Books, 1995.

PRESSMAN, Roger. S. **Engenharia de Software.** São Paulo: McGraw-Hill, 2002

SOMMERVILLE, Lan. **Engenharia de Software.** São Paulo: Addison Wesley, 2003.

SUGERMAN, J., Venkitachalam, G., and Lim, B.-H.. **Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor. In Proc.Usenix Annual Technical Conference.**2001.

TRAYAHÚ, Moreira. **Teste de Software.** Rio de Janeiro: Alta Books, 2006

## **APÊNDICE A – VIRTUAL PC**

### **BAIXANDO O VIRTUAL PC**

O download pode ser encontrado no site da Microsoft através do link:

<http://www.microsoft.com/downloads/details.aspx?familyid=04D26402-3199-48A3-AFA2-2DC0B40A73B6&displaylang=en>

#### **Recursos necessários.**

Segundo Microsoft (2007) o mínimo requerido para o Virtual PC é de:

- a) Windows 2000 com atualização de Service Pack 4; Windows Server 2003, versão Standard (32-bit x86); Windows XP com atualização de Service Pack 2;
- b) processador: Athlon®, Duron®, Celeron®, Pentium® II, Pentium III, ou Pentium 4;
- c) velocidade de processador de 400 MHz no mínimo, porém o recomendado é de 1 GHz ou acima;
- d) memória RAM é a requerida pelo sistema operacional mais adicional para a utilização de cada sistema Operacional que será instalado no Virtual PC;
- e) o mesmo se aplica para a capacidade de armazenamento do HD.

## Instalando.

Após efetuar o download do Microsoft Virtual PC, basta executar o arquivo denominado setup.exe, conforme mostra a Figura 17.



Figura 17 Arquivos baixados do Virtual PC

Em seguida irá aparecer uma tela introdutória. Basta clicar no botão *Next*.

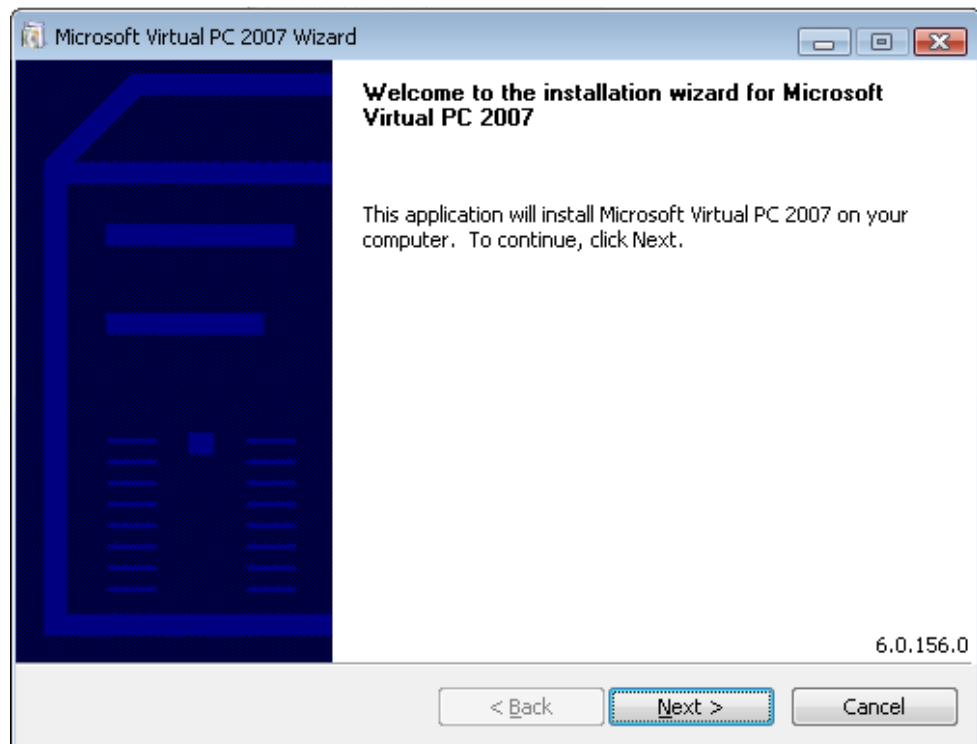


Figura 18 Instalando Virtual PC – Passo 1  
Fonte: Microsoft (2007)

Na tela seguinte será solicitada a confirmação do contrato de licença, basta marcar a opção "*I accept terms in the license agreement*" e clicar em *Next*.



Figura 19 Instalando Virtual PC – Passo 2  
Fonte: Microsoft (2007)

O próximo passo seria informar o serial do produto, porém o Virtual PC 2007 é grátis, e também informar o usuário e a organização, e após clicar em *Next*.

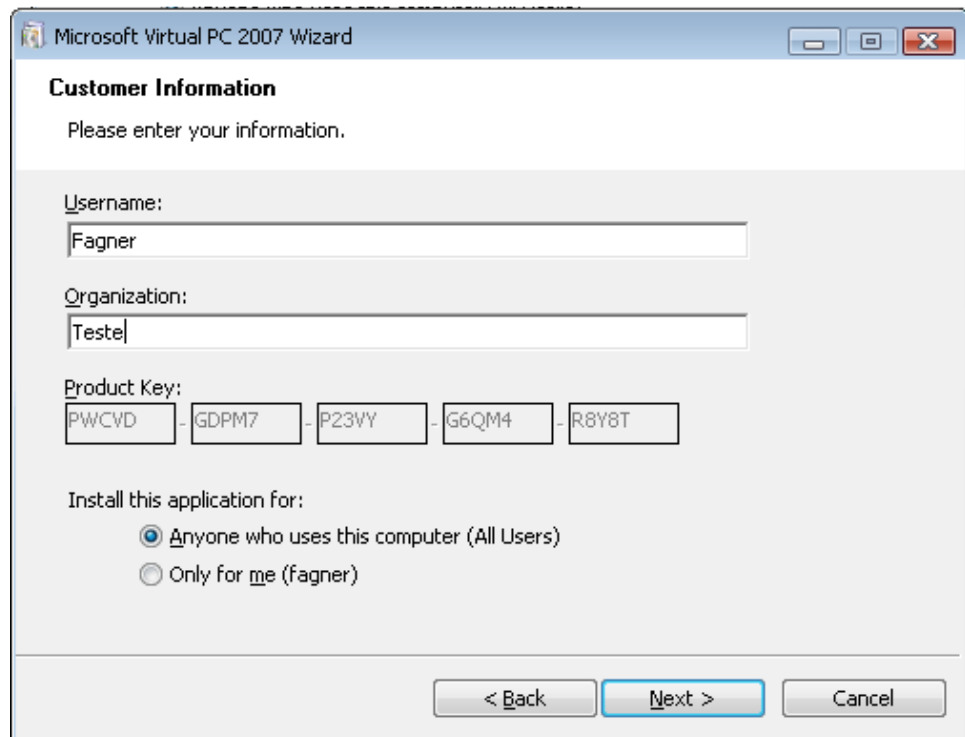


Figura 20 Instalando Virtual PC – Passo 3  
Fonte: Microsoft (2007)

Na tela seguinte o assistente de instalação confirmará o local no disco onde o software vai ser instalado.

