

## **Relatorio do Trabalho Pratico**

### **Versao Final**

Nome: Juliano Leal Goncalves

Nome: Cristiano Medeiros Dalbem

Data: 29/06/09

#### Relatório

O desenvolvimento geral do trabalho, dentro do que foi proposto inicialmente pela dupla, pode ser dividido em 6 grandes grupos: audio, video, imagens, texto, gerenciamento de arquivos e interface. Apos um certo tempo de desenvolvimento, o grupo optou por deixar de fora alguns desses itens e se focar em outros, caso contrario nao se teria nada completo para apresentar em tempo habil.

Para a interface grafica do programa, foi feita uma pesquisa aprofundada de como eh o funcionamento da biblioteca wincon.h, componente da windows.h responsavel por todas as operacoes diretas com console, atraves do site [www.msdn.com](http://www.msdn.com). Mais tarde, quando houve a possibilidade de generalizar boa parte do codigo, o grupo se viu obrigado a migrar para a utilizacao de classes de C++, e mais uma vez um bom tempo foi utilizado aprendendo alguns pontos criticos para o desenvolvimento do que o grupo queria, como por exemplo ponteiros para funcoes (utilizados nas funcoes onClick de alguns componentes graficos como botoes e menus).

Buscando uma alta modularidade e customizacao, houve uma tentativa de padronizacao dos componentes de forma a se assemelharem o maximo possivel ao que eh utilizado na vida real em sistemas operacionais modernos. O grupo fez implementacoes de janelas, menus, ~~drop downs~~ dropdowns, botoes, listboxes, scrollbars e progressbars. Alguns dos componentes que se pretendia implementar e nao foram criados sao por exemplo checkboxes, radiobuttons, sliders e edit boxes. Usando o sistema de heranca de classes, foi produzida uma estrutura final onde por exemplo uma janela poderia ter os seus proprios componentes internos, que sao tratados de forma individualizada.

A principal ferramenta na criacao desses componentes foram funcoes proprias criadas pelo grupo para desenho em um bloco de memoria. Esses blocos de memoria, por sua vez, eram associados a novos screenbuffers atraves de chamadas as funcoes CreateConsoleScreenBuffer e SetConsoleScreenBufferInfo. Dessa forma, cada componente era teoricamente independente do restante e podia ser movido/alterado/

deletado sem que o conteúdo ao seu redor sofresse modificações, facilitando em muito a montagem do layout do programa quando a parte de codificação estivesse concluída. Para essa parte do software, foi utilizada também com pequena ênfase a biblioteca `conio2.h`, que nada mais é que uma versão simplificada de um conjunto de funções da `wincon.h`, geralmente aplicadas ao standard output buffer.

Com relação ao sistema de arquivos, o que foi implementado foi uma simplificação do descrito acima: ao invés de utilizar todo o conteúdo do disco, o grupo se concentrou em tratar apenas o diretório que recebia as cópias dos arquivos de CD, e das funcionalidades propostas, apenas a navegação e seleção do diretório destino foi implementada. A cópia de CD, por sua vez, foi implementada da seguinte maneira: o usuário seleciona no menu principal "copiar CD", e então é apresentada uma lista com todos os drives de CD encontrados na máquina, para que seja feita a escolha de qual deles o conteúdo será copiado. Logo a seguir, é apresentada uma interface para que o usuário monte o caminho ao qual serão copiados os arquivos, primeiramente selecionando qual unidade de disco rígido e depois percorrendo o caminho até encontrar a pasta desejada. No momento que a dita cópia é feita, o sistema prepara uma estrutura do tipo TRIE que recebe todos os paths dos arquivos encontrados e armazena seus ponteiros nos nós finais da TRIE. Com essa estrutura em mãos, é possível fazer certas buscas por nome de arquivo/diretório, (através de outra componente no menu principal), ou por prefixos de forma eficiente.

Já para o processamento de imagens, o grupo se utilizou largamente da biblioteca de imagens DevIL. A biblioteca nos permitiu com bastante liberdade abrir diversos tipos de arquivos de imagem e realizar a conversão de suas representações para uma de luminância. Tendo apenas este canal, é feito um mapeamento do conjunto dos 255 diferentes tons de cinza da imagem para um conjunto de 13 tons engenhosamente criados a partir de 3 caracteres ASCII e diferentes tonalidades de fonte e background. Para a visualização da imagem final foram criadas 2 funcionalidades: a `fitscreen`, e a de `zoom`. Como os nomes já dizem, a primeira tratava de mostrar a imagem de forma a preencher por completo a tela, e a outra aplicava algoritmos de ampliação ou afastamento conforme eram pressionadas teclas de ação. A biblioteca disponibilizava ainda filtros aplicáveis às imagens, e o grupo teve a oportunidade de executar alguns desses mas não teve tempo de os implementar de forma apresentável no módulo gráfico, com exceção de uma leve "sharpenização" da imagem que é feita no intuito de simular maior definição do que é mostrado no console.

