

CENTRO UNIVERSITÁRIO UNIVESC
TRABALHO DE CONCLUSÃO DE CURSO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**KOMPAKT – DESCOMPACTADOR E COMPACTADOR DE ARQUIVOS EM
SHELL SCRIPT**

Tiago de Andrade

**Orientador: Prof. Márcio José Sembay,
MSc.**

LAGES

2012

TIAGO DE ANDRADE

**KOMPAKT – DESCOMPACTADOR E COMPACTADOR DE ARQUIVOS EM
SHELL SCRIPT**

Trabalho de conclusão de curso apresentado ao Centro Universitário UNIVESC de Lages, como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

**Orientador: Prof. Márcio José Sembay,
MSc.**

LAGES

2012

EQUIPE TÉCNICA

Acadêmico

Tiago de Andrade

Professor Orientador

Prof. Márcio José Sembay, MSc.

Coordenador de TCC

Prof. Márcio José Sembay, MSc.

Coordenador do Curso

Prof. Márcio José Sembay, MSc.

DEDICATÓRIA

**Dedico aos meus pais e a minha noiva, pela
motivação e apoio, e a Deus, pois sem ele
nada seria possível.**

SUMÁRIO

EQUIPE TÉCNICA.....	3
DEDICATÓRIA	4
SUMÁRIO.....	5
AGRADECIMENTOS.....	7
EPÍGRAFE	8
RESUMO	9
ABSTRACT	10
LISTA DE ABREVIATURAS.....	11
LISTA DE FIGURAS.....	12
I. INTRODUÇÃO	14
1.1 Justificativa	15
1.2 Objetivos	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos específicos.....	16
1.3 Metodologia	16
1.3.1 Estudo de Caso	16
1.3.2 Análise da Situação	17
1.3.3 Cronograma.....	18
1.3.4 Estrutura.....	19
II. FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 O Unix.....	20
2.2 GNU e a Free Software Foundation	21
2.3 O Linux.....	21
2.4 Softwares Proprietários.....	23
2.5 Softwares Livres	23
2.6 Análise de Processos	24
2.7 Levantamento de Dados	25
2.7.1 Observação Pessoal.....	25
III. PROJETO	26
3.1 Levantamentos de Requisitos	26
3.2 Requisitos Funcionais.....	26

3.3 Requisitos Não-Funcionais	26
3.4 Problemas a serem tratados	27
3.5 Melhorias	27
IV. FUNDAMENTAÇÃO TEÓRICA.....	28
4.1 Shell Script.....	28
4.2 Dialog.....	28
4.3 Ergonomia	29
4.4 Objetivos Básicos da Ergonomia	29
V. MODELAGEM DO SISTEMA	31
5.1 UML (Unified Modeling Language)	31
5.2 Diagrama de Atividade	31
5.3 Interfaces do Sistema	34
VI. CONCLUSÃO	52
VII. REFERÊNCIAS BIBLIOGRÁFICAS	53
VIII. ANEXOS	55
8.1 Código fonte Kompakt	55

AGRADECIMENTOS

Ao meu orientador, Prof. Ms. Marcio Sembay e a todos os professores que compartilharam de seu conhecimento e passaram o aprendizado necessário para minha formação acadêmica e profissional.

A todos os alunos e companheiros de classe, pelas experiências compartilhadas e convívio durante o curso.

EPÍGRAFE

**“Se fui capaz de ver mais longe, é porque
me apoiei em ombros de gigantes”.**

Isaac Newton

RESUMO

Este trabalho apresenta um sistema desenvolvido em Shell Script, para o sistema operacional Linux, um sistema que possibilita que novos usuários, inexperientes em Linux, realizem atividades rotineiras de gerenciamento e conversão de arquivos, diante de uma interface gráfica amigável e simples, sem maiores dificuldades. Com objetivo de melhorar a primeira impressão que o usuário final possui ao utilizar o Linux e ajudar no processo de adaptação ao novo sistema operacional, além de facilitar e agilizar as atividades diárias, sendo assim, foi realizado um estudo de caso no departamento de T.I da empresa Lojas Berlanda Ltda, que tornou possível a verificação das principais dificuldades e também as principais reclamações realizadas pelo usuário ao longo do expediente, utilizando o sistema operacional Linux.

Palavras Chaves: Linux, Shell Script, Sistema Operacional.

ABSTRACT

This paper presents a system developed in Shell Script for the Linux operating system, a system that enables new users inexperienced in Linux, perform routine activities of management and file conversion, before a friendly graphical user interface and simple, without difficulty. Aiming to improve the first impression that the end user has to use Linux and help the process of adaptation to the new operating system, and facilitate and streamline daily activities, so we conducted a case study in the TI department company Berlanda Stores LTDA, which made possible the verification of the main difficulties and also the main complaints suffered by the user throughout the record, using the Linux operating system.

Keywords: Linux, Shell Script, Operating System.

LISTA DE ABREVIATURAS

BZ2 – Bzip2

GNU – GNU is Not Unix

GPL - General Public License

GZ – GNU Zip

HP – Hewlett Packard

IBM – International Business Machines

MIT – Massachusetts Institute of Technology

PDV – Ponto de Venda

RAM – Random Access Memory

RAR – Roshal Archive

SO – Sistema Operacional

SPC – Serviço de Proteção ao Crédito

TAR – Tape Archive

TCC – Trabalho de Conclusão de Curso

TI – Tecnologia da Informação

UML – Unified Modeling Language

UNIX – Uniplexed Information and Computing System

LISTA DE FIGURAS

Figura 1 - Estrutura Lojas Berlanda Ltda.	17
Figura 2 - Cronograma de desenvolvimento do TCC.....	18
Figura 3 - Camadas do Sistema Operacional.....	22
Figura 4 - Fatores da Ergonomia.	30
Figura 5 - Diagrama de Atividade (Compactação).....	32
Figura 6 - Diagrama de Atividade (Descompactação).	32
Figura 7 - Diagrama de Atividade (Imagem).	33
Figura 8 - Diagrama de Atividade (Áudio).	33
Figura 9 - Diagrama de Atividade (Vídeo).....	34
Figura 10 - Menu Principal.....	34
Figura 11 - Menu Secundário (Compactação).....	35
Figura 12 - Menu Secundário (Descompactação).	35
Figura 13 - Interface de escolha (arquivo a ser compactado).....	36
Figura 14 - Interface de erro (arquivo a ser compactado).	36
Figura 15 - Interface de escolha do nome (arquivo a ser compactado).....	37
Figura 16 - Interface de erro da extensão (arquivo a ser compactado).....	37
Figura 17 - Salvar arquivo compactado.....	38
Figura 18 - Interface erro , local inexistente (arquivo a ser compactado).	38
Figura 19 - Duplicata (arquivo a ser compactado).	39
Figura 20 - Arquivo compactado.....	39
Figura 21 - Selecionar arquivo para descompactar	40
Figura 22 - Interface de erro, divergências com o arquivo (Descompactar).	40
Figura 23 - Salvar arquivo descompactado	41
Figura 24 - Interface erro, problemas com diretório (Descompactar).....	41
Figura 25 - Arquivo Descompactado.....	42
Figura 26 - Escolher imagem para conversão	42
Figura 27 - Interface de erro, divergências na imagem.	43
Figura 28 - Nome nova imagem	43
Figura 29 - Formato da imagem	44
Figura 30 - Tamanho da imagem.....	44
Figura 31 - Qualidade da Imagem	45
Figura 32 - Salvar Imagem	45
Figura 33 - Interface de erro, divergências com local para salvar a imagem.	46
Figura 34 - Imagem convertida	46
Figura 35 - Escolher vídeo para converter.....	47
Figura 36 - Nome para o novo vídeo.....	47
Figura 37 - Formato do vídeo.	48
Figura 38 - Salvar vídeo.	48
Figura 39 - Vídeo convertido.	49
Figura 40 - Escolher Áudio para ser convertido.....	49

Figura 41 - Nome arquivo de áudio.....	50
Figura 42 - Salvar arquivo de áudio.	50
Figura 43 - Arquivo de áudio convertido.	51

I. INTRODUÇÃO

Devido à popularização e boa fama do Linux em atividades empresariais, muitas empresas estão optando por utilizar estas plataformas em seus Datacenters e até mesmo em suas lojas e escritórios, com intuito de reter gastos e melhor aproveitar seus hardwares, porém tais alterações exigem que os operadores passem pelo devido treinamento de assimilação e adaptação do novo sistema operacional, pois um novo sistema vem acompanhado de novas ferramentas e novos meios de executar tarefas, e o novo, geralmente assusta aos usuários menos experientes.

Este projeto tem como principal intuito, facilitar as tarefas exercidas referentes a gerenciamento de arquivos, compactação e descompactação, manipulação de formatos, extensões e qualidade dos arquivos, o que seria um desafio para usuários iniciantes, o mesmo propõe que seja feito de forma simples e rápida.

Para que se torne eficiente e concreto, o software será formulado conforme as prioridades e deficiências de nossos usuários, observando e coletando todos os dados e informações necessárias para início do projeto, após, a implantação será realizada para verificar pontos fracos e corrigi-los, almejando torná-lo cada vez mais útil e indispensável.

Deste modo, planeja-se um software que trará benefícios e facilidades, em primeira instância para uso empresarial, mais podendo ser utilizado por qualquer usuário que sinta necessidade.

1.1 Justificativa

Atualmente são lançadas varias versões e atualizações com intuito de facilitar e popularizar o sistema operacional Linux, tornando-o mais simpático aos olhos dos usuários domésticos.

Com sua maior utilização voltada a empresas, o Linux e seus softwares são defendidos por inúmeros motivos, conforme cita Hexsel (2002):

Dentre as vantagens decorrentes da utilização de software livre salientam-se:

- Custo social é baixo;
- não fica refém de tecnologia proprietária;
- independência de fornecedor único;
- desembolso inicial próximo de zero;
- não obsolescência do hardware;
- robustez e segurança;
- possibilidade de adequar aplicativos e redistribuir versão alterada;
- suporte abundante e gratuito;
- sistemas e aplicativos geralmente muito configuráveis.

Porém, o grande ponto fraco do Linux é o uso doméstico, exige certo conhecimento na instalação e configuração de aplicativos, gerando desconforto ao usuário que não possui experiência e está experimentando um primeiro contato, tais características dificultam o uso do Linux e tendem a afastar usuários iniciantes.

Por sua vez, o usuário geralmente não se adapta ao Linux, por achá-lo um sistema de difícil manuseio, com diferentes formatos e extensões de arquivos, e a utilização do Shell constantemente, com linhas de comando para executar inúmeras atividades, assim não leva adiante o processo de assimilação e aprendizado deste sistema, e opta por outro sistema operacional, no qual já possui certo conhecimento e dispõe de todas as ferramentas e facilidades para o uso doméstico.

1.2 Objetivos

1.2.1 Objetivo Geral

Devido à dificuldade que os novos usuários em Linux apresentaram em relação ao processo de adaptação ao sistema operacional, surgiu à necessidade de prestar auxílio a estes usuários, sendo assim foi desenvolvido um software que realiza tarefas de compactação e

descompactação de arquivos, além da conversão de vídeos, imagens e músicas, as quais são as atividades em que surge maior dificuldade.

1.2.2 Objetivos específicos

Os objetivos específicos se resumem as seguintes características:

- a) Auxiliar o usuário iniciante em Linux a executar as tarefas dispostas ao software com menor dificuldade.
- b) Diminuir o uso do Terminal, para executar tarefas rotineiras.
- c) Disponibilizar uma interface gráfica limpa, objetiva e de fácil uso.

Assim, o usuário iniciante em Linux, não irá sentir com tanta intensidade a mudança de sistema operacional, ao poder utilizar um software de fácil uso e adaptação, que pode lhe auxiliar em atividades simples, mais que podem se tornar complexas para quem não tem intimidade com o sistema.

1.3 Metodologia

1.3.1 Estudo de Caso

A essência de um estudo de caso, a principal tendência em todos os tipos de estudo de caso, é que ele tenta esclarecer uma decisão ou um conjunto de decisões, o motivo pelo qual foram tomadas, como foram implementadas e com quais resultados. (SCHRAMM, 1971, apud YIN, 2001, p. 31).

Iniciando este estudo, temos como referência o setor de Tecnologia da Informação da empresa DB S/A Comércio de Móveis e Eletrodomésticos, com nome fantasia Lojas Berlanda Ltda, possui mais de 190 filiais distribuídas pelo estado, é uma das maiores redes de varejo de móveis, eletrodomésticos e eletroeletrônicos de Santa Catarina. Onde o ambiente de T.I será utilizado para consolidação do estudo.

O setor de T.I foi escolhido para dar continuidade ao estudo, pois no decorrer do dia, seus profissionais atendem diversos problemas e chamados relacionados ao Linux, o que se torna uma oportunidade para manter o estudo rico em informações e atualizado.

1.3.2 Análise da Situação

Cada filial da empresa Lojas Berlanda Ltda possui uma estrutura padrão, sendo alterada de acordo com as necessidades e características da filial, formada por dois núcleos, o setor de crediário e a área de vendas, que segue com maiores detalhes na imagem abaixo:

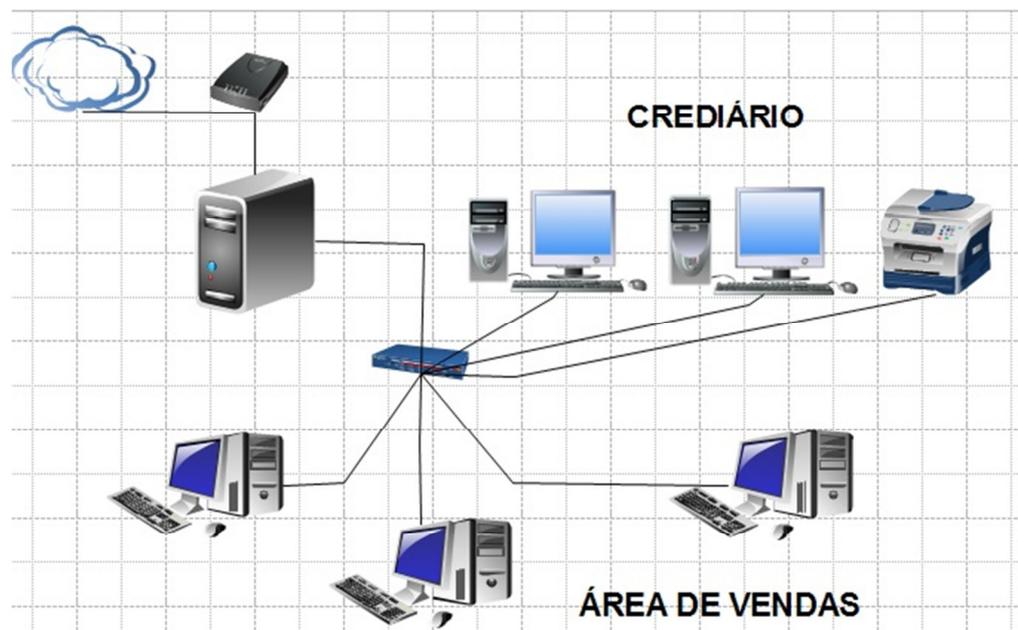


Figura 1 - Estrutura Lojas Berlanda Ltda.
Fonte: Próprio Autor (2012)

No crediário ficam instalados o Servidor, juntamente com um ou mais desktops que tem por finalidade a cobrança de prestações, impressões de carnês, finalização das vendas, criação de cadastros, análise ao SPC, venda por cartões de crédito, recargas de celulares, entre outras, juntamente com uma impressora matricial, uma impressora laser e uma ou mais impressoras de cupom fiscal, geralmente os desktops utilizam o sistema operacional Windows, visando facilitar as atividades da filial, sendo que o Servidor e PDV's tem o Linux instalado, e muitas pessoas possuem maior conhecimento no Windows. O segundo núcleo é

composto por PDV's espalhados por toda a filial, onde os vendedores conseguem atender os clientes com maior eficiência, simulando as vendas, calculando os juros e gerando os pedidos.

Atualmente um novo projeto de redução de custos foi aprovado, com objetivo de retirar todos os desktops que possuem o Windows instalado das filiais, o que gerou um enorme desconforto e muitas dúvidas as pessoas que estavam adaptadas ao sistema operacional Windows, originando muitas reclamações e um aumento significativo de atendimentos ao departamento de T.I, sobre problemas simples que os próprios funcionários resolviam antes das alterações do sistema operacional.

O intuito deste script é ser implantado inicialmente em uma máquina voltada ao suporte, que trabalhe com o sistema operacional Linux, visando auxílio e economia de tempo, nas atividades exercidas em que o software oferece suporte, sendo sempre monitorado e avaliado, sujeito a alterações, correções e melhorias. Dependendo dos resultados apresentados, a intenção é aplicar este software nas filiais da rede Berlanda, com objetivo de auxiliar os funcionários que são leigos em Linux, diminuir os chamados relacionados ao setor de T.I e conseqüentemente gerar maior facilidade aos funcionários e tornar a adaptação ao novo sistema operacional mais agradável.

1.3.3 Cronograma

Cronograma das atividades realizadas para o desenvolvimento deste projeto:

Atividades	ago/12	set/12	out/12	nov/12	dez/12
Pesquisa Bibliográfica	■				
Análise Bibliográfica					
Redação do Texto		■	■		
Modelagem do Software		■	■		
Desenvolvimento do Software			■	■	
Implantação do Software				■	
Testes Finais				■	
Correções finais				■	
Entrega e Defesa do TCC					■

Figura 2 - Cronograma de desenvolvimento do TCC.

Fonte: Próprio Autor (2012).

1.3.4 Estrutura

Para que fosse possível o desenvolvimento e conclusão deste projeto, foi realizada uma pesquisa e análise bibliográfica, onde foram averiguados diversos autores e várias fontes, como livros, artigos, e internet, visando à utilização de um conteúdo rico e atualizado para embasar este TCC. Logo após, o trabalho foi voltado à modelagem do sistema, onde segundo Castilho (2008):

Muitas empresas, atualmente, desenvolvem e levantam todos os requisitos funcionais e não funcionais de um determinado sistema, porém se esquecem de elaborar um modelo lógico do sistema, passando direto para a fase de codificação. Justificam que o prazo de entrega do produto se estenderá muito e que o custo do projeto aumentará. O que não percebem é que a má compreensão dos requisitos pode resultar na construção de um sistema com deficiências, sem amplo conhecimento do domínio do software gerando problemas futuros de desenvolvimento e de projeto, tais como aumento de custos, escopo fora do que foi planejado e descumprimento de prazos. O produto final pode não atender as necessidades do cliente, gerando retrabalho e insatisfação.

Finalizando com o desenvolvimento do software em Shell Script, a sua implantação e a fase de testes e correções.

Hardware utilizado para desenvolvimento:

- Notebook Acer Aspire 4730
- Processador Intel® Pentium® Dual Core T3400 2,16GHz
- Memória RAM de 4GB
- Disco Rígido de 250GB
- S.O Debian 6.0.5 squeeze i386 Kernel 3.0.4

Hardware utilizado para implantação e testes:

- Desktop Dell Vostro 260s
- Processador Intel® Core™ i3-2120 3.3GHz
- Memória RAM de 4GB
- Disco Rígido de 500GB
- S.O Debian 6.0.5 squeeze amd64 Kernel 3.5.3

II. FUNDAMENTAÇÃO TEÓRICA

2.1 O Unix

Considerado por muitos como o pai e precursor de todos os sistemas operacionais que conhecemos hoje em dia, e segundo Ferreira (2003):

Entre os vários sistemas operacionais existentes, o Unix é um exemplo de maturidade e eficiência. Durante mais de trinta anos tem sido utilizado universalmente nas mais diversas aplicações e plataformas de hardware, e continua ainda a ser um sistema operacional poderoso, moderno e atual, em meio a um cenário de rápidas mudanças tecnológicas, tanto em hardware como em software.

O Unix é um sistema operacional multitarefa e multiusuário, o que significa que vários usuários podem utilizar o mesmo computador ao mesmo tempo, por meio de acesso remoto e também que várias tarefas ou programas possam ser executados em paralelo. Foi desenvolvido em uma época em que os computadores eram apenas sonhos para pessoas comuns, pois eram grandes e caros.

Para Flynn (2002), o sistema operacional é uma espécie de gerente executivo, ou seja, aquela parte de um sistema de computação que administra todos os componentes de hardware e de software.

Após uma cronologia de vários anos de desenvolvimento, desde meados dos anos 70 até os dias de hoje, o Unix se mostrou um sistema operacional com grande sucesso no meio comercial, por ser desenvolvido por diversos profissionais ao redor do mundo, possui uma grande gama de opções e ferramentas totalmente configuráveis quando se trata de adaptar o sistema para uso próprio, e também se explica a complexidade do sistema, pois foi desenvolvido por pessoas com grande conhecimento no assunto, não sendo pensado para solucionar os problemas de pessoas comuns, mais sim dos próprios usuários que o desenvolveram.

Outro fato que com certeza influenciou e contribuiu para o sucesso e a progressão do Unix, foi o fato de em seus primeiros anos de vida, ser distribuído com licença gratuita, tendo sido distribuído para diversos órgãos e universidades dos Estados Unidos para pesquisa e aprimoramento, somente algum tempo depois sua licença se tornou proprietária.