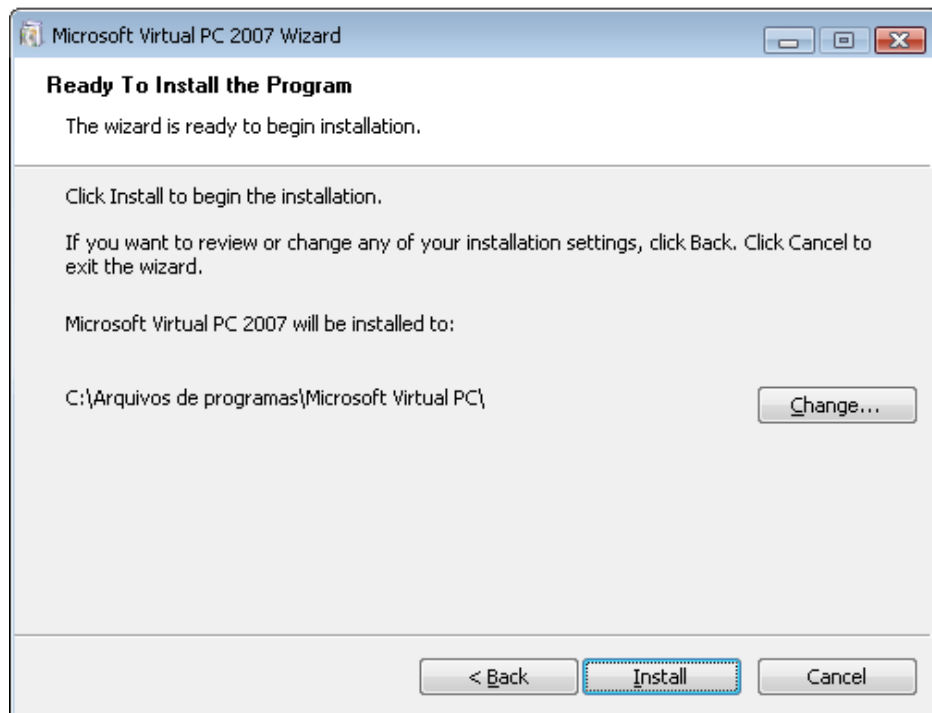


Figura 21 Instalando Virtual PC – Passo 4  
Fonte: Microsoft (2007)

Logo após irá copiar os arquivos de instalação.

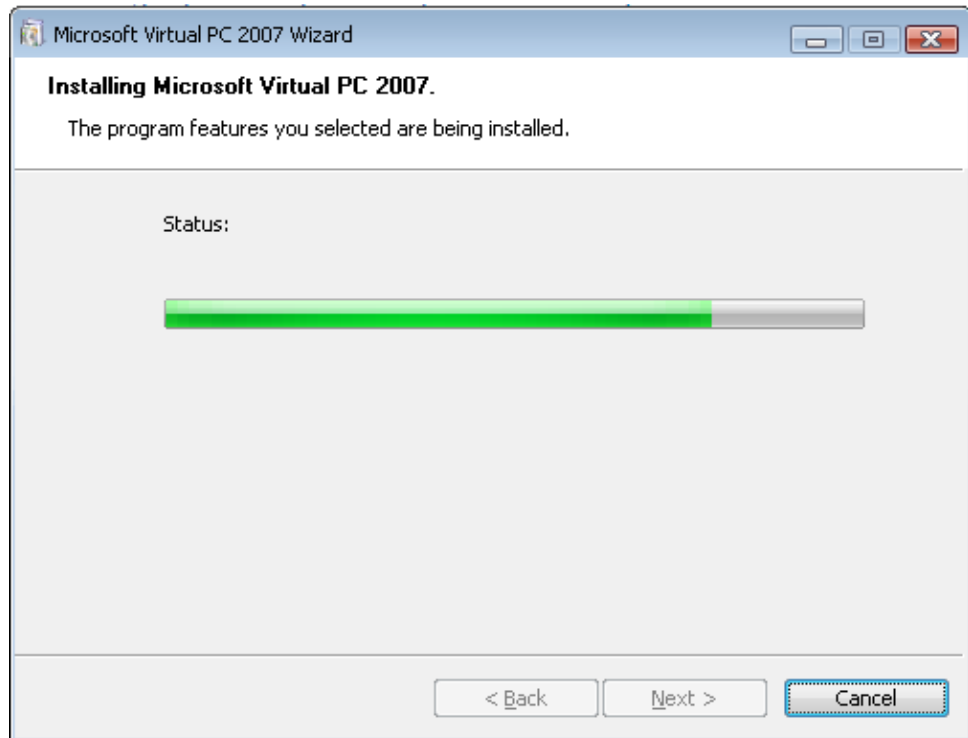


Figura 22 Instalando Virtual PC – Passo 5  
Fonte: Microsoft (2007)

E a última tela da instalação é a informação de completo.

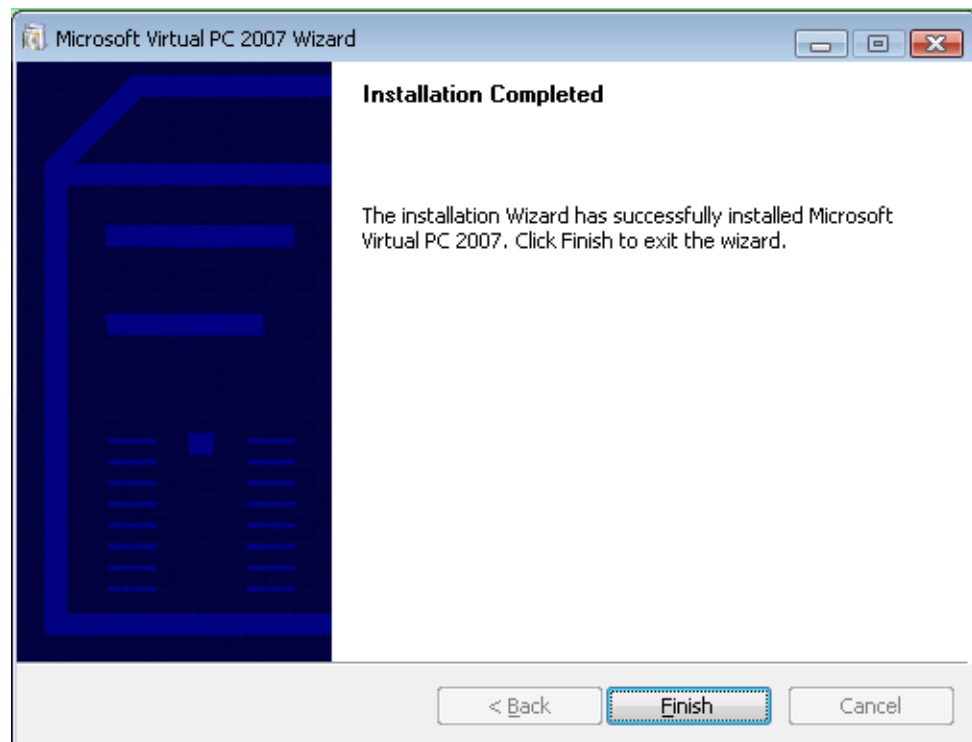


Figura 23 Instalando Virtual PC – Passo 6  
Fonte: Microsoft (2007)

Automaticamente após a instalação o Virtual PC abrirá uma tela onde pedirá para criar um novo ambiente virtual.