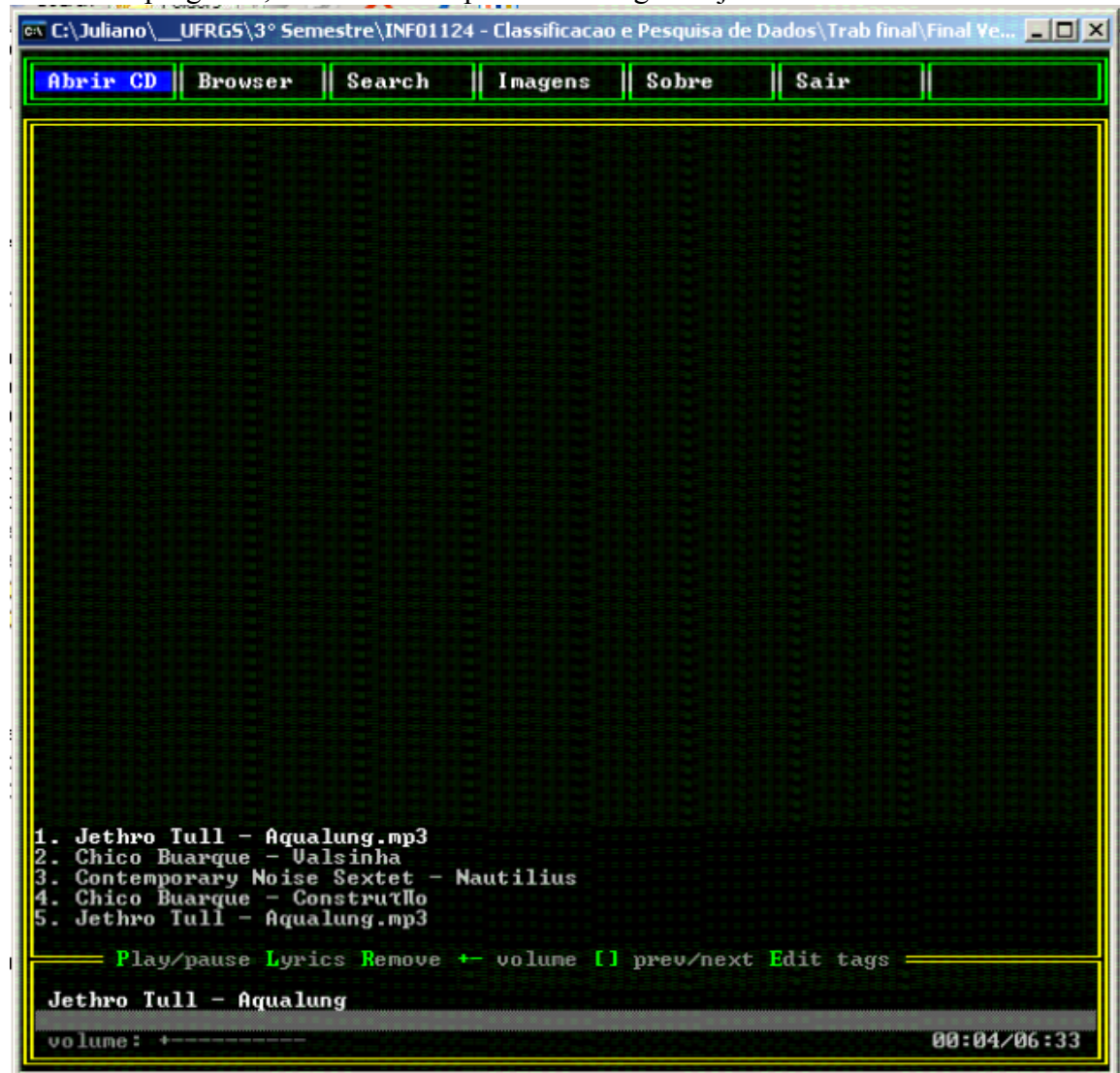
No processamento de áudio, a biblioteca preponderante foi a `irrKlang`, com a qual o grupo desenvolveu as ferramentas básicas de reprodução de áudio, como `play/pause`, `stop`, controle de volume e etc. Além disso, foram feitos alguns componentes extras como playlists dinâmicas, download e visualização automática de letras das músicas e visualização e edição dos conteúdos de IDtag de arquivos mp3. O módulo de áudio, na nossa implementação, estaria integrado com todo o resto do programa (menus, browser), e a ideia seria a de controlar o Player através de teclas de atalho enquanto que as demais