2.2 GNU e a Free Software Foundation

Em 1983, um pesquisador do Laboratório de Inteligência Artificial do MIT, (Massachusetts Institute of Technology), chamado Richard Stallman, tomou uma decisão pessoal que iria marcar profundamente a história da tecnologia da informação. Stallman deu início ao projeto GNU. Este nome incomum é de um conhecido animal africano e também o acrônimo recursivo de GNU IS NOT UNIX, ou seja, o projeto GNU teria como objetivo produzir um sistema operacional livre que pudesse fazer o mesmo que o sistema Unix. Sua proposta era construir um sistema capaz de rodar programas e aplicativos do Unix, mas que fosse livre e independente de licenças proprietárias de uso.

A ideia de constituir um sistema operacional livre foi ganhando adeptos e se consolidou na formação da Free Software Foundation, em 1984, dirigida por ele. Aproveitando o crescimento da comunicação em rede, hackers, geeks e apaixonados pela programação trocavam mensagens contendo pedaços de programas e linhas de código. Vários componentes do sistema operacional foram desenvolvidos, enquanto ganhava corpo a proposta do compartilhamento.

2.3 O Linux

Para Ferreira (2003), o Linux é um clone de Unix criado como uma alternativa barata e funcional para quem não está disposto a pagar o alto preço de um sistema Unix comercial ou não tem um computador suficientemente rápido.

Nos anos 80 alguns pesquisadores estavam interessados em desenvolver um clone melhorado e livre do sistema operacional Unix, o desafio era enorme, pois teriam de desenvolver um novo Kernel e Shell para o sistema operacional, tarefa onde muitos pesquisadores falharam. (FERREIRA, E, R. 2003).

O Kernel é o núcleo do sistema operacional, ou seja, a parte mais próxima do hardware e o Shell é a interface entre o usuário e o Kernel, um interpretador de comandos, como a imagem abaixo nos mostra:

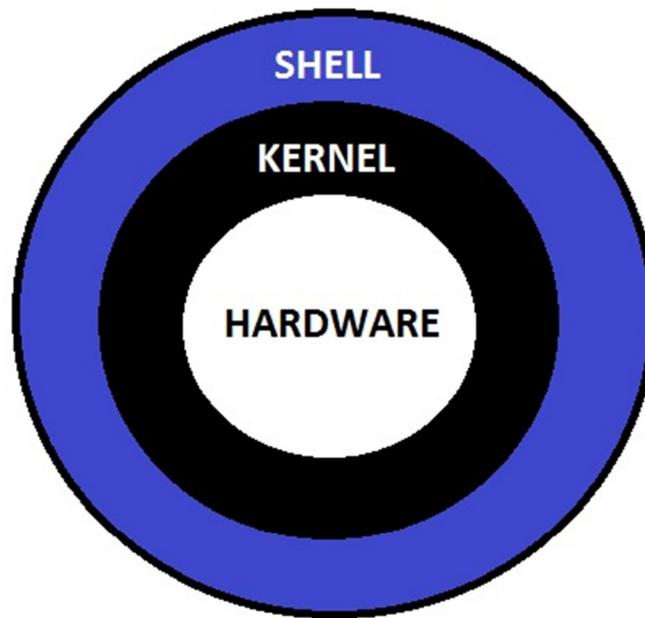


Figura 3 - Camadas do Sistema Operacional.
Fonte: Ferreira, 2003, p.33.

Um aluno da Finlândia, no final da década de 80, estava disposto a construir um Kernel clone do Unix, Linus Benedict Torvalds iniciou seu projeto particular baseado no seu interesse e vontade de desenvolver uma versão melhor do Unix, porém como descreve Silveira (2004):

Alertava que seria apenas um hobby, nada sério como o GNU. Depois de meses de trabalho desenvolveu um Kernel capaz de executar os utilitários de programação e os comandos padrão do Unix, o software se chamaria Linux, a junção de seu nome, Linus, com o sistema operacional Unix.

Segundo Silveira (2004), vendo que não conseguiria dar continuidade ao seu projeto sozinho, Linus Torvalds disponibilizou suas fontes para consulta, aonde vários programadores viriam a ter acesso ao Linux, uma vasta rede de colaboradores foi se formando para aperfeiçoar as novas versões do software.

Surgia assim um grande concorrente às tecnologias proprietárias, o sistema operacional Linux.

2.4 Softwares Proprietários

Richard Stallman, presidente da Free Software Foundation (Fundação do Software Livre), costuma comparar o software a uma receita de bolo. Ambos são um conjunto de instruções. Um software diz ao computador o que este deve fazer. Uma receita diz à pessoa as quantidades de cada ingrediente, a ordem em que devem ser misturados e outras orientações. E se as pessoas fossem impedidas de trocar receitas? Ou se fossem proibidas de melhorar a receita que conseguiram de sua mãe ou de seu vizinho?

Para Silveira (2004), quando falamos em software proprietário estamos falando de um modelo de desenvolvimento e distribuição baseado em licenças restritivas de uso. Estamos falando em autoria e propriedade do software.

O modelo de software proprietário esconde os algoritmos que o compõem, conforme cita Silveira (2004):

Apesar de ser composto por informações agrupadas e de se basear em conhecimentos acumulados pela humanidade, a indústria de software proprietário se direcionou para tentar bloquear e evitar que o caminho de seu desenvolvimento fosse semelhante ao desenvolvimento do conhecimento científico. A ciência cresce a partir do princípio de compartilhamento, e não a partir da ideia de propriedade. Por ser essencialmente social, não se aplica ao conhecimento a ideia de apropriação privada.

Quando alguém compra uma casa, tem o direito de reformá-la inteiramente, de ampliá-la ou de demolir suas paredes. Pode até revendê-la. Um software tipicamente proprietário não dá ao seu usuário nenhuma destas opções. Ele continua a ser propriedade da empresa que o vendeu. (SILVEIRA, A, S. 2004).

2.5 Softwares Livres

A essência do software livre está na liberdade de usar e desenvolver, onde segundo Hexsel (2012):

O movimento de publicação de Software Livre ganhou notoriedade nos últimos anos. Este modo de produção de software tem resultado em produtos de excelente qualidade e grande penetração em certos nichos do mercado mundial de software. A característica mais importante do software livre é a liberdade de uso, cópia, modificações e redistribuição. Esta liberdade é conferida pelos autores do programa e é efetivada através da distribuição do código-fonte dos programas, o que os

transforma em bens públicos, disponíveis para utilização por toda a comunidade e da maneira que seja mais conveniente a cada indivíduo.

A liberdade para usar, copiar, modificar e redistribuir software livre lhe confere uma série enorme de vantagens sobre o software proprietário, como afirma Hexsel (2012):

A mais importante delas é a disponibilidade do código-fonte, porque isto evita que os usuários se tornem reféns de tecnologias proprietárias. Além desta, as vantagens técnicas são também consideráveis. A comunidade de desenvolvimento de software livre está espalhada pelo mundo todo e seus participantes cooperam nos projetos através da Internet. Estima-se que participem desta comunidade mais de cem mil programadores e projetistas, com a grande maioria deles trabalhando voluntariamente em um ou mais projetos. Estima-se também que existem mais de dez milhões de usuários regulares de sistemas operacionais e aplicativos distribuídos como software livre.

Assim, a licença do software livre é uma licença não proprietária de uso. O software livre possui um autor ou vários autores, mas não possui donos. Dessa forma, o usuário do software livre também tem o direito de ser desenvolvedor, caso queira. Quem o adquire pode usá-lo para todo e qualquer fim, inclusive tem a permissão de alterá-lo completamente. (SILVEIRA, A, S. 2004).

2.6 Análise de Processos

As empresas em geral oferecem algum tipo de produto ou serviços para seus clientes. E todo e qualquer trabalho que for desenvolvido, para a realização e geração de produtos ou serviços, e que dessa forma possui algum grau de importância para a empresa, está enquadrado em algum processo. (GONÇALVES, 2000).

Não visando atender o cliente final, mais sim os funcionários que os atendem, a análise de processos segundo Nassif (2004):

Proporciona ao analista de sistemas o conhecimento da organização e o cotidiano das operações gerenciais. Possibilitando a avaliação da necessidade das funções e atividades que farão parte do sistema em questão, adequando as funcionalidades, funções, operações, tecnologias, ferramentas e recursos às pessoas que executam as tarefas e atividades que compõem o processo e o faz acontecer.

Com uma análise detalhada dos processos e levantamento de dados, é possível identificar com maior eficiência pontos a serem melhorados, onde o trabalho se torna mais ágil e claro.

2.7 Levantamento de Dados

Uma das fases mais importantes de uma pesquisa é o levantamento de dados, que consiste na obtenção de informações sobre problemas ou necessidades que podem ser melhoradas.

Para Petrarca (2004) após a elaboração do planejamento referente ao levantamento a ser realizado, iniciará “o trabalho propriamente dito”, através da verificação dos “diversos componentes sistêmicos (internos e externos) que interagem com e na empresa”.

Com o levantamento de dados o analista produzirá ferramentas que possibilitarão sua análise e a ordenação das ideias, para que posteriormente seja elaborado o projeto lógico. Com o detalhamento dos processos, também será possível estabelecer as necessidades dos usuários, e visualizar os pontos mais críticos.

2.7.1 Observação Pessoal

O planejamento da técnica de observação pessoal visa, como todo planejamento, definir os objetivos que delimitarão as ações do analista responsável por sua execução. No planejamento deve conter o que precisa ser observado, bem como a forma pela qual deve ser observado. Como também, quando e em qual ou quais locais devem ser feita a observação.

Conforme Magarão (2006) explica:

Através da observação pessoal o analista passa a ter conhecimento sobre o comportamento do processo em estudo. E isso possibilita identificar além de eventos não programados, novas formas de concepção dos processos. Dessa forma, a utilização dessa técnica tem por finalidade, além das outras já citadas, a percepção de novas oportunidades de melhoria.

Tudo que acontece no ambiente em que se desenvolve o processo em estudo é verificado através da técnica Observação Pessoal. Dessa forma o seu uso pode ser considerado muito mais eficiente.

III. PROJETO

Na fase de projeto, determina-se como o sistema funcionará para atender aos requisitos, de acordo com os recursos tecnológicos existentes, alguns dos exemplos de aspectos a serem considerados na fase do projeto são: modelagem do sistema, padrão de interface gráfica, linguagem de programação, etc. (Bezerra, E. 2005).

3.1 Levantamentos de Requisitos

Na etapa de levantamento de requisitos, o time de desenvolvimento se prende em entender o negócio que o sistema vai automatizar, esse levantamento compreende explorar as necessidades dos usuários. Para Bezerra (2005):

A atividade de levantamento de requisitos corresponde a etapa de compreensão do problema aplicada ao desenvolvimento do software. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido. Nessa etapa, os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido. Essas necessidades são geralmente denominadas requisitos.

3.2 Requisitos Funcionais

Um requisito funcional define uma função de um sistema de software ou seu componente.

- Descompactar arquivos tar, tar.gz, tar.xz, bz2, tar.bz2, rar, zip;
- Compactar arquivos tar, tar.gz, tar.xz, bz2, tar.bz2, rar, zip;
- Converter vídeos, áudios e imagens;

3.3 Requisitos Não-Funcionais

Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, segurança, e tecnologias envolvidas. Em geral, requisitos não-

funcionais são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos ou não.

- Sistema operacional Debian Squeeze 6.0.x I386;
- Dialog;
- Ffmpeg, Lame, ImageMagic;
- Unrar, Unzip, Rar, Zip, Bzip2;

3.4 Problemas a serem tratados

Os principais problemas que o sistema visa corrigir ou amenizar.

- Excesso de chamados;
- Má adaptação ao novo S.O;
- Tempo mal aproveitado;
- Insegurança do usuário;

3.5 Melhorias

Após a fase de testes e implantação, o sistema almeja a melhoria nos campos citados abaixo:

- Maior Agilidade e aproveitamento do tempo;
- Diminuição de chamados abertos;
- Melhoria das condições de trabalho;

IV. FUNDAMENTAÇÃO TEÓRICA

4.1 Shell Script

O Shell Script é considerado por muitos uma linguagem de fácil aprendizado, utilizado para facilitar a realização de inúmeras tarefas administrativas no Linux, ou até mesmo, desenvolver seus próprios programas, conforme cita Jargas (2008):

Um Shell Script é um programa feito para funcionar no interpretador de comandos (prompt) padrão do UNIX/Linux, como o Bourne Shell, ou o Bash. O script é uma lista de comandos executados em sequência (um comando após o outro). Um roteiro predefinido de comandos e parâmetros. Geralmente o que está no script é aquilo que o usuário digitaria na linha de comando. Realmente, apenas colocando-se todo o histórico da linha de comando em um arquivo, tem-se um script.

Uma curiosidade do script é que as pessoas parecem não levá-lo muito a sério. Fazem códigos feios, sem comentários, sem alinhamento, sem muita clareza, mais como o shell é poderoso e gostoso de usar, com o tempo mais e mais tarefas começam a ser desempenhadas por scripts e estes começam a crescer e se difundir. (Jargas, M, A. 2008).

4.2 Dialog

Segundo Jargas (2008), o Dialog é um programa para console (modo texto) que desenha caixas de diálogo na tela, similares as do modo gráfico, com botões, entradas para texto e menu. Essas caixas são utilizadas para compor interfaces amigáveis com o usuário, para que ele responda perguntas ou escolha opções.

O Dialog é um executável e recebe todos os parâmetros via linha de comando, então ele geralmente é usado dentro de um Shell Script. Serve para fazer programas interativos, que o usuário precisa operar durante sua execução. Tarefas comuns feitas com o Dialog são escolher uma opção em um menu, escolher um arquivo, uma data, e digitar frases ou senhas. (Jargas, M, A. 2008).

Para Jargas (2008):

Com o Dialog é possível fazer programas em shell que se "parecem" com programas gráficos, onde o usuário vê apenas telas e navega entre elas apertando os botões de "OK" e "CANCELAR". Um exemplo clássico desse tipo de interface são os

programas de instalação de software. Utilizando este conceito de telas, é possível "amarrar" o usuário ao programa, lhe apresentando as opções disponíveis, sem que ele precise ter acesso direto à linha de comando. Útil para logins restritos e para ajudar iniciantes.

4.3 Ergonomia

A ergonomia é o estudo da adaptação do trabalho ao homem. A ergonomia tem uma visão ampla, abrangendo atividades de planejamento e projeto, que ocorrem antes do trabalho ser realizado, e aqueles de controle e avaliação, que ocorrem durante e após este trabalho, onde tudo isso é necessário para que o trabalho possa atingir os resultados desejados. (Lida, I. 2005).

A associação brasileira de ergonomia adota a definição em que, entende-se por ergonomia o estudo das interações das pessoas com a tecnologia, a organização e o ambiente, objetivando intervenções e projetos que visem melhorar, a segurança, o conforto, o bem-estar e a eficácia das atividades humanas.

Em uma visão geral, a ergonomia e os profissionais que trabalham nesta área têm como principal objetivo tornar os ambientes e sistemas, compatíveis com as necessidades e limites das pessoas, visando um melhor conforto e aprendizado.

4.4 Objetivos Básicos da Ergonomia

O objetivo da ergonomia é adaptar o trabalho às capacidades e possibilidades dos seres humanos. Para Lida (2005):

A ergonomia estuda os diversos fatores que influem no desempenho do sistema produtivo e procura reduzir as suas consequências nocivas sobre o trabalhador. Assim, ela procura reduzir a fadiga, estresse, erros e acidentes, proporcionando segurança, satisfação e saúde aos trabalhadores, durante o seu relacionamento com esse sistema. A ergonomia visa em primeiro lugar, a saúde, segurança e satisfação do trabalhador, a eficiência virá como consequência.

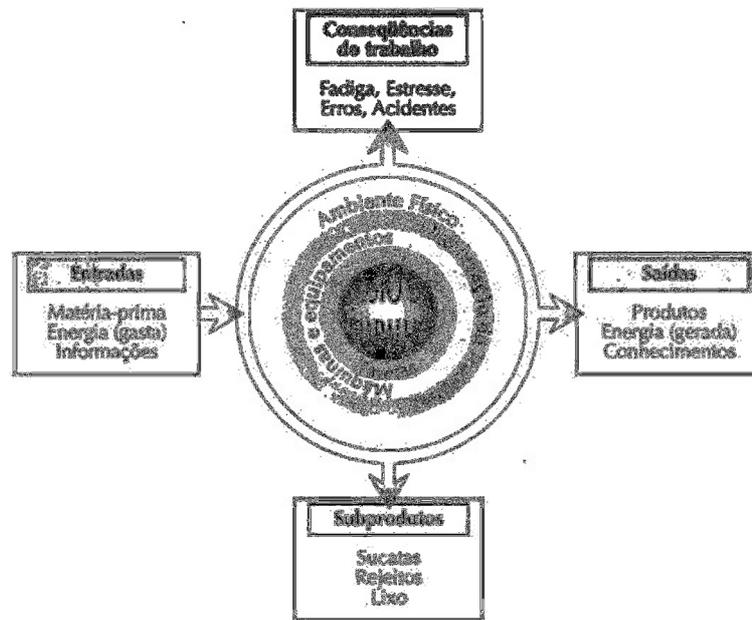


Figura 4 - Fatores da Ergonomia.
Fonte: Lida, 2005, p.4.

- Saúde: A saúde do trabalhador é mantida quando as exigências do trabalho e do ambiente não ultrapassam as suas limitações energéticas e cognitivas, evitando as situações de estresse, riscos de acidentes e doenças ocupacionais.
- Segurança: A segurança é conseguida com os projetos do posto de trabalho, ambiente e organização do trabalho, que estejam dentro das capacidades e limitações do trabalhador, de modo a reduzir os erros, estresse e fadiga.
- Satisfação: Satisfação é o resultado do atendimento das necessidades e expectativas do usuário. Os trabalhadores satisfeitos tendem a adotar comportamentos mais seguros e são mais produtivos que aqueles insatisfeitos.
- Eficiência: Eficiência é a consequência de um bom planejamento e organização do trabalho, que proporcione saúde, segurança e satisfação ao trabalhador. Ela deve ser colocada dentro de certos limites, pois o aumento indiscriminado da eficiência pode implicar em prejuízos a saúde e segurança.

V. MODELAGEM DO SISTEMA

5.1 UML (Unified Modeling Language)

A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos. Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. (Bezerra, E. 2005).

Continuando com Bezerra (2005):

Na construção de sistemas de software, assim como na construção de sistemas habitacionais, também há uma gradação de complexidade. Para a construção de sistemas de software mais complexos, também é necessário um planejamento inicial anterior. O equivalente ao projeto das plantas da engenharia civil. Essa necessidade leva o conceito de modelo, tão importante no desenvolvimento de sistemas, de uma perspectiva mais ampla, um modelo pode ser visto como uma preparação idealizada de um sistema a ser construído.

5.2 Diagrama de Atividade

Um diagrama de atividade mostra um processo de negócios ou um processo de software como um fluxo de trabalho por meio de uma série de ações. Pessoas, computadores ou componentes de software podem executar essas ações.

Você pode usar um diagrama de atividade para descrever os processos de vários tipos, como, por exemplo, um processo de negócios ou fluxo de trabalho entre usuários e o seu sistema e as etapas executadas em um caso de uso, entre outros.

A figura 05 apresenta o modelo desenvolvido com base no diagrama de atividade, referente ao módulo responsável pela parte de Compactação do sistema, desenvolvido com o software Edraw UML Diagram.

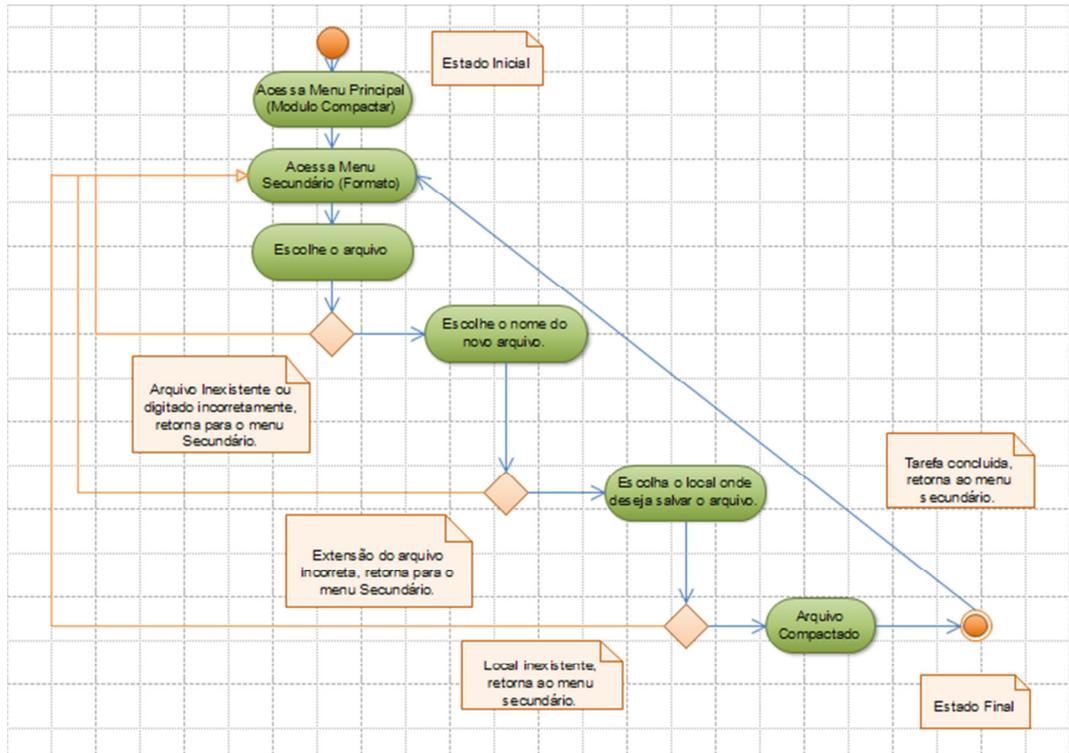


Figura 5 - Diagrama de Atividade (Compactação).
Fonte: Próprio Autor (2012)

A figura 06 representa o diagrama de atividade relacionado ao módulo de descompactação do sistema.