Quando se está utilizando o Virtual PC pela primeira vez, tem-se a necessidade da instalação dos ambientes virtuais. Porém após a primeira vez, caso tenha-se necessidade de se reinstalar ou até mesmo instalar em um outro computador, basta copiar os arquivos de cada ambiente e colocar na pasta correta. Este procedimento será explicado nos próximos capítulos.

A Figura 24 demonstra a ultima tela de processo de criação de um ambiente virtual, basta clicar em “*Finish*”.

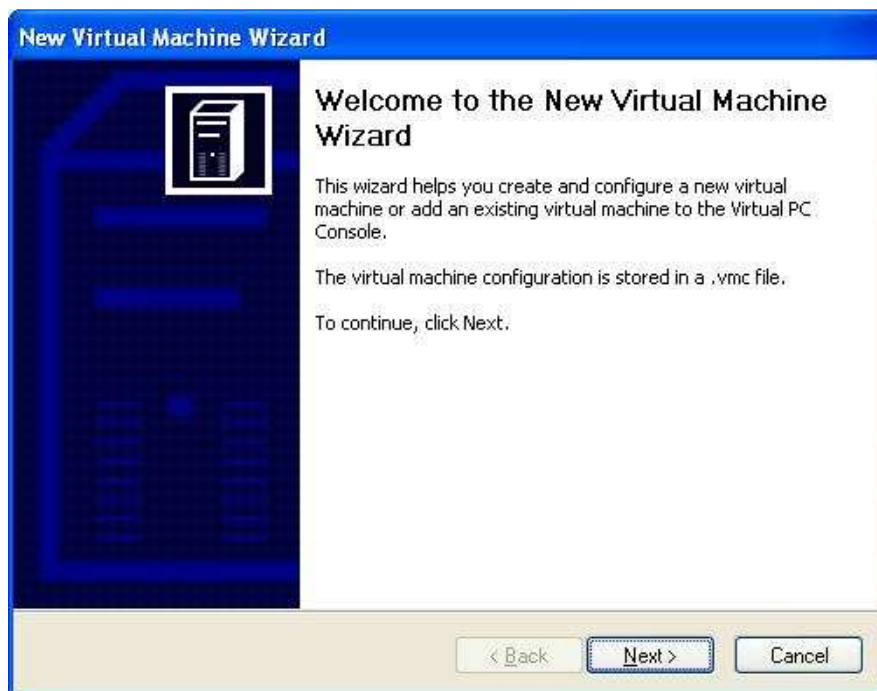


Figura 24 Instalando Virtual PC – Passo 7  
Fonte: Microsoft (2007)

Logo após este processo, será solicitada qual máquina virtual será criada.

- a) **Create a virtual machine** - Esta opção do assistente irá criar uma máquina virtual básica;

- b) **Use default settings to create a virtual machine** - Será criado automaticamente um arquivo “.vnc”, com isso a máquina virtual criada não possuirá um disco virtual, o mesmo devera ser criado posteriormente;
- c) **Add na existing virtual machine** - Será criada uma máquina virtual utilizando um arquivo “.vnc” já existente.

Foi utilizado a primeira opção.

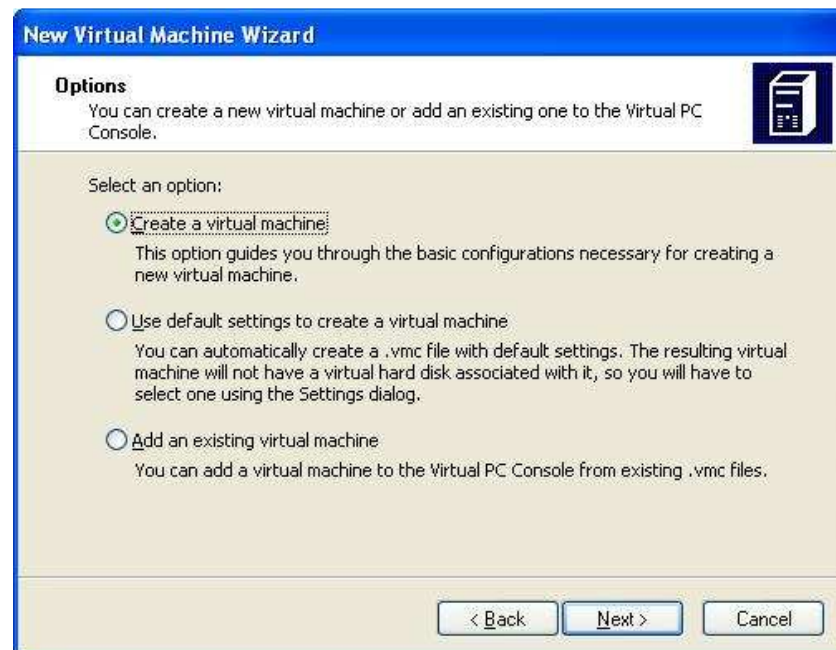


Figura 25 Criando Ambiente Virtual – Passo 1  
Fonte: Microsoft (2007)

Após será solicitado um nome para a máquina virtual.

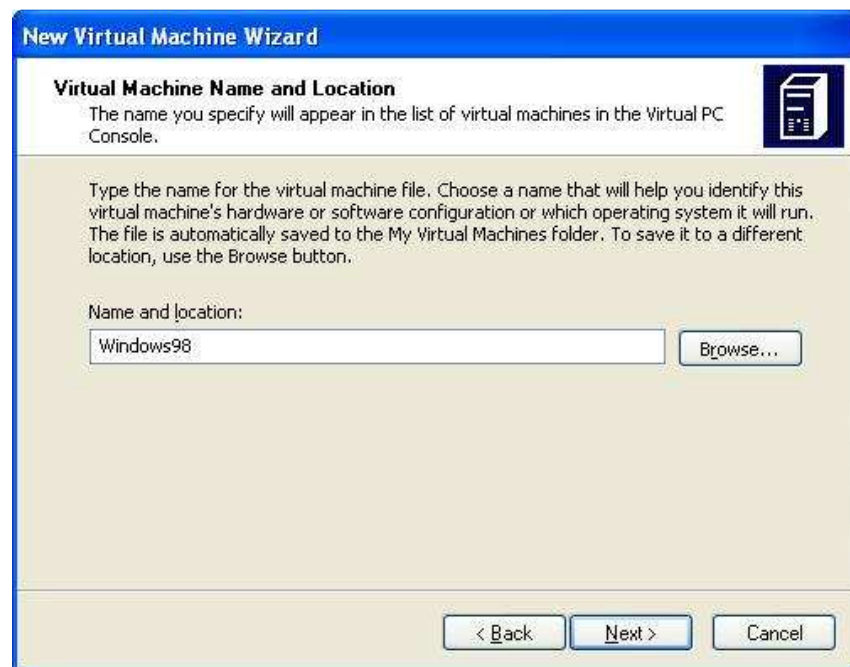


Figura 26 Criando Ambiente Virtual – Passo 2  
Fonte: Microsoft (2007)

E logo em seguida o caminho onde se deseja criar a máquina virtual.