ações do programa principais eram executadas. Para a criação desse componente em especial foram utilizadas chamadas de threads para manter os indicadores das músicas atualizados (como barra e tempos de duração) e prevenir que isso interferisse no comportamento do software.

Devido à complexidade de execução, o módulo de vídeo foi completamente descartado logo no início da produção do software e o módulo de texto, por outro lado, foi deixado de lado devido à baixa importância e ao pequeno impacto no conjunto final de funcionalidades do programa.

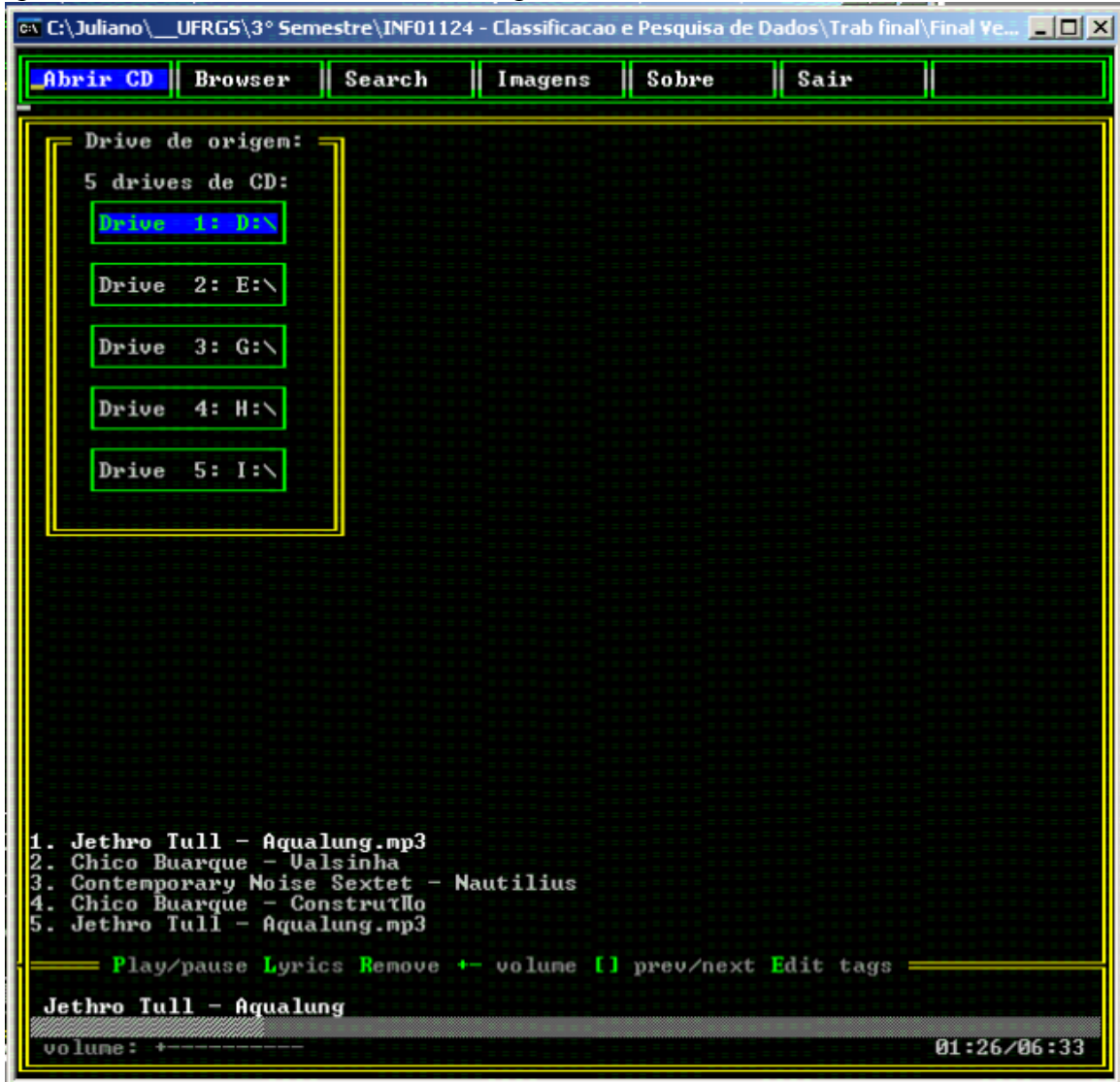
Mostraremos aqui as telas do programa, e como o usuário as percorre, da maneira mais intuitiva possível:

Ao abrir o programa, o usuário se depara com a seguinte janela:

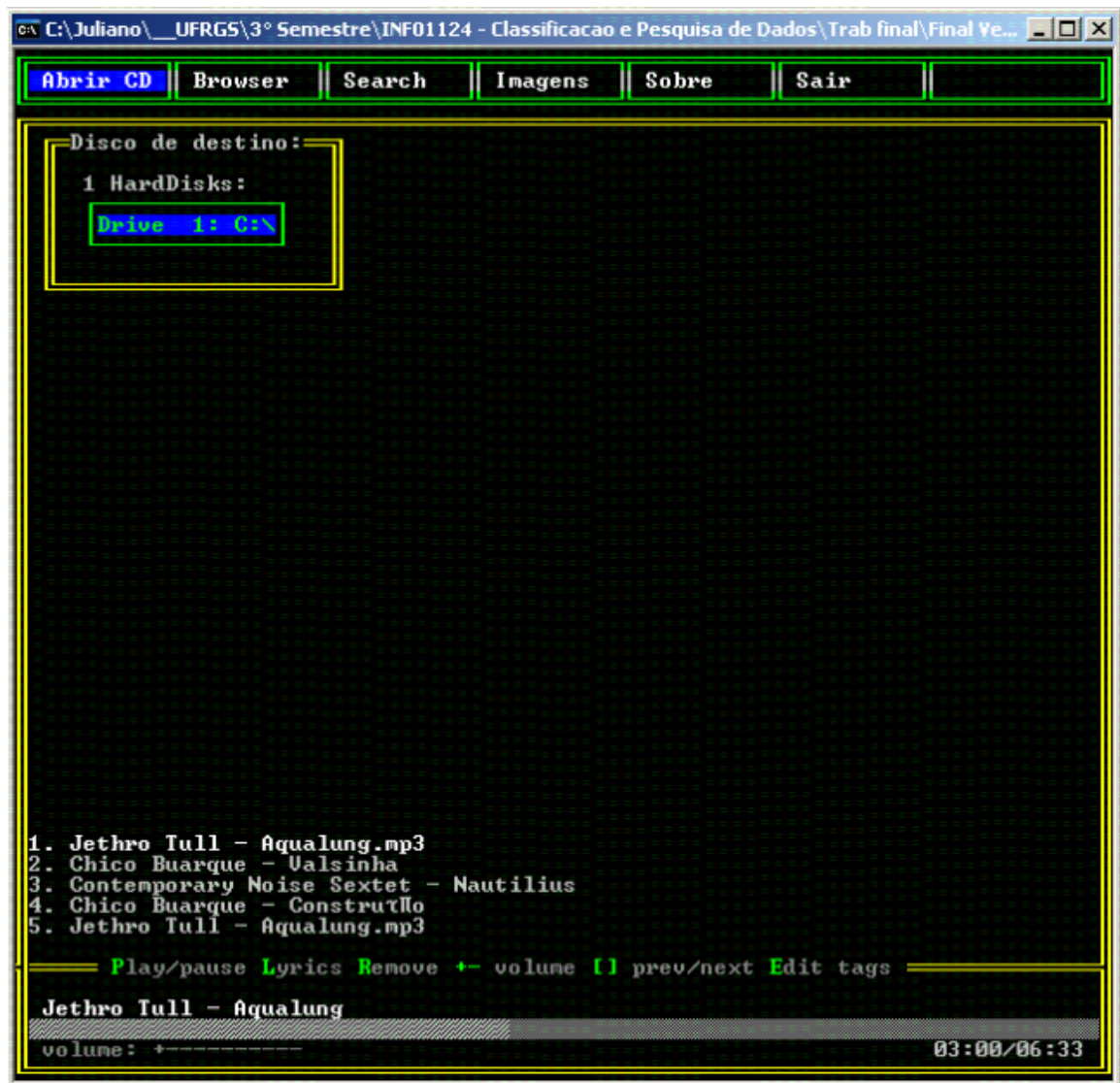


Nota-se o player de musicas na parte de baixo da janela. Ele fica ali durante toda a execucao, e seus comandos sao independentes de onde se esta no programa principal.