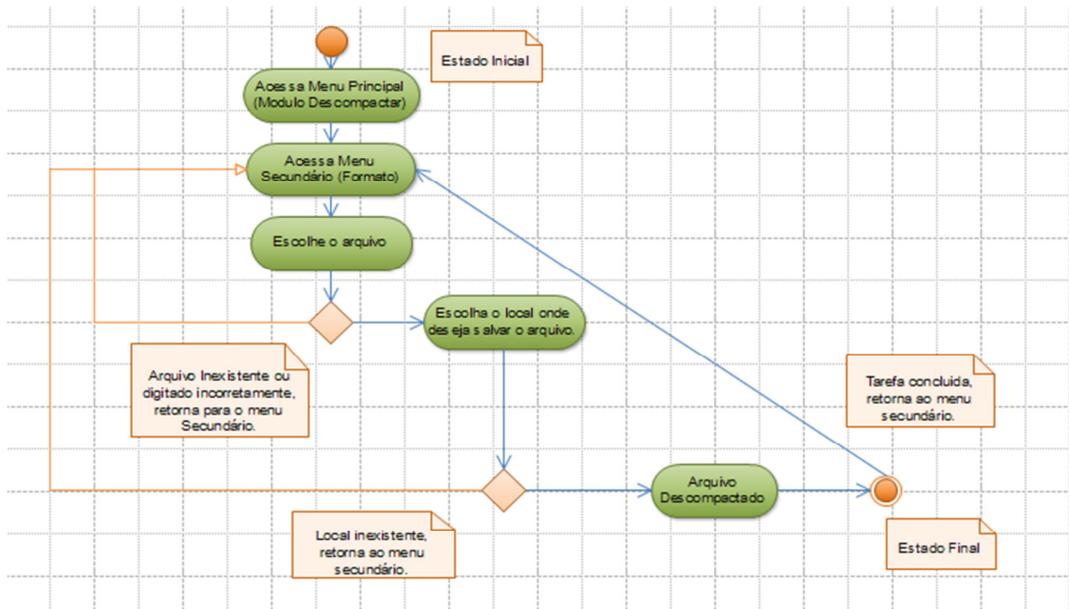


Figura 6 - Diagrama de Atividade (Descompactação).
Fonte: Próprio Autor (2012)

A figura 07 representa o diagrama de atividade relacionado ao módulo de conversão de imagens.

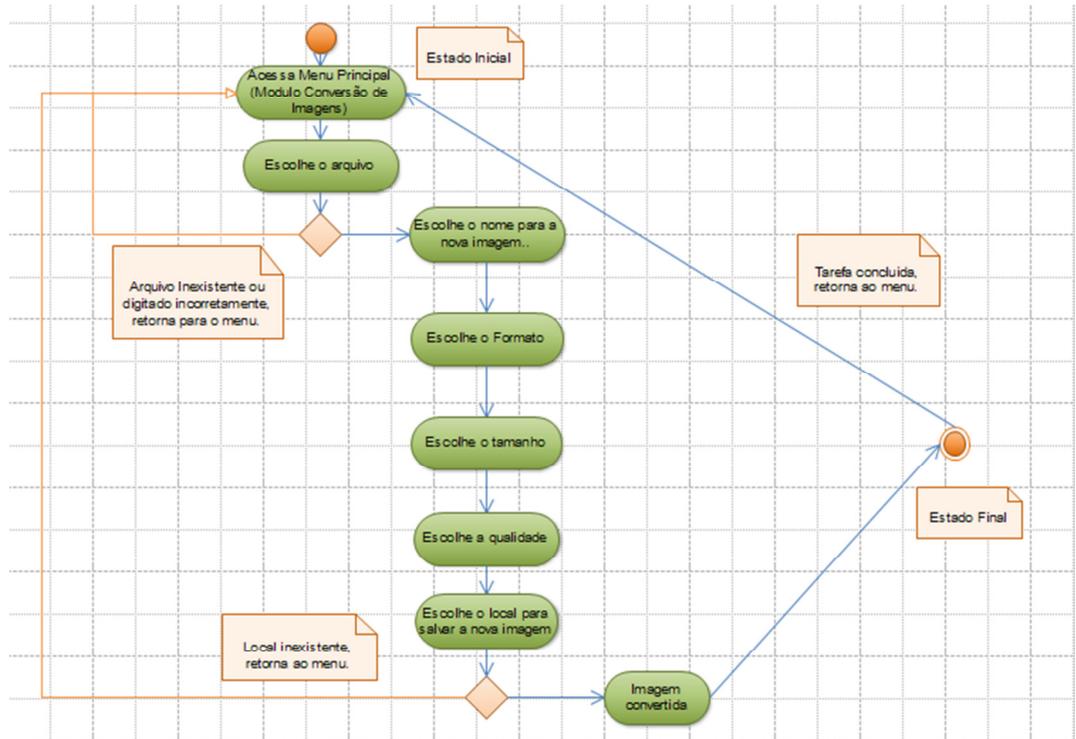


Figura 7 - Diagrama de Atividade (Imagem).
Fonte: Próprio Autor (2012)

A figura 08 representa o diagrama de atividade referente ao módulo de conversão de áudio.

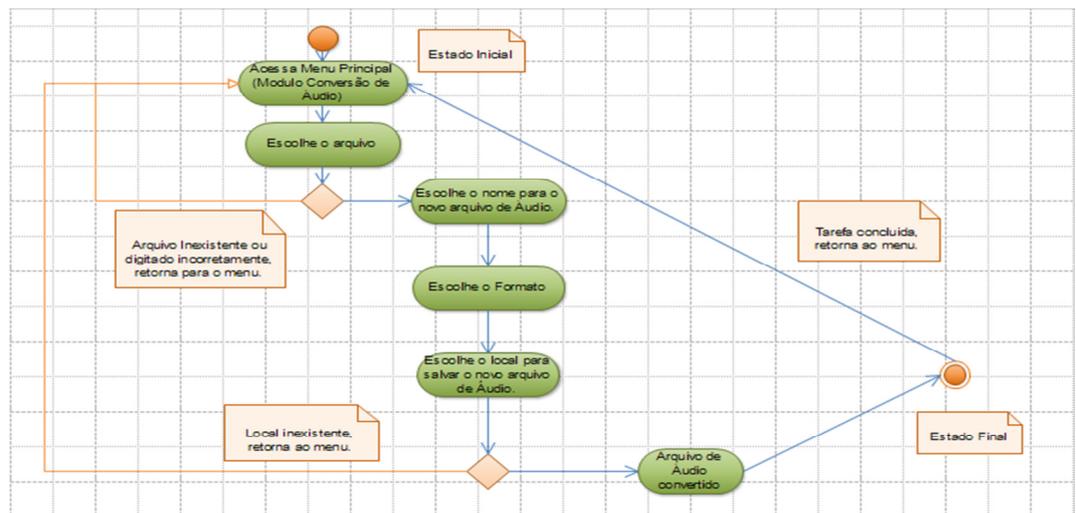


Figura 8 - Diagrama de Atividade (Áudio).
Fonte: Próprio Autor (2012)

A figura 09 representa o diagrama de atividade relacionado ao módulo de conversão de vídeos.

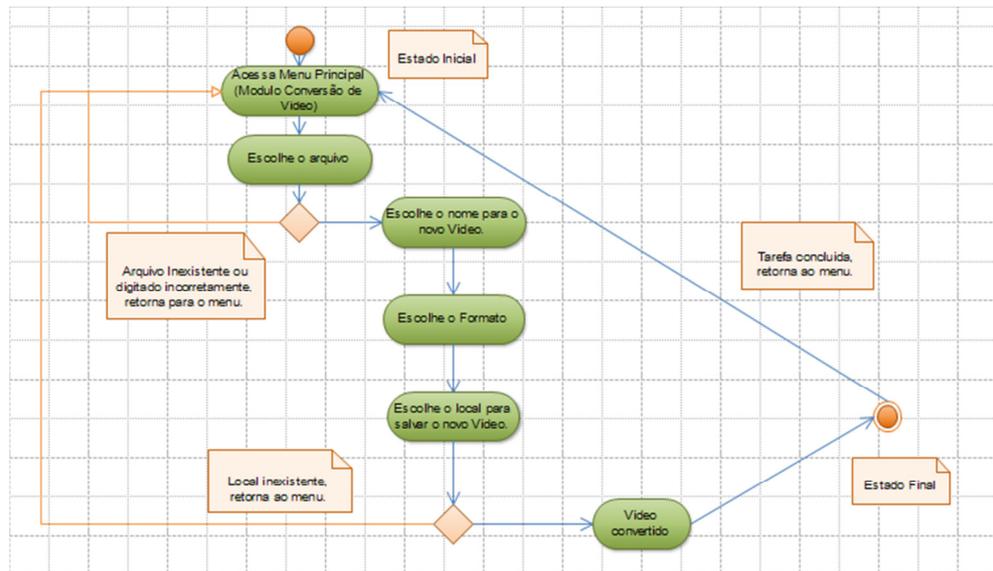


Figura 9 - Diagrama de Atividade (Vídeo).
Fonte: Próprio Autor (2012)

5.3 Interfaces do Sistema

Neste capítulo serão apresentadas as interfaces do sistema, que foi desenvolvido a partir do embasamento teórico apresentado no decorrer do TCC.

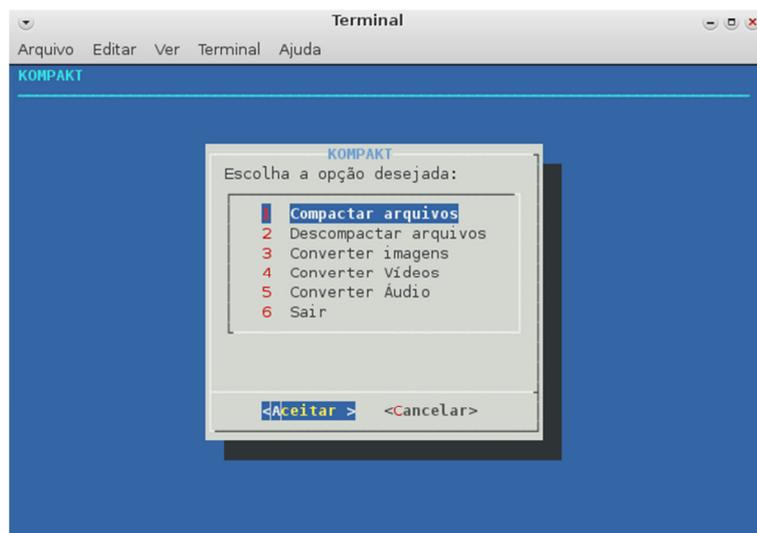


Figura 10 - Menu Principal.
Fonte: Próprio Autor (2012)

A figura 10 apresenta o menu principal do sistema, onde todas as funcionalidades podem ser acessadas.

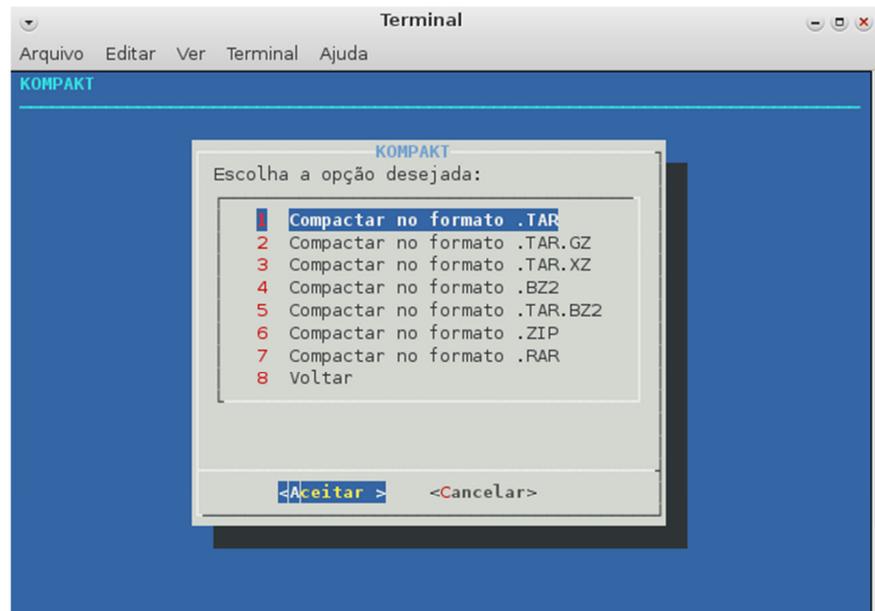


Figura 11 - Menu Secundário (Compactação).
Fonte: Próprio Autor (2012)

A figura 11 apresenta o menu secundário, responsável por exibir quais são as extensões disponíveis para o usuário compactar seus arquivos ou diretórios.

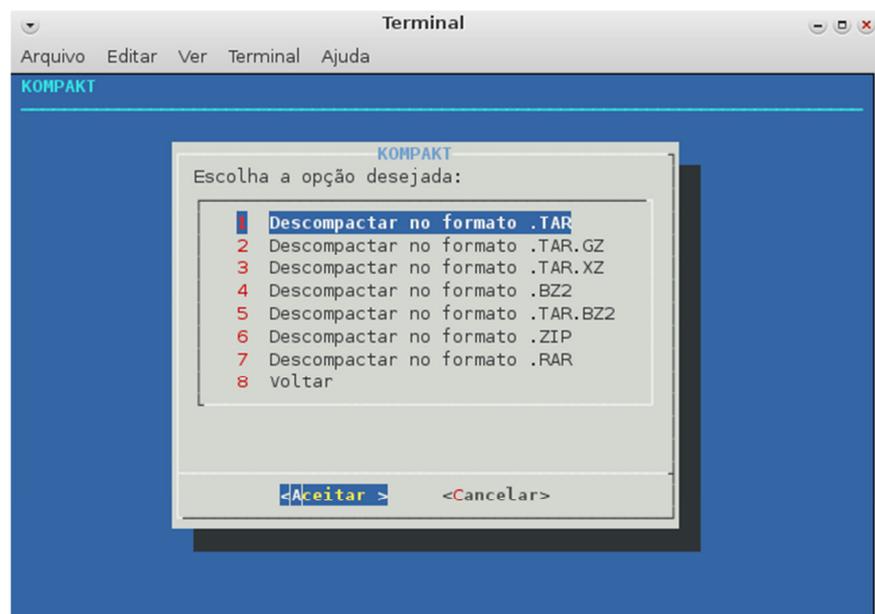


Figura 12 - Menu Secundário (Descompactação).
Fonte: Próprio Autor (2012)

A figura 12 apresenta o menu secundário, responsável por exibir quais são as extensões disponíveis para o usuário descompactar seus arquivos.

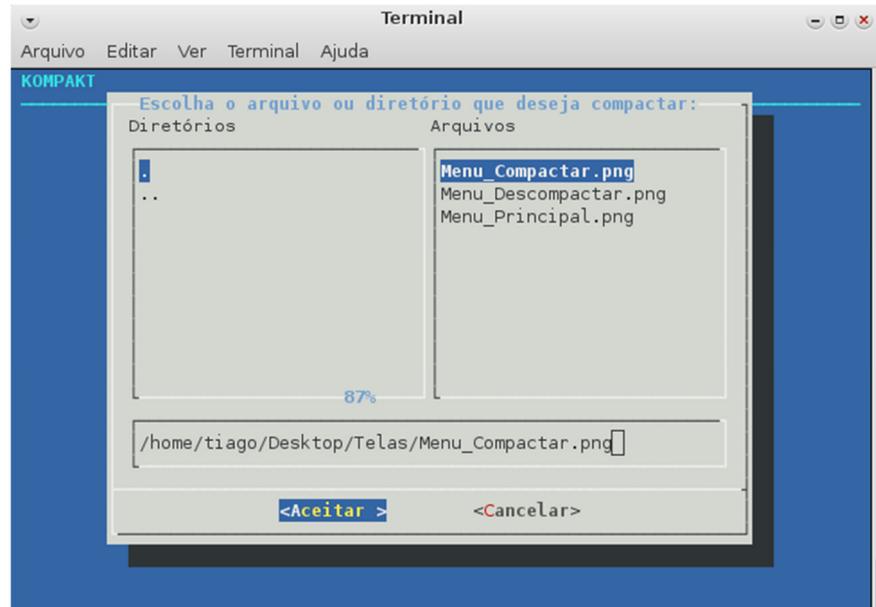


Figura 13 - Interface de escolha (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

A figura 13 é acionada após o acesso ao módulo de compactação e a escolha da extensão desejada para ser compactada, esta interface solicita que o usuário escolha o arquivo ou diretório que deseja compactar.

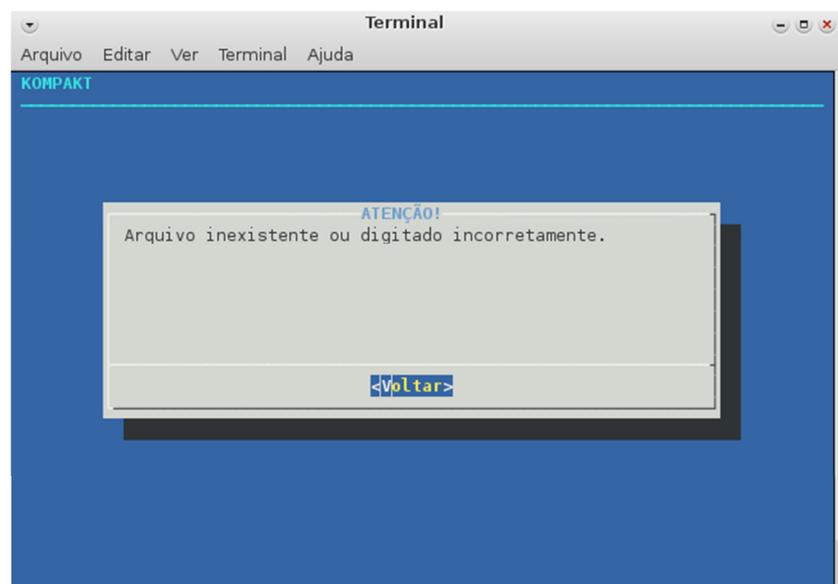


Figura 14 - Interface de erro (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

A figura 14 exibe uma mensagem de erro quando o arquivo ou diretório selecionado pelo usuário para ser compactado, não existe ou foi digitado incorretamente, caso o arquivo ou diretório sejam validados, a figura 15 é acionada.

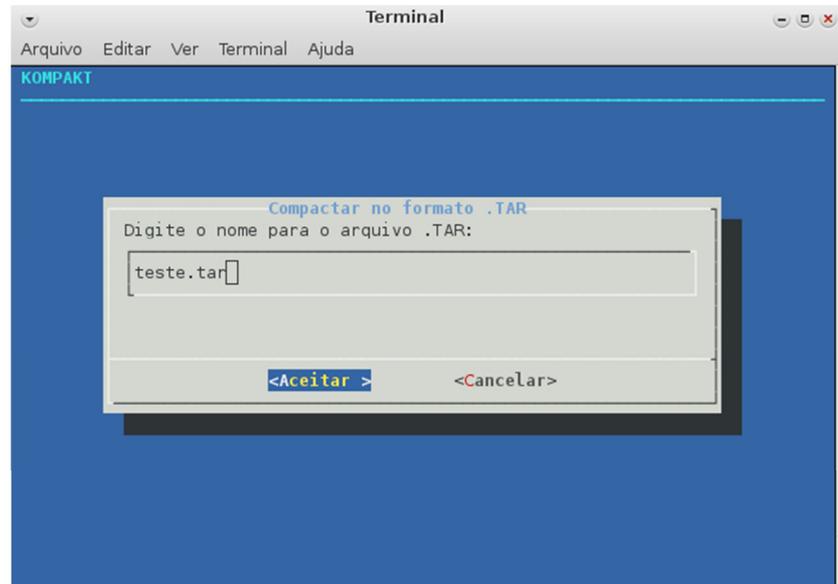


Figura 15 - Interface de escolha do nome (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

A figura 15 solicita que o usuário digite o nome do novo arquivo, que deve ser escrito juntamente com a extensão desejada. Caso seja escrito errado, aciona a figura 16.