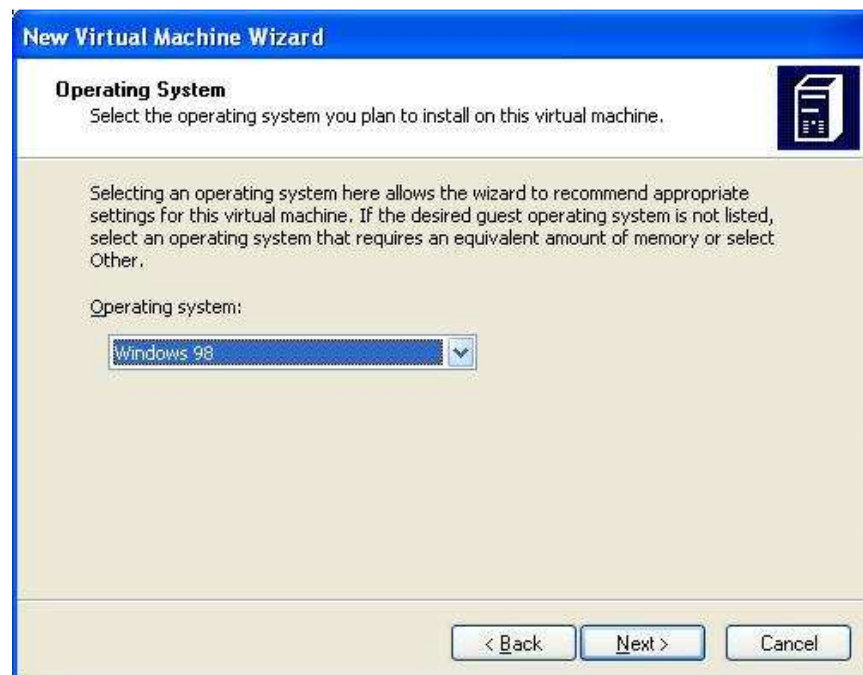


Figura 27 Criando Ambiente Virtual – Passo 3  
Fonte: Microsoft (2007)

Na Figura 28 é demonstrado a opção de selecionar quando de memória RAM será usada no ambiente virtual.

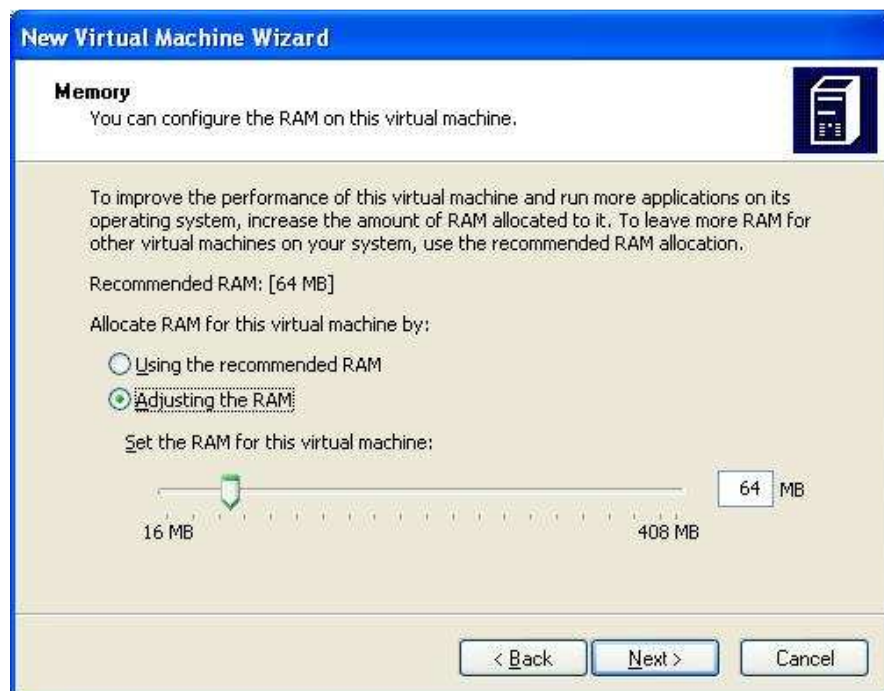


Figura 28 Criando Ambiente Virtual – Passo 4  
Fonte: Microsoft (2007)

Continuando o assistente, será solicitado a criação de um novo disco virtual ou utilizar um disco virtual já existente.



Figura 29 Criando Ambiente Virtual – Passo 5  
Fonte: Microsoft (2007)

Agora será solicitado o caminho para salvar o virtual HD.

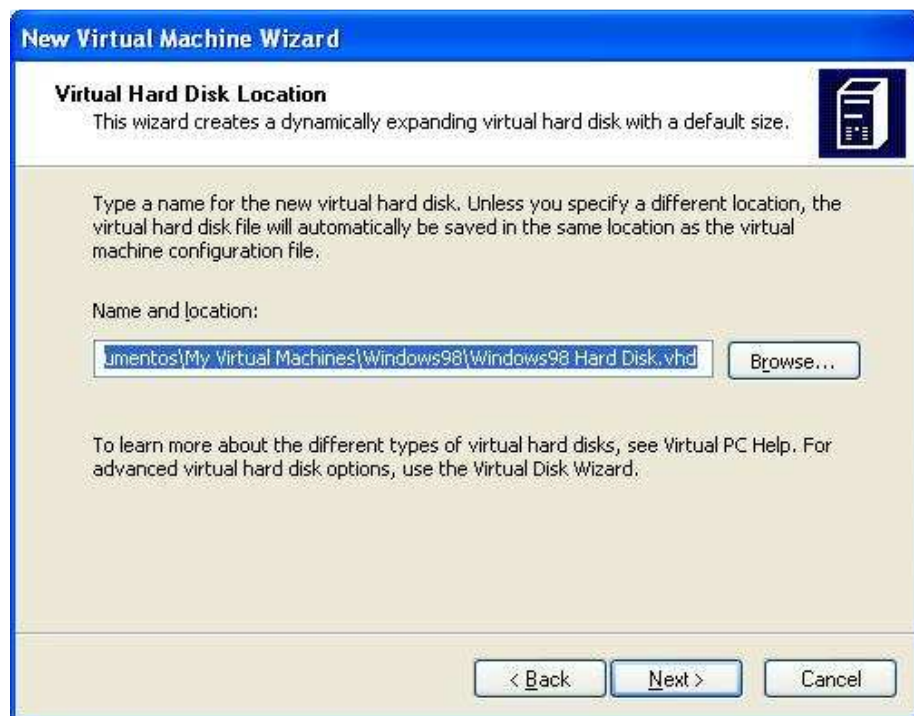


Figura 30 Criando Ambiente Virtual – Passo 6  
Fonte: Microsoft (2007)

E a seguir apenas uma tela informativa da criação da nova máquina virtual.