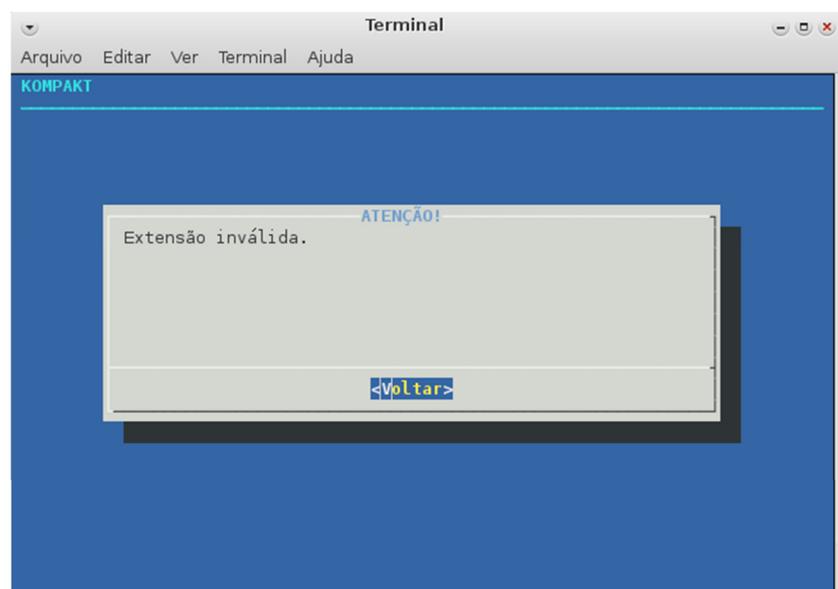


Figura 16 - Interface de erro da extensão (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

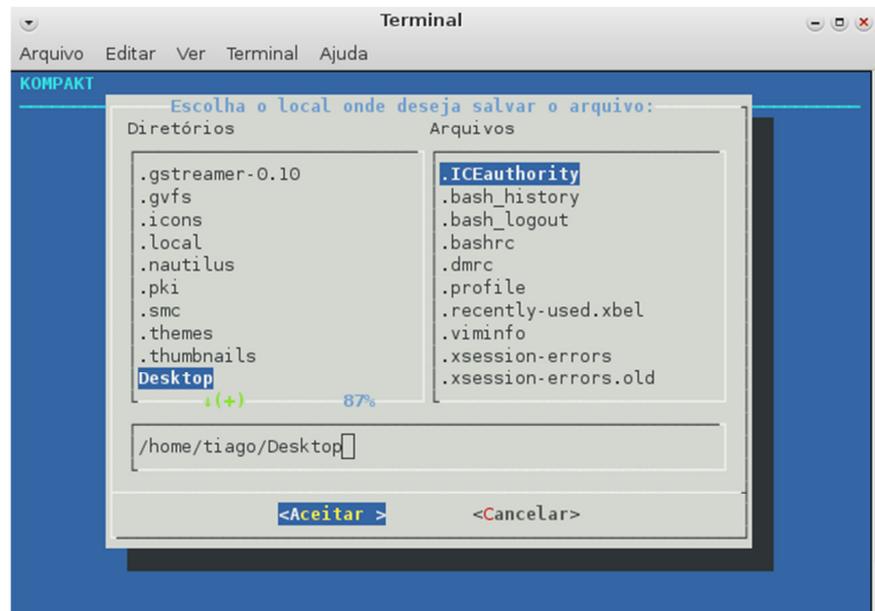


Figura 17 - Salvar arquivo compactado.
Fonte: Próprio autor (2012)

A figura 17 apresenta uma interface que solicita que o usuário entre com um caminho válido para salvar o novo arquivo.

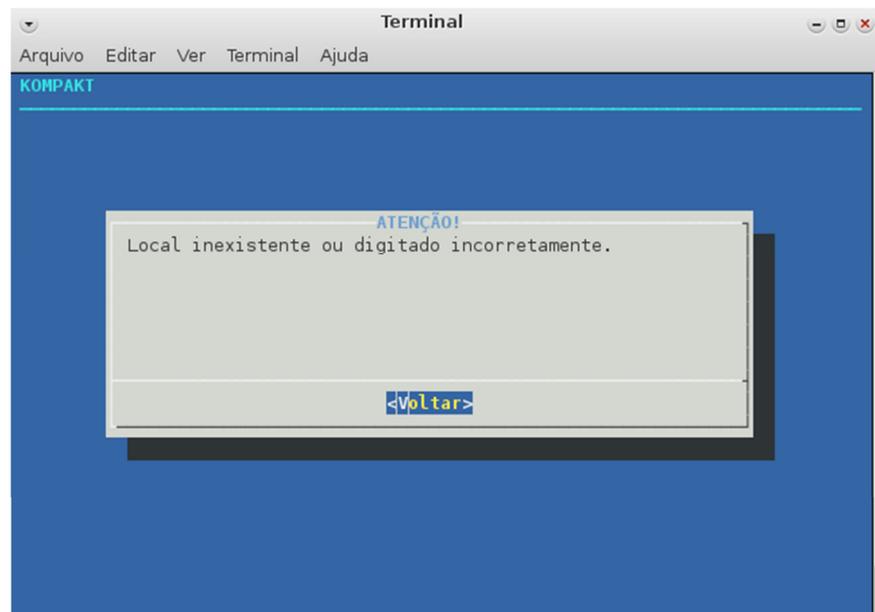


Figura 18 - Interface erro , local inexistente (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

A figura 18 mostra a interface de erro que será apresentada caso o diretório selecionado pelo usuário, para salvar o arquivo, não exista ou seja digitado incorretamente.

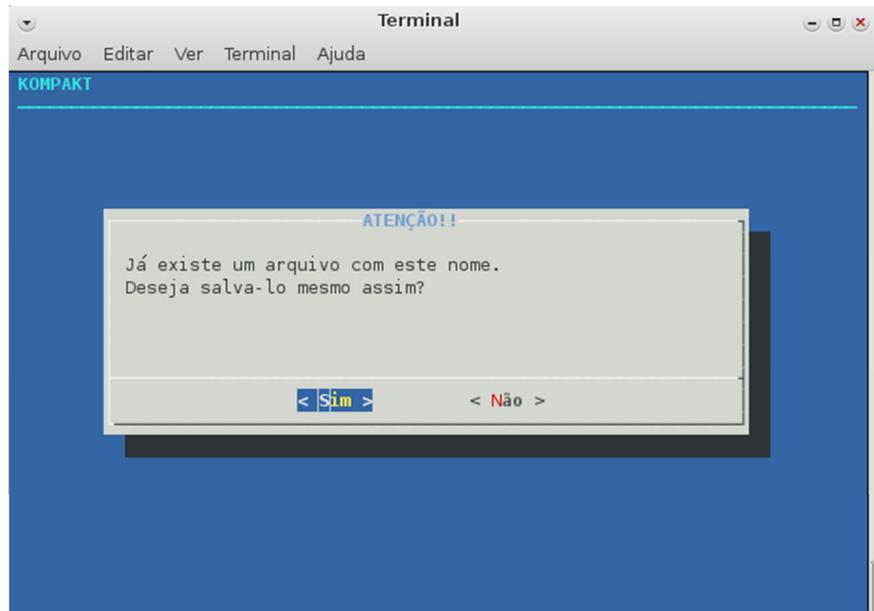


Figura 19 - Duplicata (arquivo a ser compactado).
Fonte: Próprio Autor (2012)

A figura 19 apresenta a interface exibida caso exista algum arquivo com o mesmo nome no diretório escolhido pelo usuário, nesta etapa o usuário poderá prosseguir, onde o arquivo existente será substituído, ou recusar e retornar ao menu secundário.

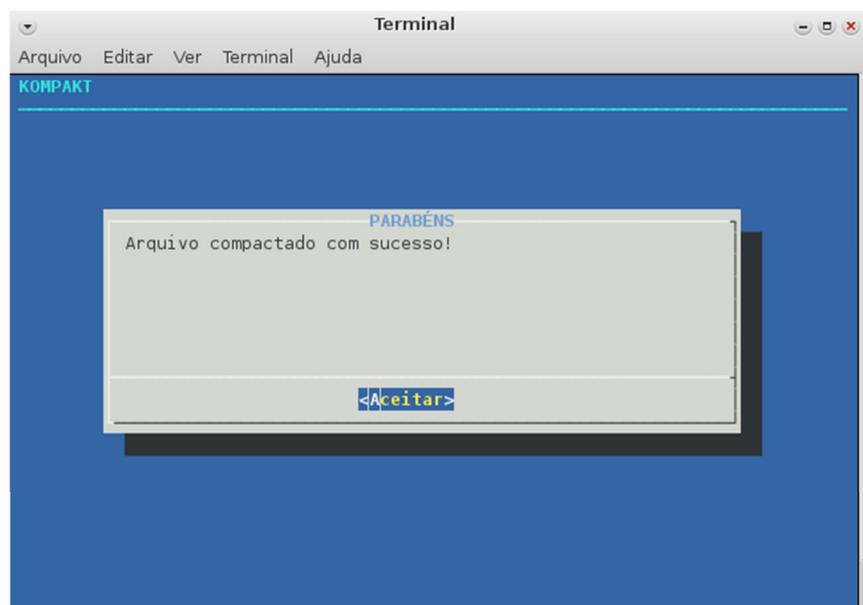


Figura 20 - Arquivo compactado
Fonte: Próprio Autor (2012)

A figura 20 é exibida quando todos os processos deste módulo foram realizados e o arquivo compactado com sucesso.

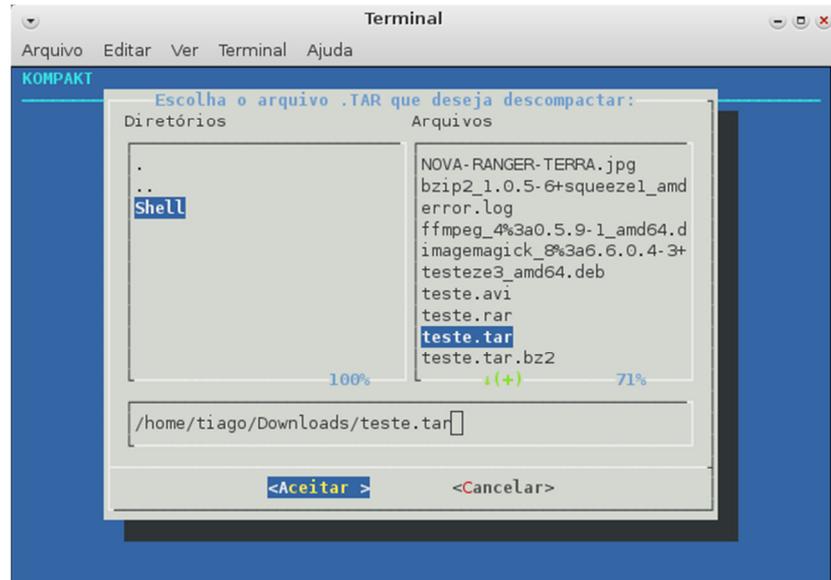


Figura 21 - Selecionar arquivo para descompactar
Fonte: Próprio Autor (2012)

A figura 21 apresenta o início do módulo de descompactação, onde o usuário deve selecionar um arquivo válido, conforme sua escolha no menu secundário.

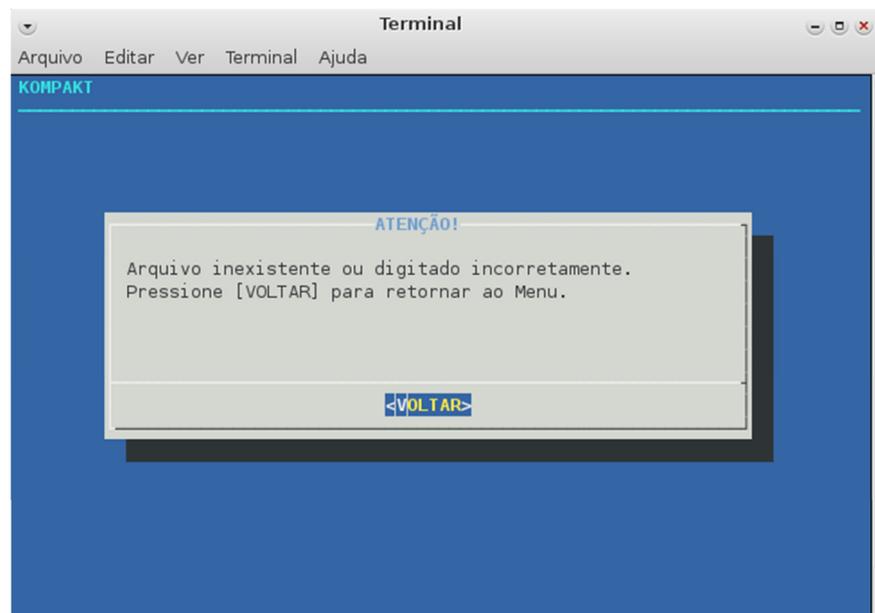


Figura 22 - Interface de erro, divergências com o arquivo (Descompactar).
Fonte: Próprio Autor (2012)

A figura 22 exibe a interface de erro que é solicitada no momento em que o arquivo selecionado pelo usuário não existe ou foi digitado incorretamente.

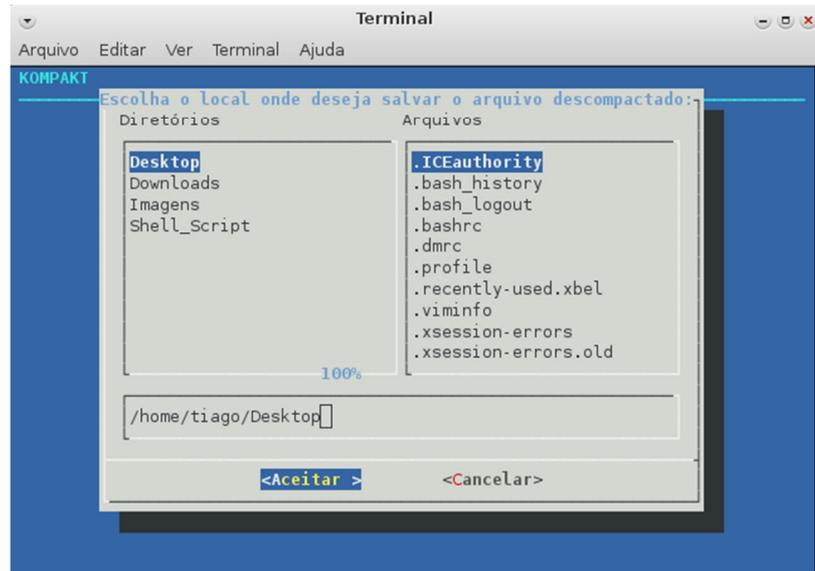


Figura 23 - Salvar arquivo descompactado
Fonte: Próprio Autor (2012)

A figura 23 exige que o usuário entre com um caminho válido para salvar o arquivo descompactado, caso o caminho não seja válido ou o diretório não exista, será exibida a interface da figura 24.

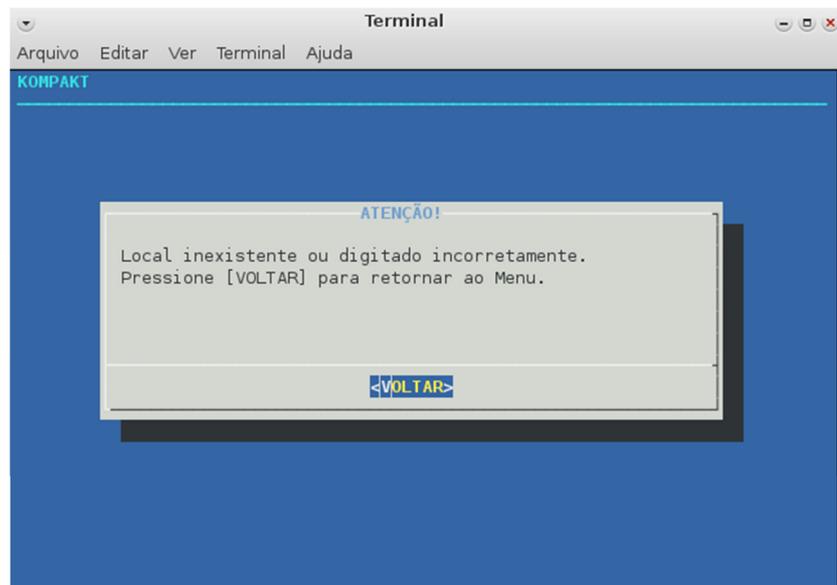


Figura 24 - Interface erro, problemas com diretório (Descompactar).
Fonte: Próprio Autor (2012)

A figura 24 exibe a interface de erro que é acionada quando não é válido o caminho selecionado pelo usuário para salvar o arquivo descompactado.

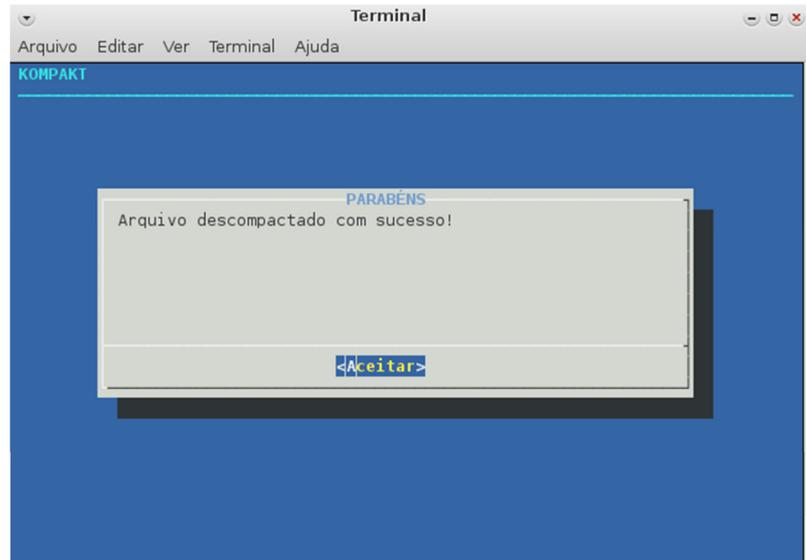


Figura 25 - Arquivo Descompactado.
Fonte: Próprio Autor (2012)

A figura 25 é exibida quando o arquivo é descompactado com sucesso.

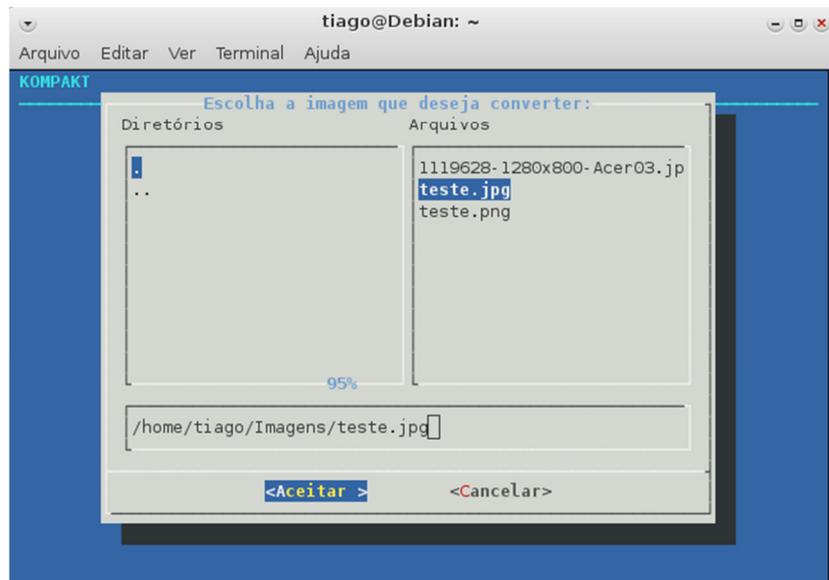


Figura 26 - Escolher imagem para conversão
Fonte: Próprio Autor (2012)

A figura 26 apresenta a tela onde o usuário escolhe em seu computador qual imagem deseja converter, entrando com um caminho válido conforme esta na figura, caso a imagem

escolhida não exista ou seja digitada incorretamente, o sistema vai chamar uma tela de atenção com o erro, que encaminhará o usuário ao menu para refazer a operação, conforme é mostrado na figura 27.

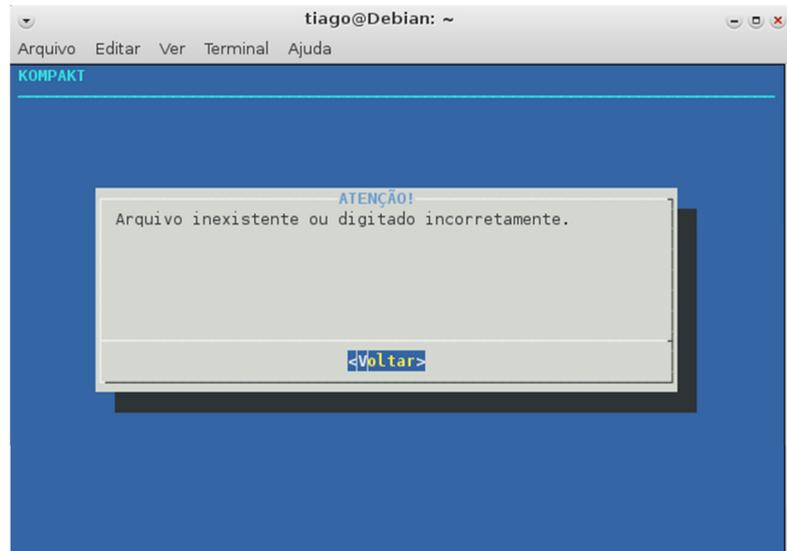


Figura 27 - Interface de erro, divergências na imagem.
Fonte: Próprio Autor (2012)

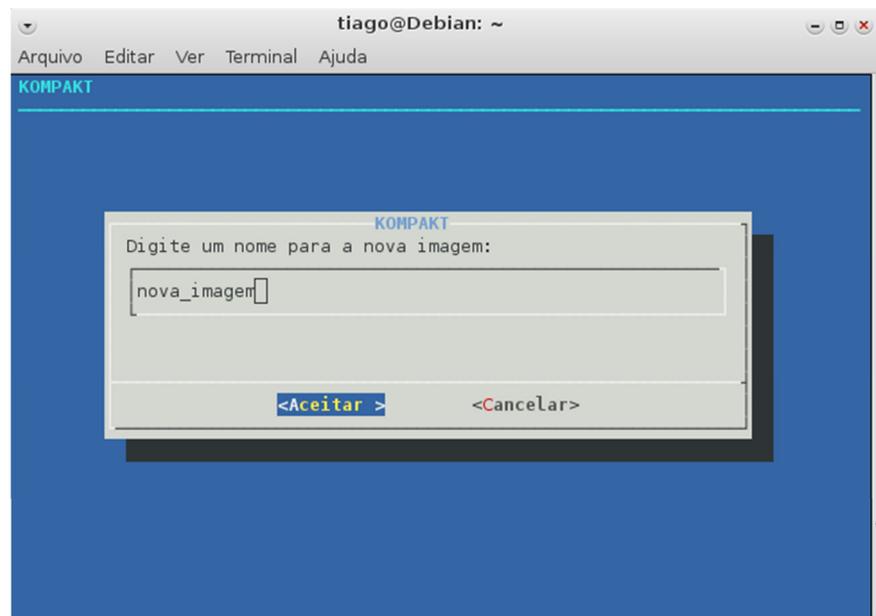


Figura 28 - Nome nova imagem
Fonte: Próprio Autor (2012)

A figura 28 pede ao usuário que digite um nome para a imagem que será convertida, nesta etapa não é necessário que entre com a extensão do arquivo, somente o nome. A extensão será escolhida nas próximas telas.

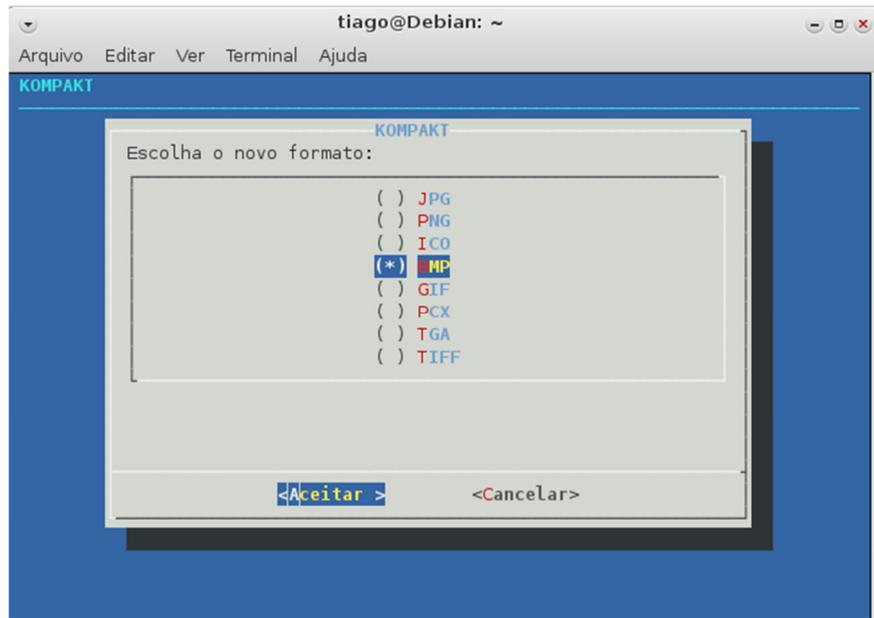


Figura 29 - Formato da imagem
Fonte: Próprio Autor (2012)

A figura 29 exibe as opções de extensão para a imagem que o usuário poderá escolher.

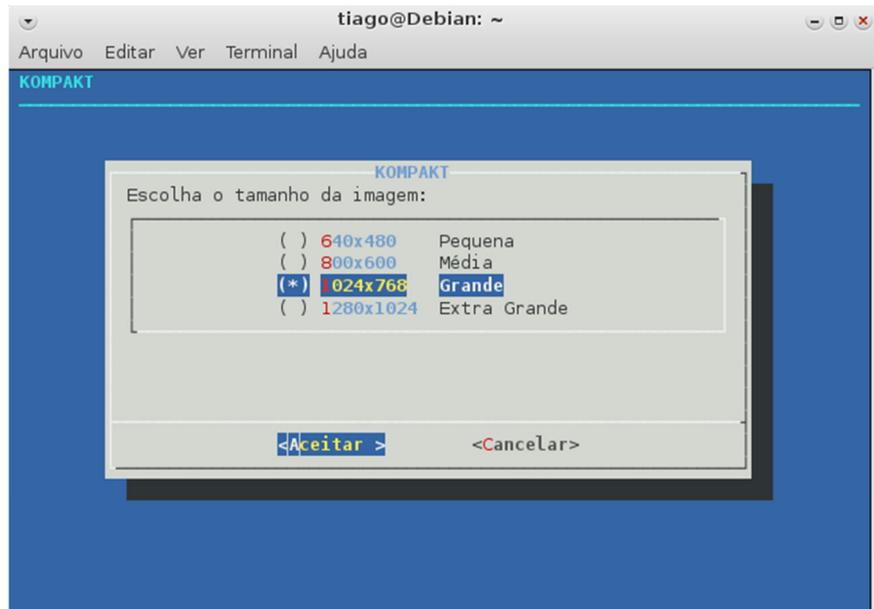


Figura 30 - Tamanho da imagem.
Fonte: Próprio Autor (2012)

A figura 30 pede ao usuário que escolha a dimensão para a nova imagem.

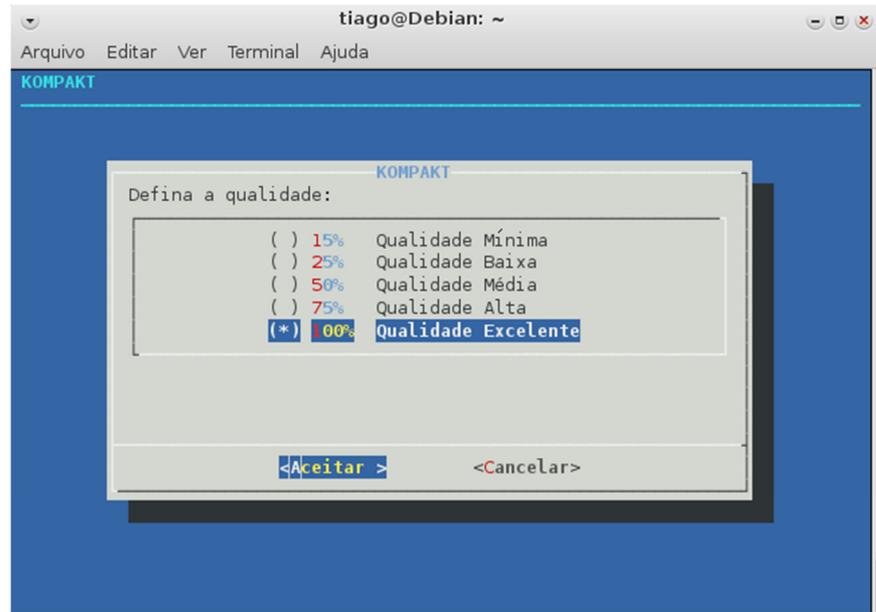


Figura 31 - Qualidade da Imagem
Fonte: Próprio Autor (2012)

A figura 31 exibe a interface que solicita ao usuário que escolha a qualidade que será aplicada a nova imagem.

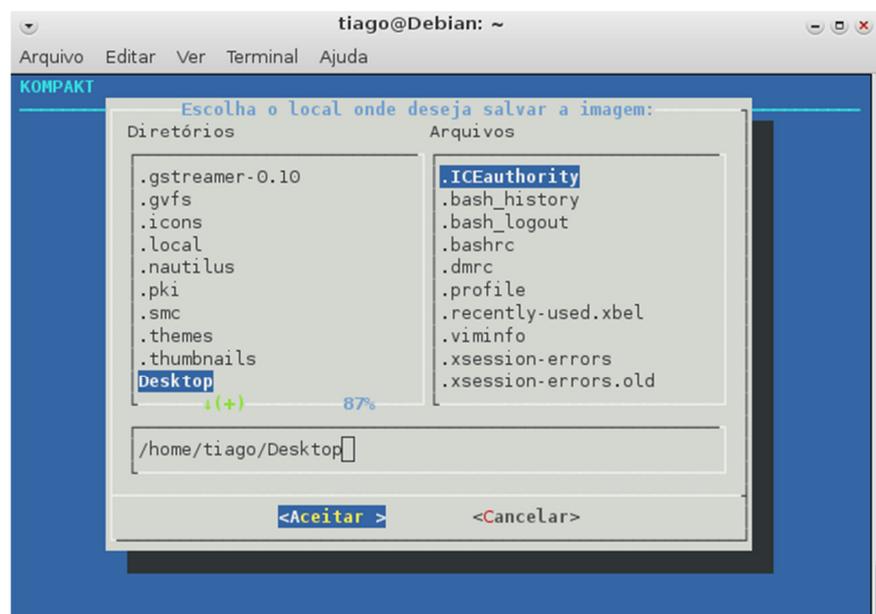
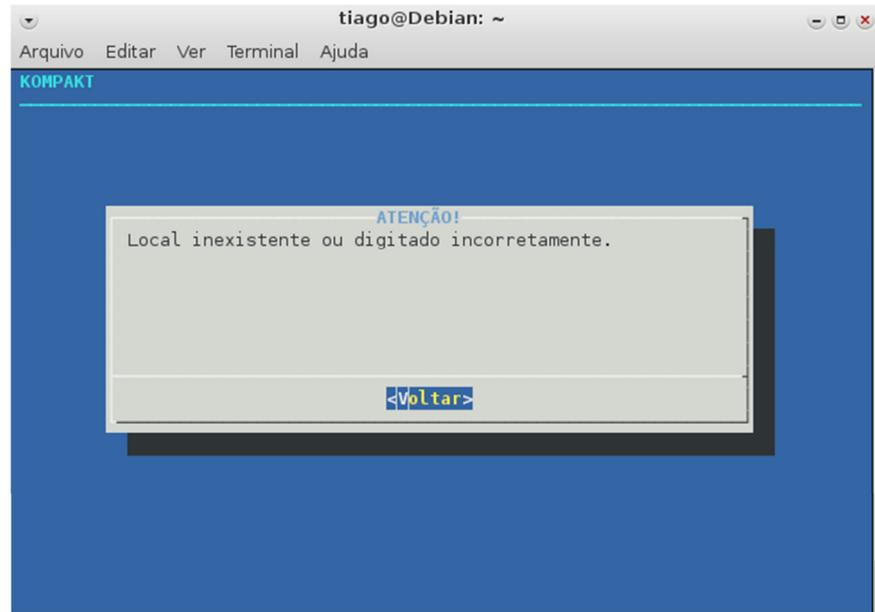


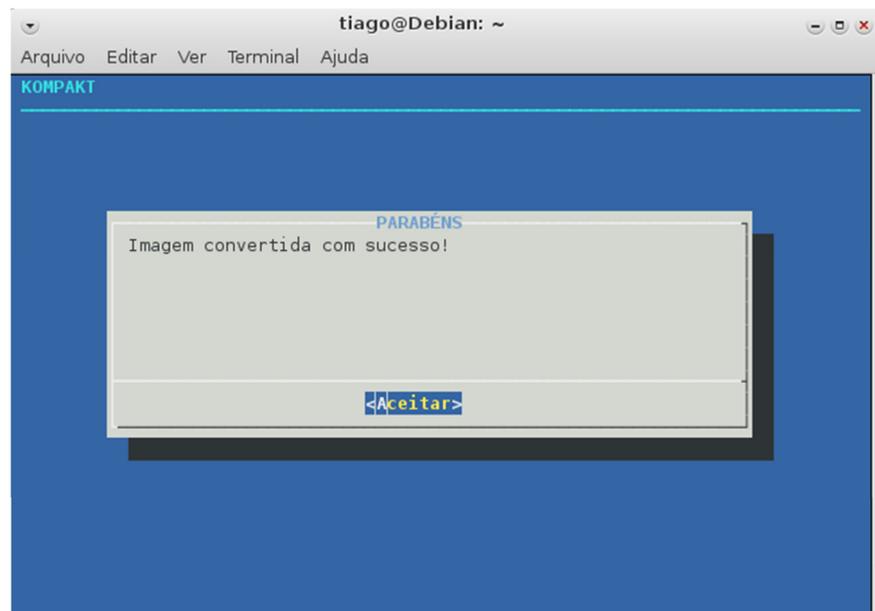
Figura 32 - Salvar Imagem
Fonte: Próprio Autor (2012)

A figura 32 apresenta a interface que pede ao usuário que defina um caminho válido para salvar a imagem convertida, caso o caminho não seja válido, o sistema apresenta uma

interface de erro, para que o usuário retorne ao menu e refaça o processo correto, conforme é mostrado na figura 33.



**Figura 33 - Interface de erro, divergências com local para salvar a imagem.
Fonte: Próprio Autor (2012)**



**Figura 34 - Imagem convertida
Fonte: Próprio Autor (2012)**

A figura 34 apresenta a interface em que a imagem foi convertida com sucesso.

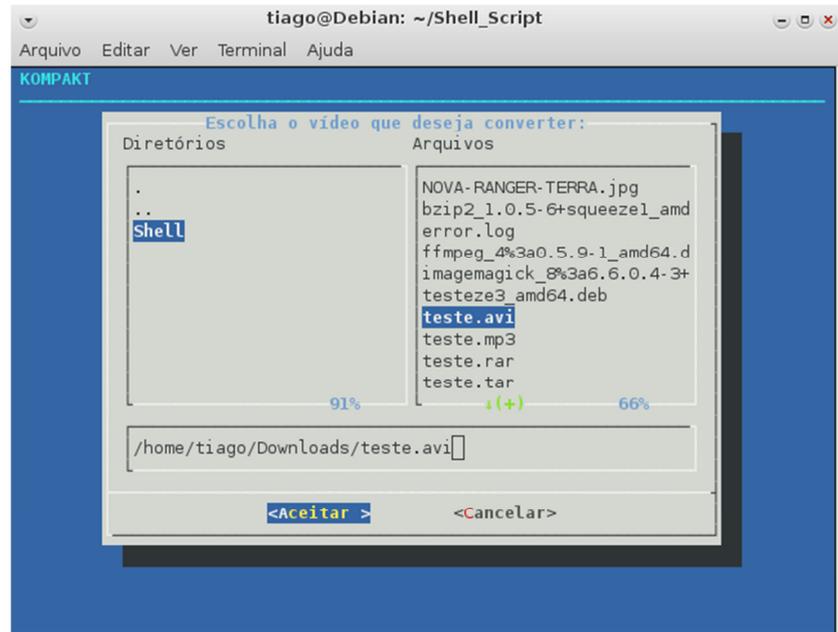


Figura 35 - Escolher vídeo para converter
Fonte: Próprio Autor (2012)

A figura 35 exibe a interface que solicita ao usuário que entre com um caminho válido, para escolher o vídeo que será convertido, caso o caminho não seja válido ou o vídeo não exista, será exibida uma interface de erro, encaminhando o usuário ao menu inicial para refazer o processo.

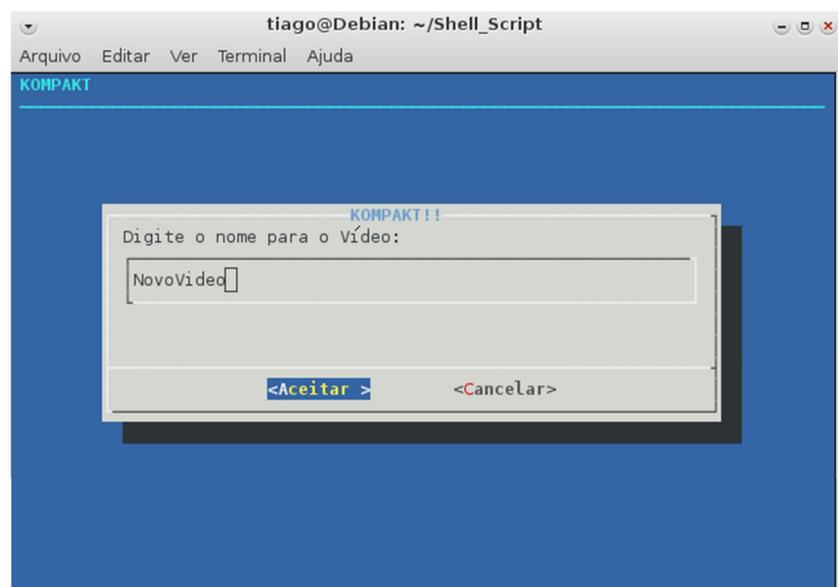


Figura 36 - Nome para o novo vídeo.
Fonte: Próprio autor (2012)

A figura 36 apresenta a interface que solicita ao usuário que entre com o nome para o novo vídeo, não é necessário digitar a extensão juntamente com o nome, pois a mesma será escolhida na próxima tela.



Figura 37 - Formato do vídeo.
Fonte: Próprio Autor (2012)

A figura 37 exibe a interface que solicita ao usuário que entre com a extensão desejada para o vídeo convertido.

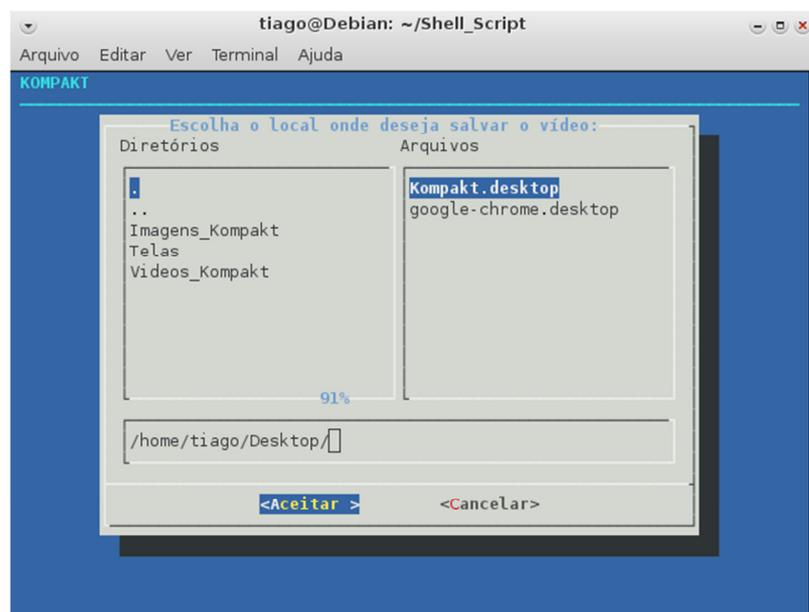


Figura 38 - Salvar vídeo.
Fonte: Próprio Autor (2012)

A figura 38 exibe a interface onde o usuário deve entrar com o caminho válido a um diretório, para salvar o vídeo convertido, caso este caminho não exista, o sistema retornará uma tela de erro.

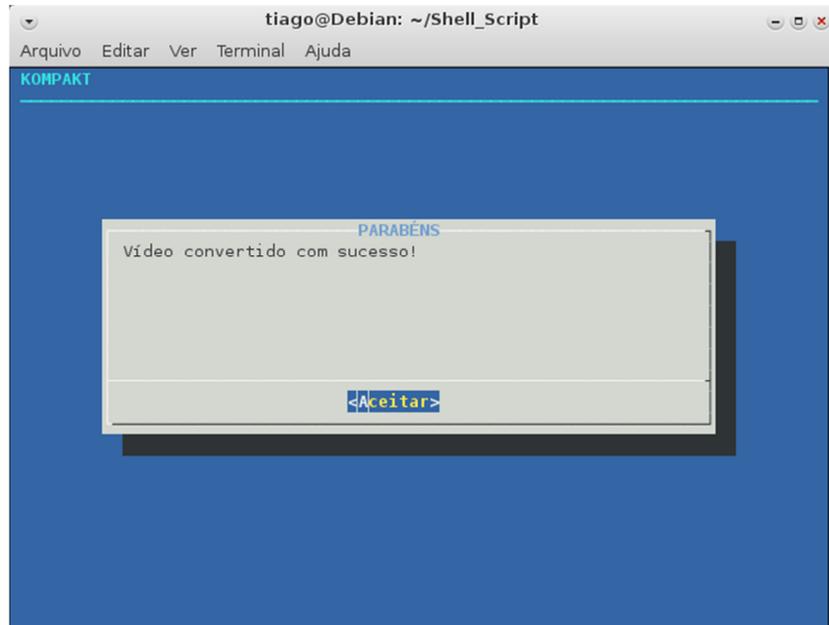


Figura 39 - Vídeo convertido.
Fonte: Próprio Autor (2012)

A figura 39 apresenta a interface informando que o vídeo foi convertido com sucesso.