Figura 31 Criando Ambiente Virtual – Passo 7  
Fonte: Microsoft (2007)

## Configurando o ambiente virtual

Depois de finalizada a parte de instalação do Virtual PC, inicia-se a fase de configuração do ambiente virtual. Algumas das configurações que serão citadas abaixo, só poderão ser configuradas após a instalação completa do ambiente virtual.

A primeira opção encontrada é a do File Name, onde se pode definir um nome para o ambiente virtual. Utilizei Windows98, pois se trata de uma versão chamada assim pela Microsoft.

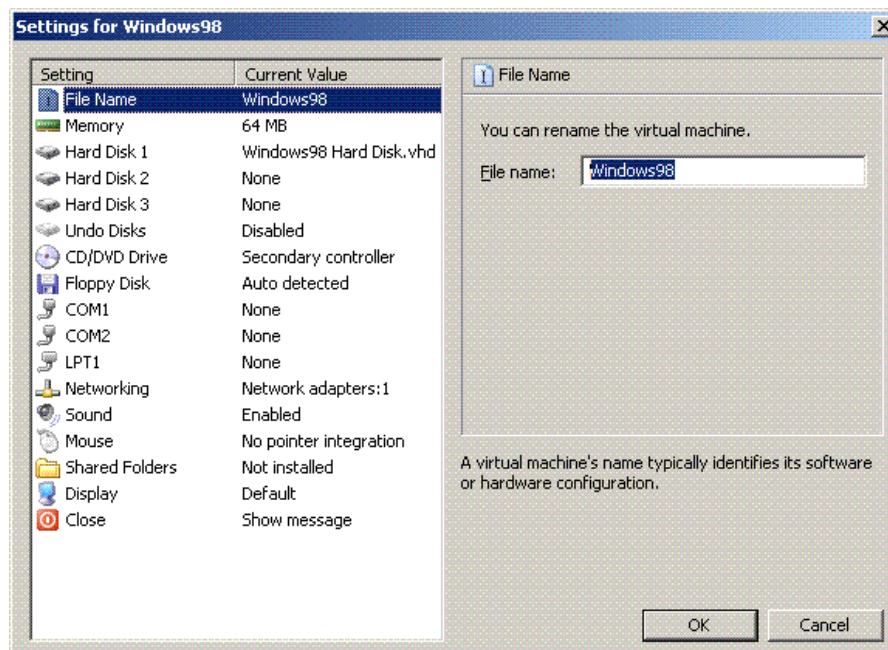


Figura 32 Configurando Ambiente Virtual – Passo 1  
Fonte: Microsoft (2007)

Conforme demonstra a Figura 33 é possível configurar a quantidade de memória que será disponibilizada ao se iniciar o ambiente virtual. A Quantidade mínima e máxima variam de acordo com a capacidade de memória física do computador.

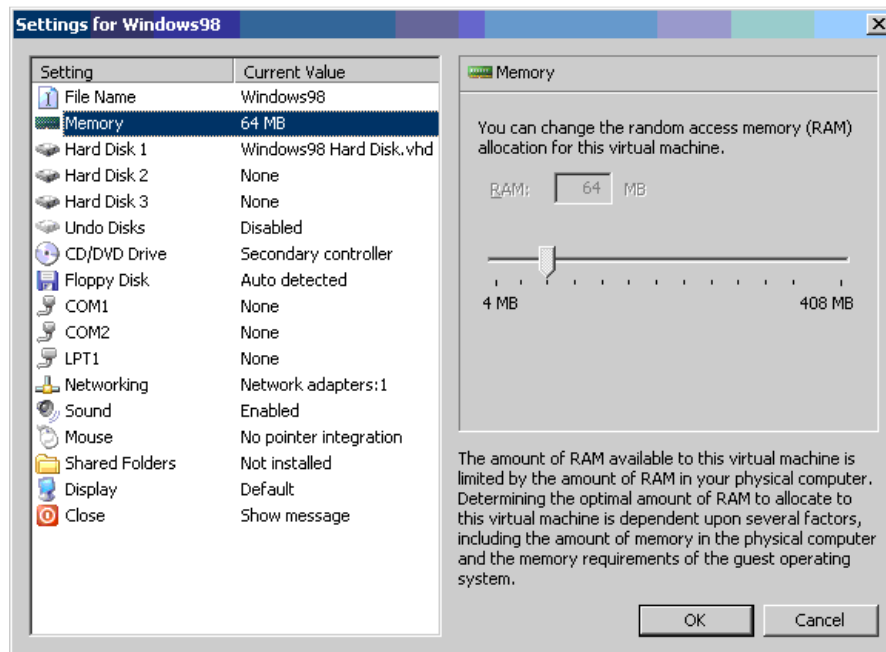


Figura 33 Configurando Ambiente Virtual – Passo 2  
Fonte: Microsoft (2007)

Logo após vem a opção de HD 1. Pode-se criar um disco físico no próprio HD, ou seja, um arquivo onde serão guardados todos os arquivos do HD deste ambiente virtual. Ainda existe a possibilidade de possuir mais de um HD.

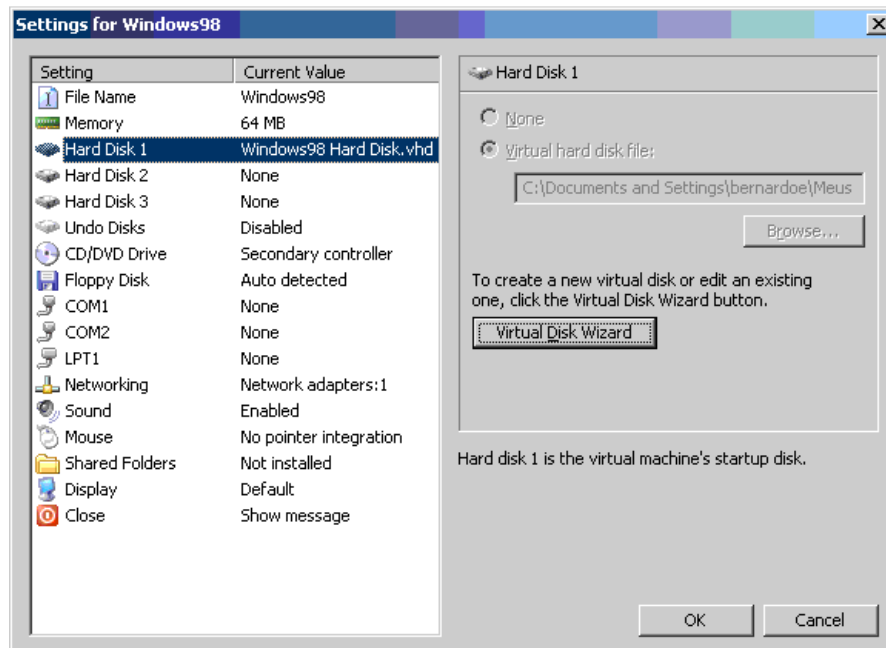


Figura 34 Configurando Ambiente Virtual – Passo 3  
Fonte: Microsoft (2007)

Na opção seguinte é possível configurar *drivers* de CD/DVD, conforme mostra a Figura 35.