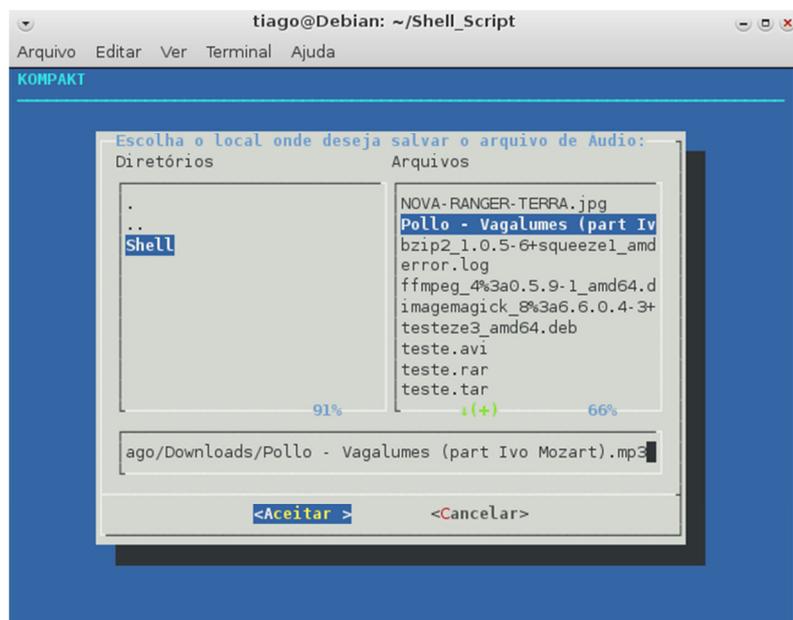


Figura 40 - Escolher Áudio para ser convertido.
Fonte: Próprio Autor (2012)

A figura 40 exibe a interface que pede ao usuário que selecione o arquivo de áudio que deseja converter.

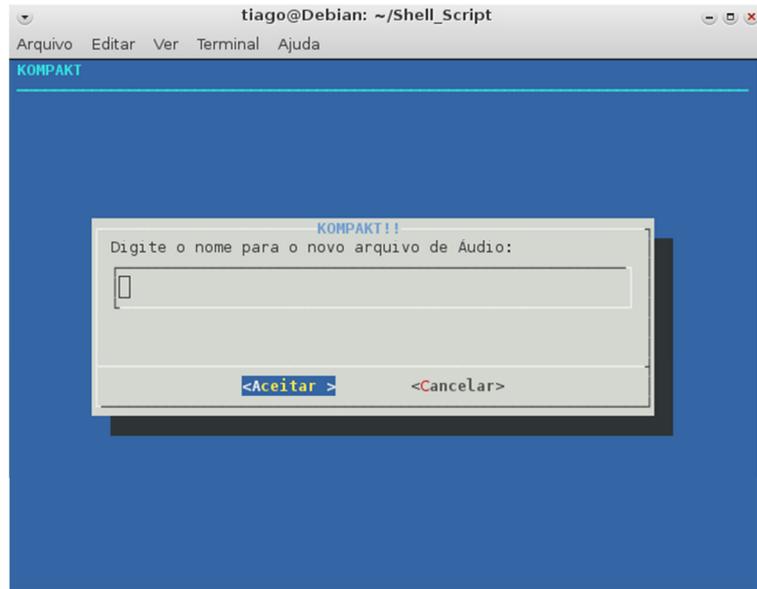


Figura 41 - Nome arquivo de áudio.
Fonte: Próprio Autor (2012)

A figura 41 apresenta a interface onde solicita que o usuário digite o nome para o novo arquivo de áudio, que será gerado após a conversão, não é necessário digitar a extensão.

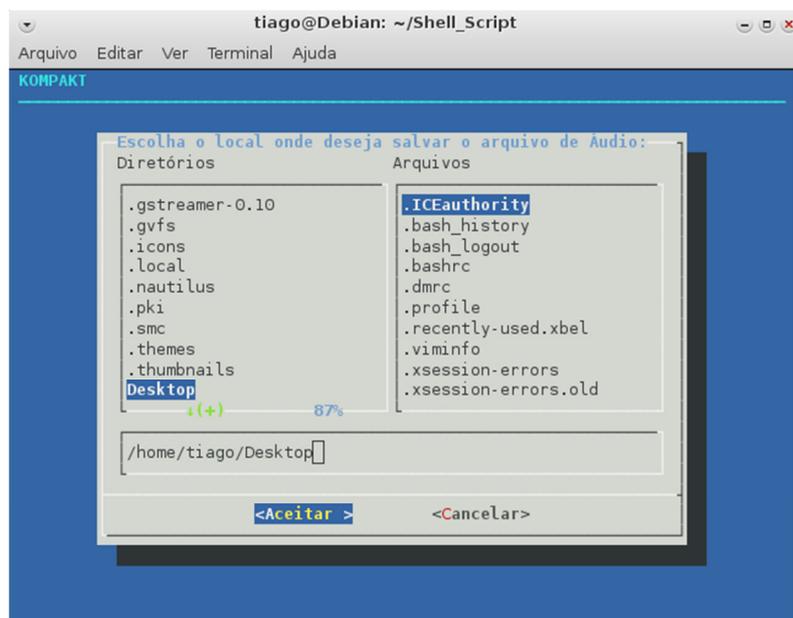


Figura 42 - Salvar arquivo de áudio.
Fonte: Próprio Autor (2012)

A figura 42 apresenta a interface onde o usuário definirá onde o arquivo convertido será salvo e a figura 43 informa que o arquivo foi convertido com sucesso.

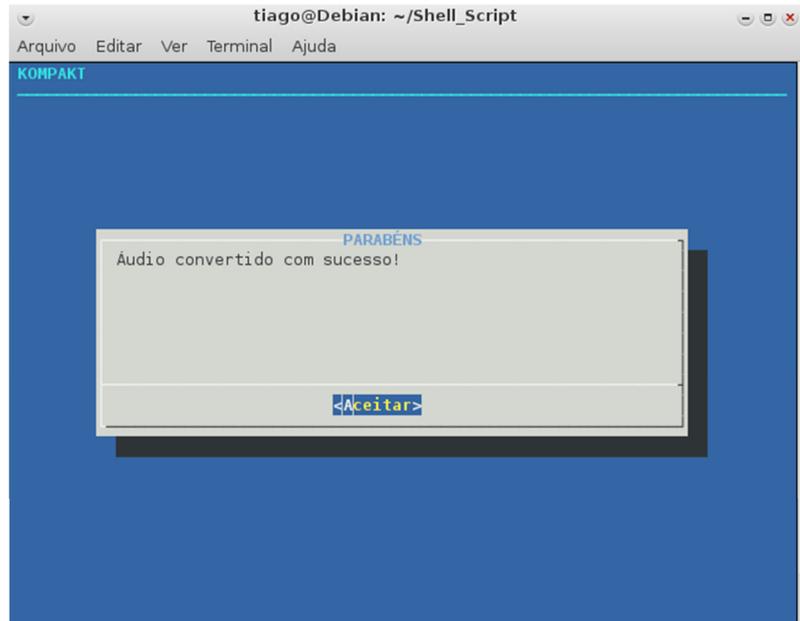


Figura 43 - Arquivo de áudio convertido.
Fonte: Próprio Autor (2012)

VI. CONCLUSÃO

Com o desenvolvimento deste sistema, pode-se verificar que para obter um resultado satisfatório, o analista deve buscar dados e informações que se aproximem o máximo possível das necessidades reais que o usuário vive no seu cotidiano, para tentar atender e suprir todas as suas necessidades e deficiências.

O uso deste sistema facilita e agiliza as atividades realizadas no cotidiano referente às funcionalidades oferecidas, tanto do usuário final como do atendente de T.I, por ser rápido e prático se torna um aplicativo de fácil adaptação.

Após o seu uso notou-se uma significativa diminuição de atendimentos relacionados às funcionalidades do sistema.

Sendo assim, é evidente que após a troca de sistema operacional, independente de qual S.O se refere, é necessário que o usuário tenha a disponibilidade de ferramentas de fácil uso e assimilação, que o auxilie nas atividades do cotidiano, para que o processo seja menos cansativo, evitando que o usuário precise de ajuda constante para realizar seu trabalho e o tornando mais independente.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

HEXSEL, A, R. **Software Livre, Propostas de Ações de Governo para Incentivar o Uso de Software Livre.** “[http://www.inf.ufpr.br/info/techrep/RT_DINF004_2002.pdf]” Acesso em: 15/09/12.

YIN, R. K. **Estudo de Caso: Planejamento e Métodos.** 2ª ed. Porto Alegre: Bookman, 2001. 31 p.

CASTILHO, M. **A importância da Modelagem no Planejamento de Desenvolvimento de Software.**

“[<http://www.dimensaotech.com/2008/08/modelagem-de-software>]” Acessado em: 20/09/12.

FERREIRA, E, R. **Linux – Guia do Administrador do Sistema.** 1ª ed. Editora: Novatec 2003.

FLYNN, I & MCHOES, M, A. **Introdução aos Sistemas Operacionais.** Editora: Cengage Learning 2002.

SILVEIRA, A, S. **Software Livre, A Luta Pela Liberdade do Conhecimento.** Editora: Fundação Perseu Abramo 2004.

GONÇALVES, L, E, J. **As empresas são grandes coleções de processos.**

Revista de Administração de Empresas. São Paulo 2000.

NASSIF, K. **Análise de processos gera interfaces amigáveis.**

“[<http://webinsider.uol.com.br/index.php/2004/04/18/analise-de-processos-gera-interfaces-amigaveis>]”. Acesso em: 25/09/12.

MAGARÃO, M. **Gestão por processo: um mecanismo de análise.**

“[http://www.eceme.ensino.eb.br/cpeaex/downloads/ead/adm_09_2006.pdf]”. Acesso em: 26/09/12

PETRARCA, I. **Apostila de Organização e Métodos.** UNILASALLE 2004.

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. Editora: Campus 2006.

JARGAS, M, A. **Shell Script Profissional**. Editora: Novatec 2008.

LIDA, I. **Ergonomia, Projeto e Produção**. 2^a ed. Editora: Edgard Blücher 2005.

VIII. ANEXOS

8.1 Código fonte Kompakt

```

#-----#
#!/bin/bash #
# #
# #
# #
#-----#

#Inicio do Menu Principal

MENU() {
clear
    OPCAO=$(dialog \
        --backtitle 'KOMPAKT' \
        --title 'KOMPAKT' \
        --stdout \
        --menu 'Escolha a opção desejada:' \
        0 0 \
        1 'Compactar arquivos' \
        2 'Descompactar arquivos' \
        3 'Converter imagens' \
        4 'Converter Vídeos' \
        5 'Converter Áudio' \
        6 'Sair' )

    retorno=$?
    [ $retorno -ne 0 ] && exit

    case $OPCAO in
        1)COMPACTAR;;
        2)DESCOMPACTAR;;
        3)CONVERT_IMAGEM;;
        4)CONVERT_VIDEO;;
        5)CONVERT_AUDIO;;
        6)exit;;
    esac
}

#Fim Menu Principal

#-----#
#
#Inicio do Menu Secundario - Compactar

COMPACTAR() {
clear
    OPCAO1=$(dialog \
        --backtitle 'KOMPAKT' \
        --title 'KOMPAKT' \
        --stdout \
        --menu 'Escolha a opção desejada:' \
        0 0 \
        1 'Compactar no formato .TAR' \
        2 'Compactar no formato .TAR.GZ' \
        3 'Compactar no formato .TAR.XZ' \
        4 'Compactar no formato .BZ2' \
        5 'Compactar no formato .TAR.BZ2' \
        6 'Compactar no formato .ZIP' \
        7 'Compactar no formato .RAR' \
        8 'Voltar' )

    retorno=$?
    [ $retorno -ne 0 ] && MENU
}

#Fim do Menu Secundario - Compactar

#-----#
#
#Inicio do Menu Secundario - Descompactar

DESCOMPACTAR() {
clear
    OPCAO2=$(dialog \
        --backtitle 'KOMPAKT' \
        --title 'KOMPAKT' \
        --stdout \
        --menu 'Escolha a opção desejada:' \
        0 0 \
        1 'Descompactar no formato .TAR' \
        2 'Descompactar no formato .TAR.GZ' \
        3 'Descompactar no formato .TAR.XZ' \
        4 'Descompactar no formato .BZ2' \
        5 'Descompactar no formato .TAR.BZ2' \
        6 'Descompactar no formato .ZIP' \
        7 'Descompactar no formato .RAR' \
        8 'Voltar' )

    retorno=$?
    [ $retorno -ne 0 ] && MENU
}

#Fim do Menu Secundario - Descompactar

#-----#
#
# Inicio das Functions que tem a finalidade de Compactar as
# extensões #
# .TAR.GZ, .TAR, .BZ2, .TAR.BZ2, .RAR, .ZIP
#
#-----#
#
case $OPCAO1 in
1)COMPACT_TAR;;
2)COMPACT_TARGZ;;
3)COMPACT_TARXZ;;
4)COMPACT_BZ2;;
5)COMPACT_TARBZ2;;
6)COMPACT_ZIP;;
7)COMPACT_RAR;;
8)MENU ;;
esac
}

#Fim do Menu Secundario - Compactar

#-----#
#
case $OPCAO2 in
1)DESCOMPACT_TAR;;
2)DESCOMPACT_TARGZ;;
3)DESCOMPACT_TARXZ;;
4)DESCOMPACT_BZ2;;
5)DESCOMPACT_TARBZ2;;
6)DESCOMPACT_ZIP;;
7)DESCOMPACT_RAR;;
8)MENU ;;
esac
}

#Fim do Menu Secundario - Descompactar

#-----#
#

```



```

                                dialog
\                                --backtitle 'KOMPAKT'
\                                --title 'PARABÉNS'
\                                --msgbox 'Arquivo compactado com
sucesso!' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] &&
COMPACTAR
                                [ $retorno -eq 255 ] &&
COMPACTAR

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Local inexistente ou digitado
incorretamente.' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
                                [ $retorno -eq 255 ] && COMPACTAR

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Extensão invalida.'
\
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
                                [ $retorno -eq 255 ] && COMPACTAR

                                fi

else

dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'Voltar'
--msgbox 'Arquivo inexistente ou digitado incorretamente.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi
}

# Final da Function que compacta a extensão .TAR.GZ
#-----#

# Inicio da function compacta na extensão .TAR
COMPACT_TAR() {
clear

NOME_TAR=$(dialog \
                                --backtitle 'KOMPAKT'
                                --title 'Escolha o arquivo ou diretório que deseja
compactar:'\
                                --stdout
                                --fselect /home/
                                10 60
                                )

                                retorno=$?
                                [ $retorno -ne 0 ] && COMPACTAR

DIRETORIO=`dirname $NOME_TAR`
ARQUIVO=`basename $NOME_TAR`

cd $DIRETORIO
if [ -e $ARQUIVO ] ;
then

NOME_COMPACT_TAR=$(dialog \
--backtitle 'KOMPAKT'
--title 'Compactar no formato .TAR'
--stdout
--inputbox 'Digite o nome para o arquivo .TAR:'
10 60
)

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

if [ ${NOME_COMPACT_TAR#*.} = "tar" ] ;
then

LOCAL=$(dialog \
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar o arquivo:'
\
--stdout
--fselect /home/
10 60
)

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

if [ -e $LOCAL ] && [ -d $LOCAL ] ;
then

#Variavel que verifica qual usuario esta
logado.
USER=`whoami`

cd $LOCAL
if [ -e Compactados_Kompakt ]
&& [ -d Compactados_Kompakt ] ;
then
cd Compactados_Kompakt

if [ -e
$NOME_COMPACT_TAR ] ;
then
dialog \
--backtitle
\
--title 'ATENÇÃO!!'
\
--yesno "\nJa existe
um arquivo com este nome.
\nDeseja salva-lo
mesmo assim?"
10 60

retorno=$?
[ $retorno -ne 0 ]
&& COMPACTAR

cd $DIRETORIO
tar -cf
$NOME_COMPACT_TAR $ARQUIVO >> .kompakt.log

```

```

                                mv
$NOME_COMPACT_TAR $LOCAL/Compactados_Kompakt
                                chown
$USER.$USER $LOCAL/Compactados_Kompakt -R
                                cd ..

                                dialog
\
                                --backtitle 'KOMPAKT'
\
                                --title 'PARABÉNS'
\
                                --msgbox 'Arquivo compactado com
sucesso!' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
&& COMPACTAR
                                [ $retorno -eq 255 ]

                                else

                                cd $DIRETORIO
                                tar -cf $NOME_COMPACT_TAR
$ARQUIVO >> .kompakt.log
                                mv $NOME_COMPACT_TAR
$LOCAL/Compactados_Kompakt
                                chown $USER.$USER
$LOCAL/Compactados_Kompakt -R
                                cd ..

                                dialog
\
                                --backtitle
'KOMPAKT' \
                                --title 'PARABÉNS'
\
                                --msgbox 'Arquivo
compactado com sucesso!' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ]
&& COMPACTAR
                                [ $retorno -eq 255 ]
&& COMPACTAR

                                fi

                                else
                                mkdir Compactados_Kompakt
                                cd Compactados_Kompakt
                                cd $DIRETORIO
                                tar -cf $NOME_COMPACT_TAR $ARQUIVO
>> .kompakt.log
                                mv $NOME_COMPACT_TAR
$LOCAL/Compactados_Kompakt
                                chown $USER.$USER
$LOCAL/Compactados_Kompakt -R
                                cd ..

                                dialog
\
                                --backtitle 'KOMPAKT'
\
                                --title 'PARABÉNS'
\
                                --msgbox 'Arquivo compactado com
sucesso!' \
                                10 60

                                retorno=$?

```

```

                                [ $retorno -eq 0 ] &&
COMPACTAR
                                [ $retorno -eq 255 ] &&
COMPACTAR

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Local inexistente ou digitado
incorretamente.' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
                                [ $retorno -eq 255 ] && COMPACTAR

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Extensão invalida.'
\
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
                                [ $retorno -eq 255 ] && COMPACTAR

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Arquivo inexistente ou digitado incorretamente.' \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && COMPACTAR
                                [ $retorno -eq 255 ] && COMPACTAR

                                fi
}

# Final da Function que compacta arquivos no formato .TAR
# -----
-#

# Inicio da Function que compacta arquivos no formato TAR.XZ

COMPACT_TARXZ() {
clear

                                NOME_TARXZ=$(dialog
                                --backtitle 'KOMPAKT'
                                --title 'Escolha o arquivo ou diretório que deseja
compactar:\
                                --stdout
                                --fselect /home/
                                10 60
                                )

                                retorno=$?
                                [ $retorno -ne 0 ] && COMPACTAR

                                DIRETORIO_TARXZ=`dirname $NOME_TARXZ`

```



```

--title 'ATENÇÃO!' \
--ok-label 'Voltar' \
--msgbox 'Local inexistente ou digitado \
incorretamente.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'Voltar' \
--msgbox 'Extensão invalida.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'Voltar' \
--msgbox 'Arquivo inexistente ou digitado incorretamente.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi
}

# Final da Function que compacta arquivos no formato TAR.XZ
# -----
-#

# Inicio da Function que compacta arquivos no formato .BZ2
COMPACT_BZ2() {
clear

NOME_BZ2=$(dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo que deseja compactar:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

DIRETORIO_BZ2=`dirname $NOME_BZ2`
ARQUIVO_BZ2=`basename $NOME_BZ2`

cd $DIRETORIO_BZ2
if [ -e $ARQUIVO_BZ2 ];
then

#NOME_COMPACT_BZ2=$(dialog \
#--backtitle 'KOMPAKT' \
#--title 'Compactar no formato .BZ2' \
#--stdout \
#--inputbox 'Digite o nome para o arquivo .BZ2:' \
#10 60 )

#retorno=$?
#[ $retorno -ne 0 ] && COMPACTAR

# if [ ${NOME_COMPACT_BZ2#*.} = "bz2" ];
# then

LOCAL=$(dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

if [ -e $LOCAL ] && [ -d $LOCAL ];
then

#Variavel que verifica qual usuario esta
logado.

USER=`whoami`

cd $LOCAL
if [ -e Compactados_Kompakt ]
&& [ -d Compactados_Kompakt ];
then
cd Compactados_Kompakt

if [ -e
$NOME_BZ2.bz2 ];
then
dialog \
--backtitle \
--title 'ATENÇÃO!!' \
--yesno "\nJa existe
um arquivo com este nome.
\nDeseja salva-lo
mesmo assim?" \
10 60

retorno=$?
[ $retorno -ne 0 ]
&& COMPACTAR

cd $DIRETORIO_BZ2
bzip2 -z
$NOME_BZ2 >> kompakt.log
cp -a
$NOME_BZ2.bz2 $LOCAL/Compactados_Kompakt
chown
$USER.$USER $LOCAL/Compactados_Kompakt -R
cd ..

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo compactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ]
&& COMPACTAR

```

```

else
    # else
    # dialog
    # --backtitle 'KOMPAKT'
    # --title 'ATENÇÃO!'
    # --ok-label 'Voltar'
    # --msgbox 'Extensão invalida.'
    # 10 60
    # retorno=$?
    # [ $retorno -eq 0 ] && COMPACTAR
    # [ $retorno -eq 255 ] && COMPACTAR
    # fi

    else
    dialog
    --backtitle 'KOMPAKT'
    --title 'PARABÉNS'
    --msgbox 'Arquivo compactado com sucesso!'
    10 60
    retorno=$?
    [ $retorno -eq 0 ]
    && COMPACTAR
    [ $retorno -eq 255 ]
    && COMPACTAR
    fi

    else
    mkdir Compactados_Kompakt
    cd Compactados_Kompakt
    cd $DIRETORIO_BZ2
    bzip2 -z $NOME_BZ2 >> .kompakt.log
    cp -a $NOME_BZ2.bz2
    $LOCAL/Compactados_Kompakt
    chown $USER.$USER
    $LOCAL/Compactados_Kompakt -R
    cd ..

    dialog
    --backtitle 'KOMPAKT'
    --title 'PARABÉNS'
    --msgbox 'Arquivo compactado com sucesso!'
    10 60
    retorno=$?
    [ $retorno -eq 0 ] && COMPACTAR
    [ $retorno -eq 255 ] && COMPACTAR
    fi

    else
    dialog
    --backtitle 'KOMPAKT'
    --title 'ATENÇÃO!'
    --ok-label 'Voltar'
    --msgbox 'Local inexistente ou digitado incorretamente.'
    10 60
    retorno=$?
    [ $retorno -eq 0 ] && COMPACTAR
    [ $retorno -eq 255 ] && COMPACTAR
    fi

    # Final da Function que compacta arquivos no formato .BZ2
    # -----
    # Inicio da Function que compacta arquivos no formato .TAR.BZ2
    COMPACT_TARBZ2() {
    clear
    NOME_TARBZ2=$(dialog
    --backtitle 'KOMPAKT'
    --title 'Escolha o arquivo ou diretório que deseja compactar:'
    --stdout
    --fselect /home/
    10 60
    retorno=$?
    [ $retorno -ne 0 ] && COMPACTAR
    DIRETORIO_TARBZ2=`dirname $NOME_TARBZ2`
    ARQUIVO_TARBZ2=`basename $NOME_TARBZ2`
    cd $DIRETORIO_TARBZ2
    if [ -e $ARQUIVO_TARBZ2 ];
    then
    NOME_COMPACT_TARBZ2=$(dialog
    --backtitle 'KOMPAKT'
    --title 'Compactar no formato .TAR.BZ2'
    --stdout
    --inputbox 'Digite o nome para o arquivo .TAR.BZ2:'
    10 60
    retorno=$?
    [ $retorno -ne 0 ] && COMPACTAR
    if [ ${NOME_COMPACT_TARBZ2#*.} = "tar.bz2"
    ];
    then
    LOCAL=$(dialog

```



```

dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'Voltar' \
--msgbox 'Extensão invalida.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi

else

dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'Voltar' \
--msgbox 'Arquivo inexistente ou digitado incorretamente.' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi
}

# Final da Function que compacta arquivos no formato
.TAR.BZ2

# -----
-#

# Inicio da Function que compacta arquivos no formato .RAR

COMPACT_RAR() {
clear

NOME_RAR=$(dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo ou diretório que deseja
compactar:\
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

DIRETORIO_RAR=`dirname $NOME_RAR`
ARQUIVO_RAR=`basename $NOME_RAR`

cd $DIRETORIO_RAR
if [ -e $ARQUIVO_RAR ];
then

NOME_COMPACT_RAR=$(dialog \
--backtitle 'KOMPAKT' \
--title 'Compactar no formato .RAR' \
--stdout \
--inputbox 'Digite o nome para o arquivo .RAR:' \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

if [ ${NOME_COMPACT_RAR#*.} = "rar" ];
then

LOCAL=$(dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && COMPACTAR

if [ -e $LOCAL ] && [ -d $LOCAL ];
then
#Variavel que verifica qual usuario esta
logado.
USER=`whoami`

cd $LOCAL
if [ -e Compactados_Kompakt ]
&& [ -d Compactados_Kompakt ];
then
cd Compactados_Kompakt

if [ -e
$NOME_COMPACT_RAR ];
then
dialog \
--backtitle \
'KOMPAKT' \
--title 'ATENÇÃO!!' \
--yesno "\nJa existe
um arquivo com este nome.
mesmo assim?" \
\nDeseja salva-lo
10 60

retorno=$?
[ $retorno -ne 0 ]
&& COMPACTAR

cd $DIRETORIO_RAR
rar a
$NOME_COMPACT_RAR $ARQUIVO_RAR >>
.kompakt.log
mv
$NOME_COMPACT_RAR $LOCAL/Compactados_Kompakt
chown
$USER.$USER $LOCAL/Compactados_Kompakt -R
cd ..

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo compactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ]
&& COMPACTAR

else

cd
$DIRETORIO_RAR
rar a $NOME_COMPACT_RAR
$ARQUIVO_RAR >> .kompakt.log
mv $NOME_COMPACT_RAR
$LOCAL/Compactados_Kompakt
chown $USER.$USER
$LOCAL/Compactados_Kompakt -R
cd ..

```

```

[ $retorno -eq 255 ] && COMPACTAR
    dialog
    \
    --backtitle
'KOMPAKT' \
    \
    --title 'PARABÉNS'
compactado com sucesso!' \
    \
    --msgbox 'Arquivo
    \
    10 60
    \
    retorno=$?
[ $retorno -eq 0 ]
&& COMPACTAR
[ $retorno -eq 255 ]
fi
else
    mkdir Compactados_Kompakt
    cd Compactados_Kompakt
    cd $DIRETORIO_RAR
    rar a $NOME_COMPACT_RAR
    $ARQUIVO_RAR >> .kompakt.log
    mv $NOME_COMPACT_RAR
    $LOCAL/Compactados_Kompakt
    chown $USER.$USER
    $LOCAL/Compactados_Kompakt -R
    cd ..
    dialog
    \
    --backtitle 'KOMPAKT'
    \
    --title 'PARABÉNS'
    \
    --msgbox 'Arquivo compactado com
sucesso!' \
    \
    10 60
    \
    retorno=$?
[ $retorno -eq 0 ] &&
COMPACTAR
[ $retorno -eq 255 ] &&
COMPACTAR
fi
else
    dialog
    \
    --backtitle 'KOMPAKT'
    \
    --title 'ATENÇÃO!'
    \
    --ok-label 'Voltar'
    \
    --msgbox 'Local inexistente ou digitado
incorretamente.' \
    \
    10 60
    \
    retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR
fi
else
    dialog
    \
    --backtitle 'KOMPAKT'
    \
    --title 'ATENÇÃO!'
    \
    --ok-label 'Voltar'
    \
    --msgbox 'Extensão inválida.'
    \
    10 60
    \
    retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
fi
fi
}
# Final da Function que compacta arquivos no formato .RAR
# -----
# Inicio da Function que compacta arquivos no formato .ZIP
COMPACT_ZIP() {
clear
    NOME_ZIP=$(dialog
    \
    --backtitle 'KOMPAKT'
    \
    --title 'Escolha o arquivo ou diretório que deseja
compactar:\
    \
    --stdout
    \
    --fselect /home/
    \
    10 60
    )
    \
    retorno=$?
[ $retorno -ne 0 ] && COMPACTAR
DIRETORIO_ZIP=`dirname $NOME_ZIP`
ARQUIVO_ZIP=`basename $NOME_ZIP`
cd $DIRETORIO_ZIP
if [ -e $ARQUIVO_ZIP ];
then
    NOME_COMPACT_ZIP=$(dialog
    \
    --backtitle 'KOMPAKT'
    \
    --title 'Compactar no formato .ZIP'
    \
    --stdout
    \
    --inputbox 'Digite o nome para o arquivo .ZIP:'
    \
    10 60
    )
    \
    retorno=$?
[ $retorno -ne 0 ] && COMPACTAR
    if [ ${NOME_COMPACT_ZIP#*.} = "zip" ];
    then
        LOCAL=$(dialog
        \
        --backtitle 'KOMPAKT'
        \
        --title 'Escolha o local onde deseja salvar o arquivo:'
        \
        --stdout
        \
        --fselect /home/
        \
        10 60
        )
        \
        retorno=$?
[ $retorno -ne 0 ] && COMPACTAR
        if [ -e $LOCAL ] && [ -d $LOCAL ];
        then

```



```

retorno=$?
[ $retorno -eq 0 ] && COMPACTAR
[ $retorno -eq 255 ] && COMPACTAR

fi
}

# Final da Function que compacta arquivos no formato .ZIP

# -----
-#
# -----
#
# Início das Functions que tem a finalidade de Descompactar as
extensões #
# .TAR.GZ, .TAR, .BZ2, .TAR.BZ2, .RAR, .ZIP
#
# -----
-#

# Descompactando arquivos no formato .TAR.GZ

DESCOMPACT_TARGZ() {

ARQUIVO_TARGZ=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo .TAR.GZ que deseja descompactar:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

#Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_TARGZ=`dirname $ARQUIVO_TARGZ`

#Variavel que filtra o nome do arquivo
NOME_TARGZ=`basename $ARQUIVO_TARGZ`

cd $DIRETORIO_TARGZ
if [ -e $NOME_TARGZ ] && [ ${NOME_TARGZ##*.} =
"tar.gz" ];
then

#Variavel que identifica o usuario que esta logado no sistema
USER=`whoami`

SALVAR_TARGZ=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo
descompactado:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

if [ -e $$SALVAR_TARGZ ] && [ -d
$$SALVAR_TARGZ ];
then
cd $$SALVAR_TARGZ

if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ];
then
cd Descompactados_Kompakt
tar -vzxf $ARQUIVO_TARGZ >>

.kompakt.log

cd ..

```

```

chown $USER.$USER
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] &&
DESCOMPACTAR

else
mkdir Descompactados_Kompakt
cd Descompactados_Kompakt
tar -vzxf $ARQUIVO_TARGZ >>

.kompakt.log

cd ..
chown $USER.$USER
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR

```

```

[ $retorno -eq 255 ] && DESCOMPACTAR
fi
}

# Final da function que descompacta os arquivos no formato
.TAR.GZ
# -----#

# Inicio da Function que descompacta os arquivos no formato
.TAR

DESCOMPACT_TAR() {

ARQUIVO_TAR=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo .TAR que deseja descompactar:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

#Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_TAR=`dirname $ARQUIVO_TAR`

#Variavel que filtra o nome do arquivo
NOME_TAR=`basename $ARQUIVO_TAR`

cd $DIRETORIO_TAR
if [ -e $NOME_TAR ] && [ ${NOME_TAR#*.} = "tar" ] ;
then

#Variavel que identifica o usuario que esta logado no sistema
USER1=`whoami`

SALVAR_TAR=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo
descompactado:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

if [ -e $SALVAR_TAR ] && [ -d $SALVAR_TAR ]
;
then
cd $SALVAR_TAR

if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ] ;
then
cd Descompactados_Kompakt
tar -xvf $ARQUIVO_TAR >>

.kompakt.log

cd ..
chown $USER1.$USER1
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

```

```

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] &&
DESCOMPACTAR

else
mkdir Descompactados_Kompakt
cd Descompactados_Kompakt
tar -xvf $ARQUIVO_TAR >>

.kompakt.log

cd ..
chown $USER1.$USER1
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi
}

# Final da Function que descompacta os arquivos no formato
.TAR
# -----#

#inicio da Function que descompacta os arquivos no formao
.TAR.XZ

DESCOMPACT_TARXZ() {

```

```

ARQUIVO_TARXZ=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo .TAR.XZ que deseja descompactar:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

#Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_TARXZ=`dirname $ARQUIVO_TARXZ`

#Variavel que filtra o nome do arquivo
NOME_TARXZ=`basename $ARQUIVO_TARXZ`

cd $DIRETORIO_TARXZ
if [ -e $NOME_TARXZ ] && [ ${NOME_TARXZ##*.} =
"tar.xz" ];
then

#Variavel que identifica o usuario que esta logado no sistema
USER=`whoami`

SALVAR_TARXZ=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo
descompactado:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

    if [ -e $$SALVAR_TARXZ ] && [ -d
$$SALVAR_TARXZ ];
    then
        cd $$SALVAR_TARXZ

        if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ];
        then
            cd Descompactados_Kompakt
            tar -Jxf $ARQUIVO_TARXZ >>
.kompakt.log

            cd ..
            chown $USER.$USER
            Descompactados_Kompakt -R
            chmod 777 Descompactados_Kompakt -R

            dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

            retorno=$?
            [ $retorno -eq 0 ] && DESCOMPACTAR
            [ $retorno -eq 255 ] &&
DESCOMPACTAR

            else
                mkdir Descompactados_Kompakt
                cd Descompactados_Kompakt
                tar -Jxf $ARQUIVO_TARXZ >>
.kompakt.log

                cd ..
                chown $USER.$USER
                Descompactados_Kompakt -R
                chmod 777 Descompactados_Kompakt -R

```

```

dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n' \
10 60

retorno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi
}

# Final da Function que descompacta os arquivos no formato
.TAR.XZ
# -----#

# Inicio da Function que descompacta os arquivos no formato
.BZ2

DESCOMPACT_BZ2() {
ARQUIVO_BZ2=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo .BZ2 que deseja descompactar:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

#Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_BZ2=`dirname $ARQUIVO_BZ2`

```

```

#Variavel que filtra o nome do arquivo
NOME_BZ2=`basename $ARQUIVO_BZ2`

cd $DIRETORIO_BZ2
if [ -e $NOME_BZ2 ];
then

#Variavel que identifica o usuario que esta logado no sistema
USER2=`whoami`

SALVAR_BZ2=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo \
descompactado:' \
--stdout \
--fselect /home/ \
10 60 )

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

    if [ -e $SALVAR_BZ2 ] && [ -d $SALVAR_BZ2 ];
    then
        cd $SALVAR_BZ2

        if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ];
        then
            cd Descompactados_Kompakt
            bzip2 -d $ARQUIVO_BZ2 >>
.kompakt.log

            cd ..
            chown $USER2.$USER2
            Descompactados_Kompakt -R
            chmod 777 Descompactados_Kompakt -R

            dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

            retorno=$?
            [ $retorno -eq 0 ] && DESCOMPACTAR
            [ $retorno -eq 255 ] &&
DESCOMPACTAR

            else
                mkdir Descompactados_Kompakt
                cd Descompactados_Kompakt
                bzip2 -d $ARQUIVO_BZ2 >>
.kompakt.log

                cd ..
                chown $USER2.$USER2
                Descompactados_Kompakt -R
                chmod 777 Descompactados_Kompakt -R

                dialog \
--backtitle 'KOMPAKT' \
--title 'PARABÉNS' \
--msgbox 'Arquivo descompactado com
sucesso!' \
10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] && DESCOMPACTAR

                fi
            else
                dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n'
10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] && DESCOMPACTAR

                fi
            else
                dialog \
--backtitle 'KOMPAKT' \
--title 'ATENÇÃO!' \
--ok-label 'VOLTAR' \
--msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n'
10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] && DESCOMPACTAR

                fi
            }

# Final da Function que descompacta os arquivos no formato
.BZ2

# -----#

# Inicio da Function que descompacta os arquivos no formato
.TAR.BZ2

DESCOMPACT_TARBZ2() {
    ARQUIVO_TARBZ2=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o arquivo .TAR.BZ2 que deseja descompactar:' \
--stdout \
--fselect /home/ \
10 60 )

    retorno=$?
    [ $retorno -ne 0 ] && DESCOMPACTAR

    #Variavel que filtra o diretório onde o arquivo se encontra
    DIRETORIO_TARBZ2=`dirname $ARQUIVO_TARBZ2`

    #Variavel que filtra o nome do arquivo
    NOME_TARBZ2=`basename $ARQUIVO_TARBZ2`

    cd $DIRETORIO_TARBZ2
    if [ -e $NOME_TARBZ2 ] && [ ${NOME_TARBZ2#*.} =
"tar.bz2" ];
    then

        #Variavel que identifica o usuario que esta logado no sistema
        USER3=`whoami`

        SALVAR_TARBZ2=$( dialog \
--backtitle 'KOMPAKT' \
--title 'Escolha o local onde deseja salvar o arquivo \
descompactado:' \
--stdout \

```

```

--fselect /home/
10 60

retorno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

    if [ -e $$SALVAR_TARBZ2 ] && [ -d
$$SALVAR_TARBZ2 ];
    then
        cd $$SALVAR_TARBZ2

            if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ];
            then
                cd Descompactados_Kompakt
                tar -jxvf $ARQUIVO_TARBZ2 >>
.kompakt.log
                cd ..
                chown $USER3.$USER3
Descompactados_Kompakt -R
                chmod 777 Descompactados_Kompakt -R

                dialog
                    \
                    --backtitle 'KOMPAKT'
                    \
                    --title 'PARABÉNS'
                    \
                    --msgbox 'Arquivo descompactado com
sucesso!'
                    \
                    10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] &&
DESCOMPACTAR

                else
                mkdir Descompactados_Kompakt
                cd Descompactados_Kompakt
                tar -jxvf $ARQUIVO_TARBZ2 >>
.kompakt.log
                cd ..
                chown $USER3.$USER3
Descompactados_Kompakt -R
                chmod 777 Descompactados_Kompakt -R

                dialog
                    \
                    --backtitle 'KOMPAKT'
                    \
                    --title 'PARABÉNS'
                    \
                    --msgbox 'Arquivo descompactado com
sucesso!'
                    \
                    10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] && DESCOMPACTAR

                fi

                else
                dialog
                    \
                    --backtitle 'KOMPAKT'
                    \
                    --title 'ATENÇÃO!'
                    \
                    --ok-label 'VOLTAR'
                    \
                    --msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n'
                    \
                    10 60

                retorno=$?
                [ $retorno -eq 0 ] && DESCOMPACTAR
                [ $retorno -eq 255 ] && DESCOMPACTAR

                fi

                # Final da Function que descompacta os arquivos no formato
.TAR.BZ2

                # -----#

                # Inicio da Function que descompacta os arquivos no formato
.RAR

                DESCOMPACT_RAR() {

                ARQUIVO_RAR=$( dialog
                    \
                    --backtitle 'KOMPAKT'
                    \
                    --title 'Escolha o arquivo .RAR que deseja descompactar:'
                    \
                    --stdout
                    \
                    --fselect /home/
                    \
                    10 60

                retorno=$?
                [ $retorno -ne 0 ] && DESCOMPACTAR

                #Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_RAR=`dirname $ARQUIVO_RAR`

                #Variavel que filtra o nome do arquivo
NOME_RAR=`basename $ARQUIVO_RAR`

                cd $DIRETORIO_RAR
                if [ -e $NOME_RAR ] && [ ${NOME_RAR#*.} = "rar" ];
                then

                #Variavel que identifica o usuario que esta logado no sistema
USER4=`whoami`

                SALVAR_RAR=$( dialog
                    \
                    --backtitle 'KOMPAKT'
                    \
                    --title 'Escolha o local onde deseja salvar o arquivo
descompactado:'
                    \
                    --stdout
                    \
                    --fselect /home/
                    \
                    10 60

                retorno=$?
                [ $retorno -ne 0 ] && DESCOMPACTAR

                if [ -e $$SALVAR_RAR ] && [ -d $$SALVAR_RAR ]
;
                then
                cd $$SALVAR_RAR

                if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ];
                then
                cd Descompactados_Kompakt

```

```

unrar x $ARQUIVO_RAR >>

.kompakt.log
cd ..
chown $USER4.$USER4
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Arquivo descompactado com
sucesso!'
10 60

returno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

DESCOMPACTAR
else
mkdir Descompactados_Kompakt
cd Descompactados_Kompakt
unrar x $ARQUIVO_RAR >>

.kompakt.log
cd ..
chown $USER4.$USER4
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Arquivo descompactado com
sucesso!'
10 60

returno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'VOLTAR'
--msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n'
10 60

returno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

fi

else
dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'VOLTAR'
--msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n'
10 60

returno=$?
[ $retorno -eq 0 ] && DESCOMPACTAR
[ $retorno -eq 255 ] && DESCOMPACTAR

DESCOMPACT_ZIP() {
ARQUIVO_ZIP=$( dialog
--backtitle 'KOMPAKT'
--title 'Escolha o arquivo .ZIP que deseja descompactar:'
--stdout
--fselect /home/
10 60 )

returno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

#Variavel que filtra o diretório onde o arquivo se encontra
DIRETORIO_ZIP=`dirname $ARQUIVO_ZIP`

#Variavel que filtra o nome do arquivo
NOME_ZIP=`basename $ARQUIVO_ZIP`

cd $DIRETORIO_ZIP
if [ -e $NOME_ZIP ] && [ ${NOME_ZIP#*.} = "zip" ] ;
then

#Variavel que identifica o usuario que esta logado no sistema
USER5=`whoami`

SALVAR_ZIP=$( dialog
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar o arquivo
descompactado:'
--stdout
--fselect /home/
10 60 )

returno=$?
[ $retorno -ne 0 ] && DESCOMPACTAR

if [ -e $SALVAR_ZIP ] && [ -d $SALVAR_ZIP ] ;
then
cd $SALVAR_ZIP

if [ -e Descompactados_Kompakt ] && [ -
d Descompactados_Kompakt ] ;
then
cd Descompactados_Kompakt
unzip $ARQUIVO_ZIP >> .kompakt.log
cd ..
chown $USER5.$USER5
Descompactados_Kompakt -R
chmod 777 Descompactados_Kompakt -R

dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Arquivo descompactado com
sucesso!'
10 60