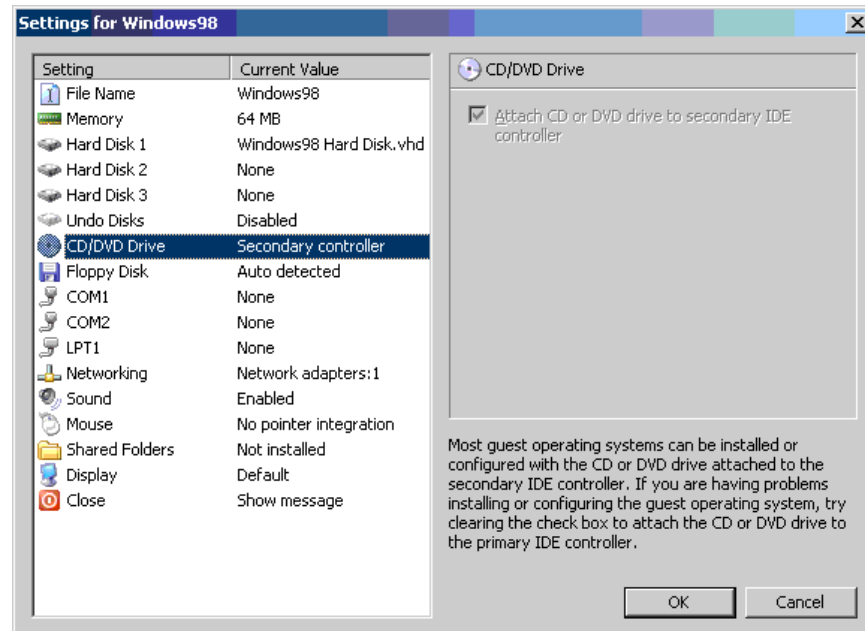


Figura 35 Configurando Ambiente Virtual – Passo 4  
Fonte: Microsoft (2007)

Logo após a opção de disquete, podemos marcar a opção de autodetecção do *driver*.

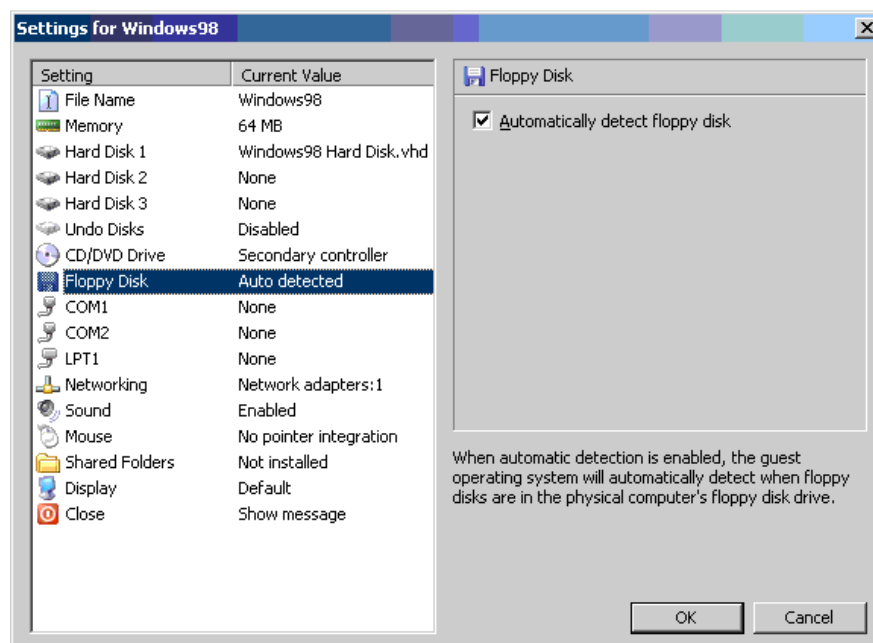


Figura 36 Configurando Ambiente Virtual – Passo 5  
Fonte: Microsoft (2007)

Também se pode configurar as portas COM1, COM2 e LPT1.

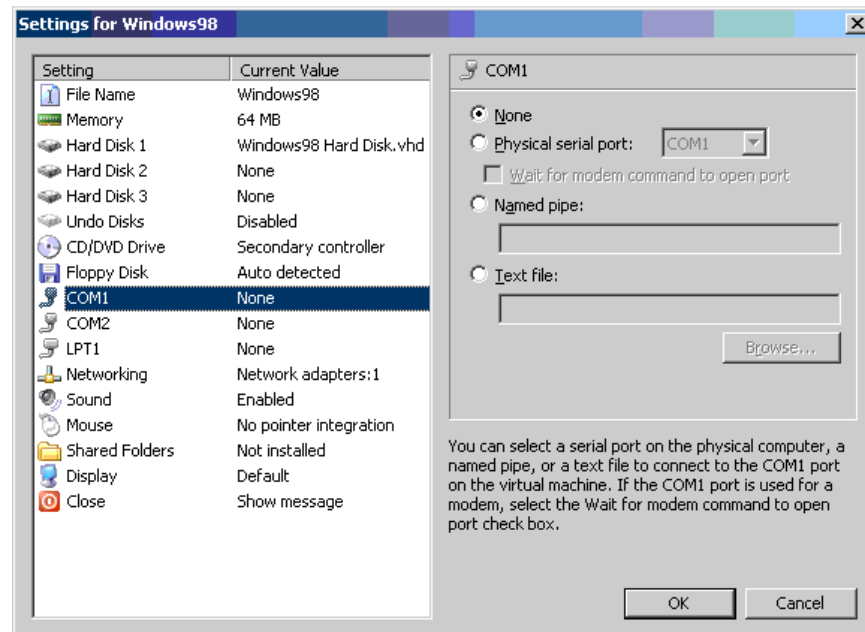


Figura 37 Configurando Ambiente Virtual – Passo 6  
Fonte: Microsoft (2007)

A Figura 38 demonstra que a configuração de rede já foi automaticamente detectada.