```

```

        retorno=$?
        [ $retorno -eq 0 ] && DESCOMPACTAR
DESCOMPACTAR [ $retorno -eq 255 ] &&

        else
        mkdir Descompactados_Kompakt
        cd Descompactados_Kompakt
        unzip $ARQUIVO_ZIP >> .kompakt.log
        cd ..
        chown $USER5.$USER5
Descompactados_Kompakt -R
        chmod 777 Descompactados_Kompakt -R

        dialog
        --backtitle 'KOMPAKT'
        --title 'PARABÉNS'
        --msgbox 'Arquivo descompactado com
sucesso!'
        10 60

        retorno=$?
        [ $retorno -eq 0 ] && DESCOMPACTAR
        [ $retorno -eq 255 ] && DESCOMPACTAR
        fi

        else
        dialog
        --backtitle 'KOMPAKT'
        --title 'ATENÇÃO!'
        --ok-label 'VOLTAR'
        --msgbox '\nLocal inexistente ou digitado
incorretamente.
\nPressione [VOLTAR] para retornar ao Menu.\n'
        10 60

        retorno=$?
        [ $retorno -eq 0 ] && DESCOMPACTAR
        [ $retorno -eq 255 ] && DESCOMPACTAR
        fi

    else
        dialog
        --backtitle 'KOMPAKT'
        --title 'ATENÇÃO!'
        --ok-label 'VOLTAR'
        --msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n'
        10 60

        retorno=$?
        [ $retorno -eq 0 ] && DESCOMPACTAR
        [ $retorno -eq 255 ] && DESCOMPACTAR
        fi
    }

# Final da Function que descompacta os arquivos no formato
.ZIP

# -----#
#####
#####
# Inicio do módulo que converte imagens.
#

        #####
        #####
        # -----#

        # Inicio da Function que converte imagens

        CONVERT_IMAGEM() {
        NOME_IMAGEM=$(dialog
        \
        --backtitle 'KOMPAKT'
        --title 'Escolha a imagem que deseja converter:'
        \
        --stdout
        --fselect /home/
        10 60
        )

        retorno=$?
        [ $retorno -ne 0 ] && MENU

        DIRETORIO_IMAGEM=`dirname $NOME_IMAGEM`
        ARQUIVO_IMAGEM=`basename $NOME_IMAGEM`

        cd $DIRETORIO_IMAGEM

        if [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "jpg" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "jpeg" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "png" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "bmp" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "gif" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "pcx" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "tga" ]; then
        IMAGEM_CONVERTER

        elif [ -e $ARQUIVO_IMAGEM ] && [
        ${ARQUIVO_IMAGEM##*.} = "tiff" ]; then
        IMAGEM_CONVERTER

        else
        dialog
        --backtitle 'KOMPAKT'
        --title 'ATENÇÃO!'
        --ok-label 'VOLTAR'
        --msgbox '\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n'
        10 60

        fi

        retorno=$?
        [ $retorno -eq 0 ] && MENU
        [ $retorno -eq 255 ] && MENU
    }

```

```

IMAGEM_CONVERTER() {
    NOME_NOVA_IMAGEM=$(dialog
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--inputbox 'Digite um nome para a nova imagem:'
10 60

    retorno=$?
    [ $retorno -ne 0 ] && MENU

    FORMATO_IMAGEM=$(dialog
\
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--radiolist 'Escolha o novo formato:'
10 60 0
jpg " on
png " off
bmp " off
gif " off
pcx " off
tga " off
tiff " off

    retorno=$?
    [ $retorno -ne 0 ] && MENU

    DIMENSAO_IMAGEM=$(dialog
\
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--radiolist 'Escolha o tamanho da imagem:'
\
10 60 0
640x480 'Pequena' off
800x600 'Média' off
1024x768 'Grande' on
1280x1024 'Extra Grande' off

    retorno=$?
    [ $retorno -ne 0 ] && MENU

    QUALIDADE_IMAGEM=$(dialog
\
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--radiolist 'Defina a qualidade:'
10 60 0
15% 'Qualidade Mínima' off
25% 'Qualidade Baixa' off
50% 'Qualidade Média' off
75% 'Qualidade Alta' off
100% 'Qualidade Excelente' on

    retorno=$?
    [ $retorno -ne 0 ] && MENU

    LOCAL=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar a imagem:'
\
--stdout
--fselect /home/
10 60

    retorno=$?
    [ $retorno -ne 0 ] && MENU

    if [ -e $LOCAL ] && [ -d $LOCAL ] ;
    then
        #Variavel que verifica qual usuario esta logado.
        USER=`whoami`
        IMAGEM_CONVERTIDA=$NOME_NOVA_IMA
        GEM.$FORMATO_IMAGEM

        cd $LOCAL
        if [ -e Imagens_Kompakt ] && [ -d
        Imagens_Kompakt ] ;
        then
            cd Imagens_Kompakt

            if [ -e
            $NOME_NOVA_IMAGEM.$FORMATO_IMAGEM ] ;
            then
                dialog
                \
                --backtitle 'KOMPAKT'
                \
                --title 'ATENÇÃO!!'
                \
                --yesno "\nJa existe uma
                imagem com este nome.
                \nDeseja salva-la mesmo
                assim?"
                \
                10 60

                retorno=$?
                [ $retorno -ne 0 ] && MENU

                cd $DIRETORIO_IMAGEM
                convert -resize
                $DIMENSAO_IMAGEM -quality $QUALIDADE_IMAGEM
                $ARQUIVO_IMAGEM $IMAGEM_CONVERTIDA >>
                .kompakt.log

                mv
                $IMAGEM_CONVERTIDA $LOCAL/Imagens_Kompakt
                chown $USER.$USER Imagens_Kompakt -R
                cd ..

                dialog
                \
                --backtitle 'KOMPAKT'
                \
                --title 'PARABÉNS'
                \
                --msgbox 'Imagem convertida com sucesso!'
                \
                10 60

                retorno=$?
                [ $retorno -eq 0 ] && MENU
                [ $retorno -eq 255 ] && MENU

            else

                cd $DIRETORIO_IMAGEM
                convert -resize $DIMENSAO_IMAGEM -quality
                $QUALIDADE_IMAGEM $ARQUIVO_IMAGEM
                $IMAGEM_CONVERTIDA >> .kompakt.log
                mv $IMAGEM_CONVERTIDA
                $LOCAL/Imagens_Kompakt
                chown $USER.$USER Imagens_Kompakt -R
                cd ..

                dialog
                \
                --backtitle 'KOMPAKT'
                \
                --title 'PARABÉNS'
                \
                --msgbox 'Imagem convertida
                com sucesso!'
                \
                10 60

                retorno=$?
            fi
        fi
    fi
}

```

```

[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi

else
    mkdir Imagens_Kompakt
    cd Imagens_Kompakt
    cd $DIRETORIO_IMAGEM
    convert -resize $DIMENSAO_IMAGEM -quality
$QUALIDADE_IMAGEM $ARQUIVO_IMAGEM
$IAGEMEM_CONVERTIDA >> .kompakt.log
    mv $IAGEMEM_CONVERTIDA
$LOCAL/Imagens_Kompakt
    chown $USER.$USER Imagens_Kompakt -R
    cd ..

    dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Imagem convertida com sucesso!'
10 60

    retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi

else
    dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'Voltar'
--msgbox 'Local inexistente ou digitado incorretamente.'
10 60

    retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi
}

# Final da Function que converte imagens

# -----#

# Inicio da Function que converte Áudio

CONVERT_AUDIO() {

NOME_AUDIO=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o arquivo de áudio que deseja converter para
MP3:'
--stdout
--fselect /home/
10 60

)

retorno=$?
[ $retorno -ne 0 ] && MENU

DIRETORIO_AUDIO=`dirname $NOME_AUDIO`
ARQUIVO_AUDIO=`basename $NOME_AUDIO`

cd $DIRETORIO_AUDIO

if [ -e $ARQUIVO_AUDIO ] && [ ${ARQUIVO_AUDIO#*.}
= "flac" ]; then
    AUDIO_CONVERTER_OGG_AAC

elif [ -e $ARQUIVO_AUDIO ] && [
${ARQUIVO_AUDIO#*.} = "aac" ]; then
    AUDIO_CONVERTER_OGG_AAC

elif [ -e $ARQUIVO_AUDIO ] && [
${ARQUIVO_AUDIO#*.} = "ogg" ]; then
    AUDIO_CONVERTER_OGG_AAC

elif [ -e $ARQUIVO_AUDIO ] && [
${ARQUIVO_AUDIO#*.} = "wav" ]; then
    AUDIO_CONVERTER_WAV

else

    dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'VOLTAR'
--msgbox '\nArquivo inexistente, digitado incorretamente ou não
suportado pelo software.
Pressione [VOLTAR] para retornar ao Menu.\n'
10 60

fi

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

}

AUDIO_CONVERTER_WAV() {

NOME_NOVO_AUDIO=$(dialog
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--inputbox 'Digite um nome para o novo Áudio:'
10 60

)

retorno=$?
[ $retorno -ne 0 ] && MENU

#FORMATO_AUDIO=$(dialog
\
#--backtitle 'KOMPAKT'
#--title 'KOMPAKT'
#--stdout
#--radiolist 'Escolha o formato:'
#10 60 0
#flac 'Conversão de FLAC para MP3' on
\
#aac 'Conversão de ACC para MP3' off
\
#ogg 'Conversão de OGG para MP3' off
\
#wav 'Conversão de WAV para MP3' off
)

#retorno=$?
#[ $retorno -ne 0 ] && MENU

LOCAL=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar o arquivo de áudio:'
\
--stdout
--fselect /home/
10 60

)

retorno=$?
[ $retorno -ne 0 ] && MENU

if [ -e $LOCAL ] && [ -d $LOCAL ];
then

```

```

#Variavel que verifica qual usuario esta logado.
USER=`whoami`
AUDIO_CONVERTIDO=$NOME_NOVO_AUDIO.

mp3
cd $LOCAL
  if [ -e Áudio_Kompakt ] && [ -d
Áudio_Kompakt ] ;
  then
    cd Áudio_Kompakt

    if [ -e
$NOME_NOVO_AUDIO.mp3 ] ;
    then
      dialog
        \
        --backtitle 'KOMPAKT'
        \
        --title 'ATENÇÃO!!'
        \
        --yesno "\nJa existe um arquivo
de áudio com este nome.
assim?"
        \
        10 60

      retorno=$?
      [ $retorno -ne 0 ] && MENU

      cd $DIRETORIO_AUDIO
      lame -S -h
$ARQUIVO_AUDIO $AUDIO_CONVERTIDO
      mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
      chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
      cd ..

      dialog
        \
        --backtitle 'KOMPAKT'
        \
        --title 'PARABÉNS'
        \
        --msgbox 'Áudio convertido com sucesso!'
        \
        10 60

      retorno=$?
      [ $retorno -eq 0 ] && MENU
      [ $retorno -eq 255 ] && MENU

    else
      cd $DIRETORIO_AUDIO
      lame -S -h $ARQUIVO_AUDIO
$AUDIO_CONVERTIDO
      mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
      chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
      cd ..

      dialog
        \
        --backtitle 'KOMPAKT'
        \
        --title 'PARABÉNS'
        \
        --msgbox 'Áudio convertido
com sucesso!'
        \
        10 60

      retorno=$?
      [ $retorno -eq 0 ] && MENU
      [ $retorno -eq 255 ] && MENU

    fi
  fi
}

# Início da conversão a partir do formato flac
AUDIO_CONVERTER_FLAC() {
  NOME_NOVO_AUDIO=$(dialog
  \
  --backtitle 'KOMPAKT'
  \
  --title 'KOMPAKT'
  \
  --stdout
  \
  --inputbox 'Digite um nome para o novo Áudio:'
  \
  10 60
  )

  retorno=$?
  [ $retorno -ne 0 ] && MENU

  LOCAL=$(dialog
  \
  --backtitle 'KOMPAKT'
  \
  --title 'Escolha o local onde deseja salvar o arquivo de Áudio:'
  \
  --stdout
  \
  --fselect /home/
  \
  10 60
  )

  retorno=$?
  [ $retorno -ne 0 ] && MENU

  if [ -e $LOCAL ] && [ -d $LOCAL ] ;
  then

    #Variavel que verifica qual usuario esta logado.
    USER=`whoami`
    AUDIO_CONVERTIDO=$NOME_NOVO_AUDIO.
    mp3

```

```

cd $LOCAL
if [ -e Áudio_Kompakt ] && [ -d
Áudio_Kompakt ];
then
cd Áudio_Kompakt

if [ -e
$NOME_NOVO_AUDIO.mp3 ];
then
dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!!'
\
--yesno "\nJa existe um arquivo
de Áudio com este nome.
\nDeseja salva-lo mesmo
assim?"
\
10 60

retorno=$?
[ $retorno -ne 0 ] && MENU

cd $DIRETORIO_AUDIO
flac -s -o
$NOME_NOVO_AUDIO.wav $ARQUIVO_AUDIO
lame -S -h
$NOME_NOVO_AUDIO.wav $AUDIO_CONVERTIDO
rm
$NOME_NOVO_AUDIO.wav > /dev/null
mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
cd ..

dialog
\
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Áudio convertido com sucesso!'
10 60

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

else
cd $DIRETORIO_AUDIO
flac -s -o $NOME_NOVO_AUDIO.wav
$ARQUIVO_AUDIO
lame -S -h $NOME_NOVO_AUDIO.wav
$AUDIO_CONVERTIDO
rm $NOME_NOVO_AUDIO.wav > /dev/null
mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
cd ..

dialog
\
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Áudio convertido
com sucesso!'
10 60

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi

fi

else
mkdir Áudio_Kompakt
cd Áudio_Kompakt
cd $DIRETORIO_AUDIO
flac -s -o $NOME_NOVO_AUDIO.wav
$ARQUIVO_AUDIO
lame -S -h $NOME_NOVO_AUDIO.wav
$AUDIO_CONVERTIDO
rm $NOME_NOVO_AUDIO.wav > /dev/null
mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
chown $USER.$USER $LOCAL/Áudio_Kompakt -R
cd ..

dialog
\
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Áudio convertido com sucesso!'
10 60

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi

else
dialog
\
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'Voltar'
--msgbox 'Local inexistente ou digitado incorretamente.'
10 60

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

fi

}

AUDIO_CONVERTER_OGG_AAC() {
NOME_NOVO_AUDIO=$(dialog
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--inputbox 'Digite um nome para o novo Áudio:'
10 60
)

retorno=$?
[ $retorno -ne 0 ] && MENU

LOCAL=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar o arquivo de Áudio:'
\
--stdout
--fselect /home/
10 60
)

retorno=$?
[ $retorno -ne 0 ] && MENU

if [ -e $LOCAL ] && [ -d $LOCAL ];
then

#Variavel que verifica qual usuario esta logado.
USER=`whoami`
AUDIO_CONVERTIDO=$NOME_NOVO_AUDIO.
mp3

```

```

cd $LOCAL
if [ -e Áudio_Kompakt ] && [ -d
Áudio_Kompakt ];
then
cd Áudio_Kompakt
else
mkdir Áudio_Kompakt
cd Áudio_Kompakt
cd $DIRETORIO_AUDIO
ffmpeg -i $ARQUIVO_AUDIO
$NOME_NOVO_AUDIO.wav >/dev/null 2>&1
lame -S -h $NOME_NOVO_AUDIO.wav
$AUDIO_CONVERTIDO
rm $NOME_NOVO_AUDIO.wav >/dev/null
mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
cd ..
dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Áudio convertido com sucesso!'
10 60
retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU
else
cd $DIRETORIO_AUDIO
ffmpeg -i $ARQUIVO_AUDIO
$NOME_NOVO_AUDIO.wav >/dev/null 2>&1
lame -S -h $NOME_NOVO_AUDIO.wav
$AUDIO_CONVERTIDO
rm $NOME_NOVO_AUDIO.wav >/dev/null
mv $AUDIO_CONVERTIDO
$LOCAL/Áudio_Kompakt
chown $USER.$USER
$LOCAL/Áudio_Kompakt -R
cd ..
dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Áudio convertido
com sucesso!'
10 60
retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU
fi
}
# Final da Function que converte Áudio
# -----#
# Inicio da Function que converte arquivos de Vídeo
CONVERT_VIDEO() {
NOME_VIDEO=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o vídeo que deseja converter:'
--stdout
--fselect /home/
10 60
)
retorno=$?
[ $retorno -ne 0 ] && MENU
DIRETORIO_VIDEO=`dirname $NOME_VIDEO`
ARQUIVO_VIDEO=`basename $NOME_VIDEO`
cd $DIRETORIO_VIDEO
if [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##*.}
= "avi" ]; then
VIDEO_CONVERTER

```

```

elif [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##.}
= "wmv" ]; then
VIDEO_CONVERTER

elif [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##.}
= "mkv" ]; then
VIDEO_CONVERTER

elif [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##.}
= "mpg" ]; then
VIDEO_CONVERTER

elif [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##.}
= "vob" ]; then
VIDEO_CONVERTER

elif [ -e $ARQUIVO_VIDEO ] && [ ${ARQUIVO_VIDEO##.}
= "flv" ]; then
VIDEO_CONVERTER

else

dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!'
--ok-label 'VOLTAR'
--msgbox "\nArquivo inexistente ou digitado incorretamente.
Pressione [VOLTAR] para retornar ao Menu.\n"
\
10 60

fi

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

}

VIDEO_CONVERTER() {

NOME_NOVO_VIDEO=$(dialog
--backtitle 'KOMPAKT'
--title 'KOMPAKT!!'
--stdout
--inputbox 'Digite o nome para o novo vídeo:'
\
10 60
)

retorno=$?
[ $retorno -ne 0 ] && MENU

FORMATO_VIDEO=$(dialog
\
--backtitle 'KOMPAKT'
--title 'KOMPAKT'
--stdout
--radiolist 'Escolha o novo formato:'
\
10 60 0
avi " on

wmv " off

mkv " off
mpg " off
vob " off
flv " off
)

retorno=$?
[ $retorno -ne 0 ] && MENU

LOCAL=$(dialog
--backtitle 'KOMPAKT'
--title 'Escolha o local onde deseja salvar o vídeo:'
\
--stdout
\

```

```

--fselect /home/
10 60
)

retorno=$?
[ $retorno -ne 0 ] && MENU

if [ -e $LOCAL ] && [ -d $LOCAL ];
then

#Variavel que verifica qual usuario esta logado.
USER=`whoami`
VIDEO_CONVERTIDO=$NOME_NOVO_VIDEO.
$FORMATO_VIDEO

cd $LOCAL
if [ -e Vídeos_Kompakt ] && [ -d
Vídeos_Kompakt ];
then
cd Vídeos_Kompakt

if [ -e
$NOME_NOVO_VIDEO.$FORMATO_VIDEO ];
then
dialog
--backtitle 'KOMPAKT'
--title 'ATENÇÃO!!'
\
--yesno "\nJa existe um vídeo
com este nome.
\nDeseja salva-lo mesmo
assim?"
\
10 60

retorno=$?
[ $retorno -ne 0 ] && MENU

cd $DIRETORIO_VIDEO
ffmpeg -i $NOME_VIDEO
$VIDEO_CONVERTIDO >/dev/null 2>&1
mv $VIDEO_CONVERTIDO
$LOCAL/Vídeos_Kompakt
chown $USER.$USER $LOCAL/Vídeos_Kompakt -
R
cd ..
dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
--msgbox 'Vídeo convertido com sucesso!'
\
10 60

retorno=$?
[ $retorno -eq 0 ] && MENU
[ $retorno -eq 255 ] && MENU

else

cd $DIRETORIO_VIDEO
ffmpeg -i $NOME_VIDEO
$VIDEO_CONVERTIDO >/dev/null 2>&1
mv $VIDEO_CONVERTIDO
$LOCAL/Vídeos_Kompakt
chown $USER.$USER
$LOCAL/Vídeos_Kompakt -R
cd ..

dialog
--backtitle 'KOMPAKT'
--title 'PARABÉNS'
\
\

```

```

                                --msgbox 'Vídeo convertido
com sucesso!'                \
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && MENU
                                [ $retorno -eq 255 ] && MENU

                                fi

                                else
                                mkdir Vídeos_Kompakt
                                cd Vídeos_Kompakt
                                cd $DIRETORIO_VIDEO
                                ffmpeg -i $NOME_VIDEO $VIDEO_CONVERTIDO
>/dev/null 2>&1
                                mv $VIDEO_CONVERTIDO
                                $LOCAL/Vídeos_Kompakt
                                chown $USER.$USER $LOCAL/Vídeos_Kompakt -R
                                cd ..

                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'PARABÉNS'
                                --msgbox 'Vídeo convertido com sucesso!'
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && MENU
                                [ $retorno -eq 255 ] && MENU

                                fi

                                else
                                dialog
                                --backtitle 'KOMPAKT'
                                --title 'ATENÇÃO!'
                                --ok-label 'Voltar'
                                --msgbox 'Local inexistente ou digitado incorretamente.'
\
                                10 60

                                retorno=$?
                                [ $retorno -eq 0 ] && MENU
                                [ $retorno -eq 255 ] && MENU

                                fi
}

# Final da Function que converte arquivos de Vídeo

# -----#
#####
#####
# FIM #
#####
#####
# -----#
MENU

```