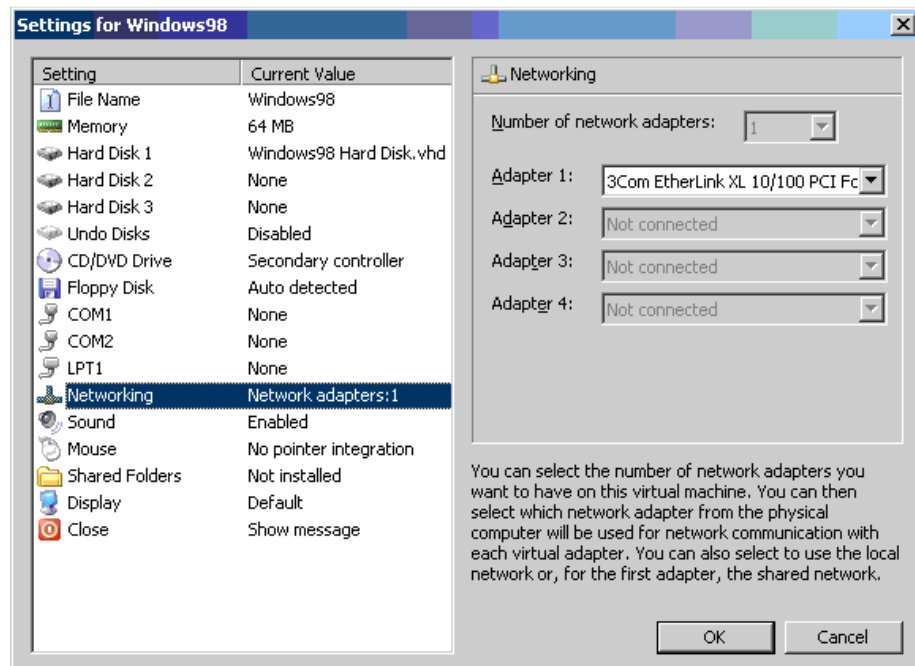


Figura 38 Configurando Ambiente Virtual – Passo 7  
Fonte: Microsoft (2007)

O mesmo ocorre para *drivers* de para som e mouse.

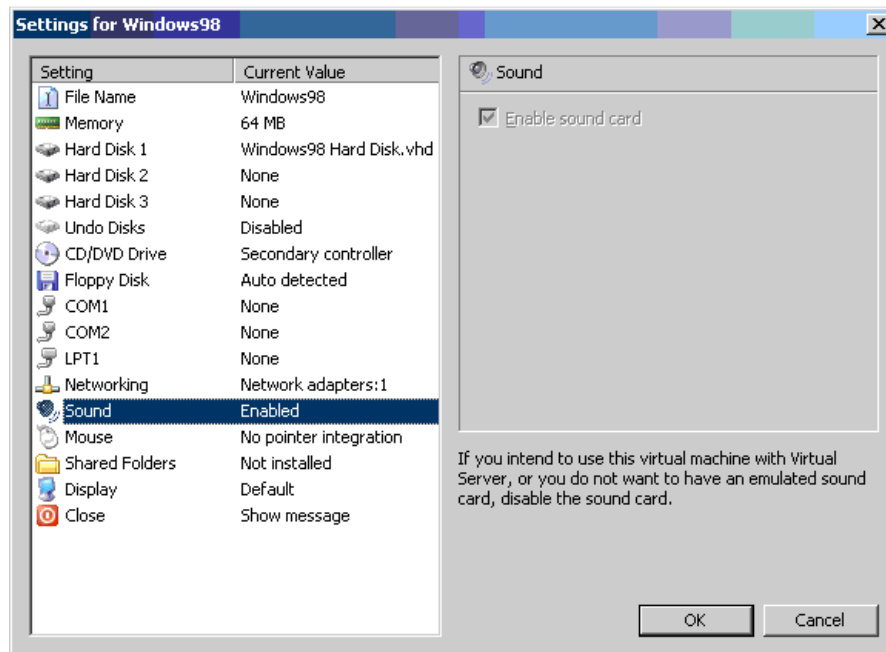


Figura 39 Configurando Ambiente Virtual – Passo 8  
Fonte: Microsoft (2007)

Como mostra a Figura 40 se podem compartilhar determinadas pastas do computador hospedeiro para o ambiente virtual, essa opção só estará habilitada após a inicialização do ambiente virtual.

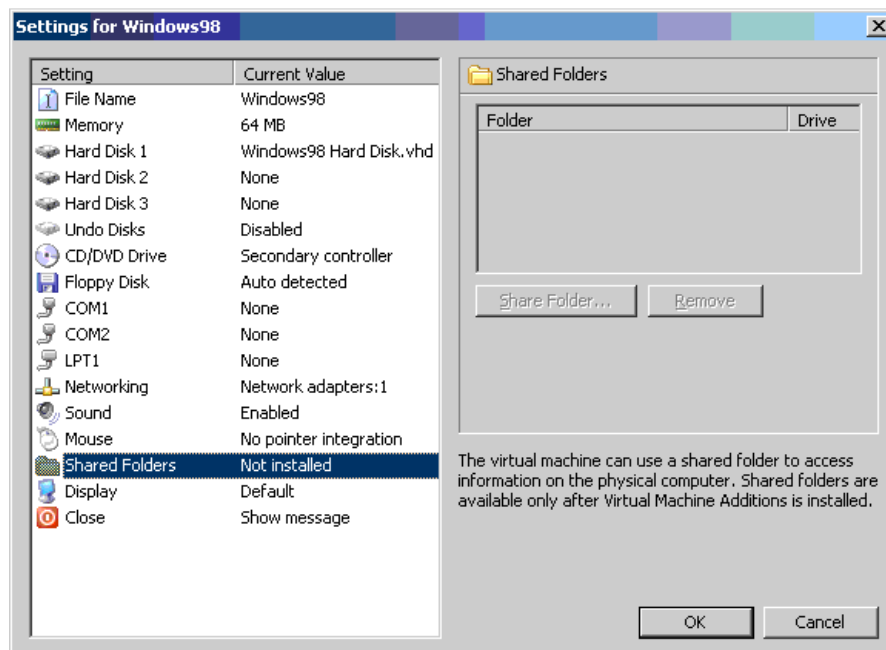


Figura 40 Configurando Ambiente Virtual – Passo 9  
Fonte: Microsoft (2007)

Em seguida é possível configurar as opções de display e opções de fechamento.

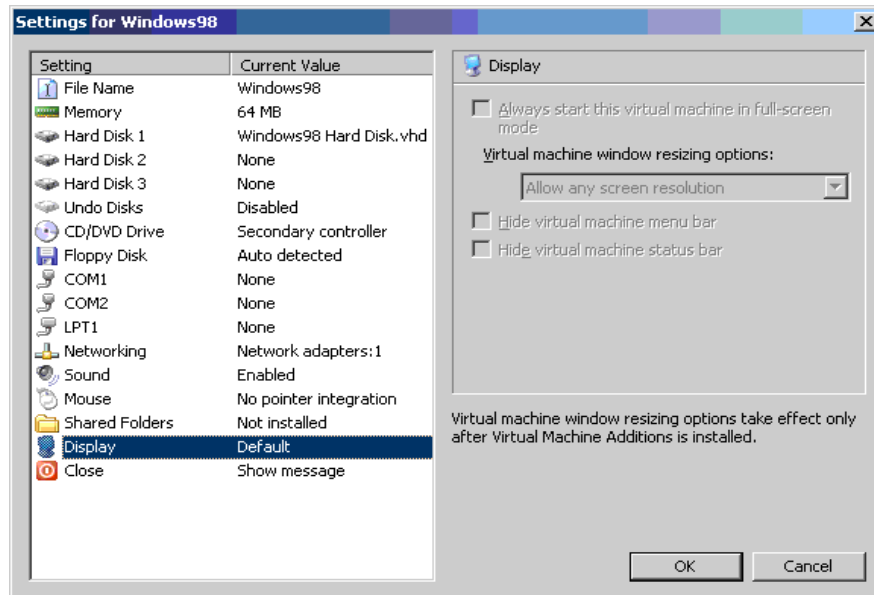


Figura 41 Configurando Ambiente Virtual – Passo 10  
 Fonte: Microsoft (2007)

A próxima etapa é a instalação do sistema operacional que será executado no ambiente virtual.

No console do Virtual PC selecionando o ambiente virtual criado, basta clicar em Start.

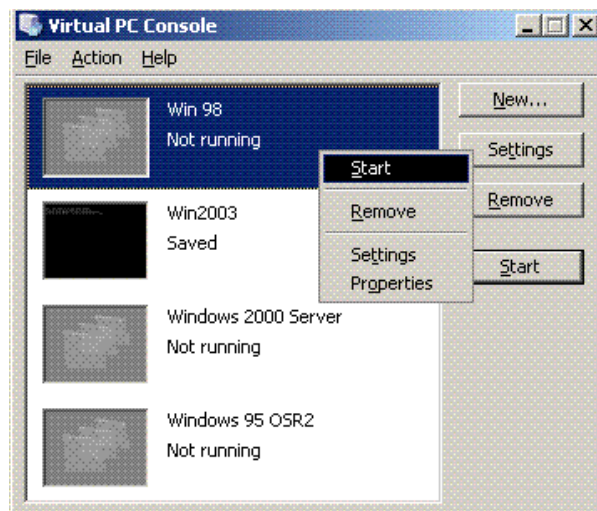


Figura 42 Inicializando o ambiente virtual – Parte 1  
 Fonte: Microsoft (2007)

Nesse momento aparecerá a tela de inicialização do ambiente virtual, porém ainda não foi instalado nenhum sistema operacional.

```
CD-ROM Device Driver for IDE (Four Channels)
Driver Version      : 3.2
Device Name        : MSCD001
Chipset Type       : 82371AB(PIIX4)
Drive 0: Port= 170 (Secondary Channel), Master IRQ= 15
Firmware Version   :
Transfer Mode      : Programmed I/O mode4
```

Figura 43 Inicializando o ambiente virtual – Parte 2  
Fonte: Microsoft (2007)

Todo o procedimento realizado a partir de agora está sendo executado no ambiente virtual, ou seja, a unidade “c:” referida demonstrada na Figura 44 é o HD do ambiente virtual.

```
Invalid media type reading drive C
Abort, Retry, Fail?Volume in drive C has no label

Invalid media type reading drive C
Abort, Retry, Fail?
Invalid media type reading drive C
Abort, Retry, Fail?Fail on INT 24

C:\>format c:

Invalid media type reading drive C
Abort, Retry, Fail?
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format (Y/N)?Y

Formatting 16,37.73M
Format complete.
Writing out file allocation table
Complete.
Calculating free space (this may take several minutes)...
Complete.

Volume label (11 characters, ENTER for none)?
```

Figura 44 Inicializando o ambiente virtual – Parte 3  
Fonte: Microsoft (2007)

Toda a instalação do sistema operacional deve ser completada normalmente.



Figura 45 Inicializando o ambiente virtual – Parte 4  
Fonte: Microsoft (2007)

## Utilizando um ambiente virtual

Após todo o processo de instalação se dar por completo o ambiente está pronto para ser utilizado.

A figura 46 demonstra dois ambientes sendo utilizados ao mesmo tempo.

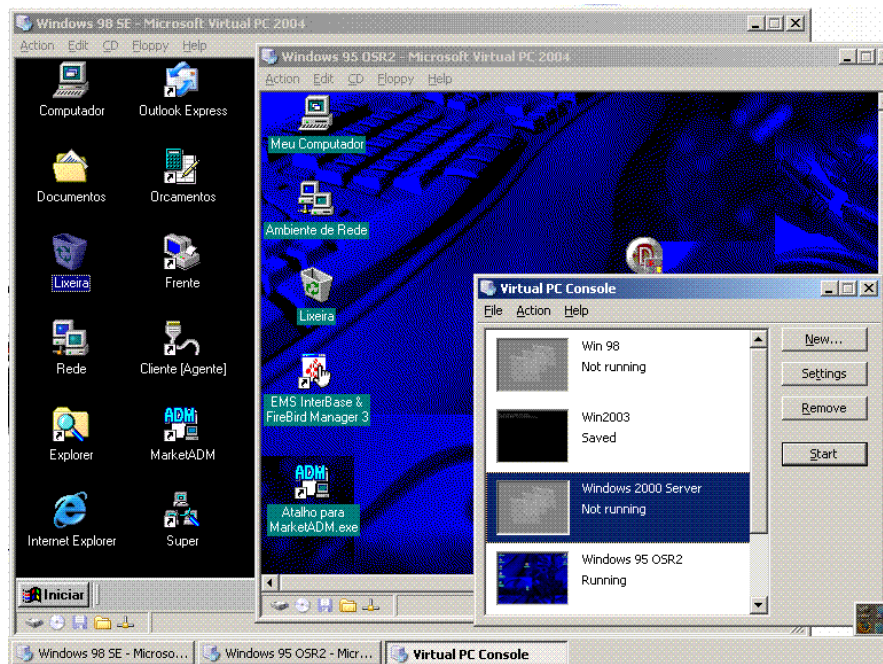


Figura 46 Utilizando o ambiente virtual  
Fonte: Microsoft (2007)

Está sendo utilizado Windows 98 e Windows 95, tudo isso com Windows XP de sistema operacional do computador.

Assim como em um computador real, se pode fazer quaisquer operação dentro do sistema operacional, desde copia de arquivos, configuração de rede, e até mesmo a própria formatação do HD.

Como já comentado, o HD do ambiente virtual é totalmente independente do HD da máquina física, sendo que os arquivos armazenados no ambiente virtual não serão vistos na máquina física, assim como os arquivos armazenados na máquina física não serão vistos no ambiente virtual.