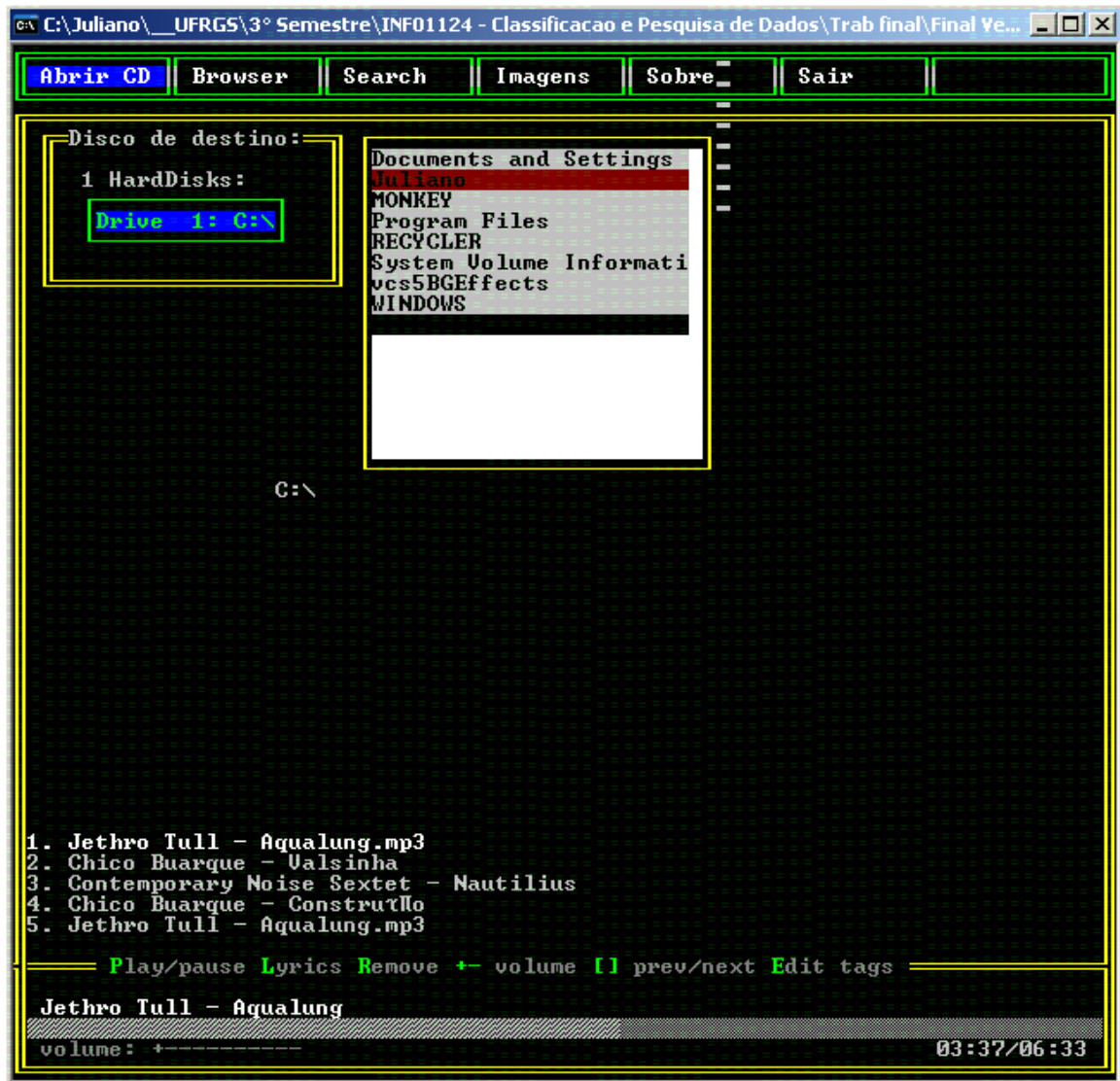
Selecionando “Abrir CD”, o programa verifica as unidades de CD-Rom disponiveis e apresenta-as num menu de selecao, da seguinte forma:



Ao selecionar a unidade de CD-Rom de origem, o usuario deve selecionar para qual HD de destino a informacao sera copiada:

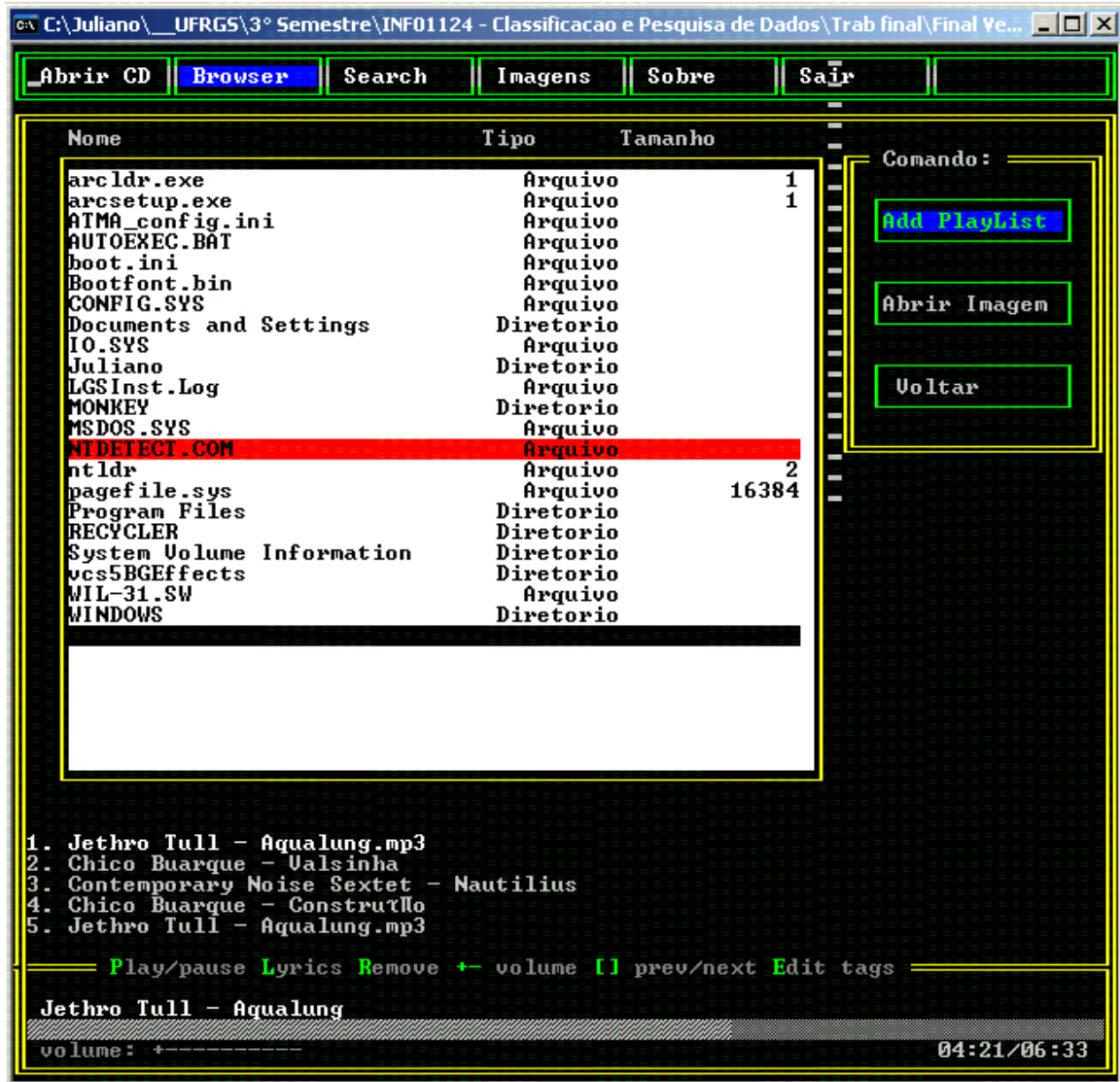


Apos selecionar a unidade de destino, deve-se percorrer o caminho ateh a pasta desejada, e entao pressionar a tecla para iniciar a copia dos arquivos (tecla padrao a ser definida).

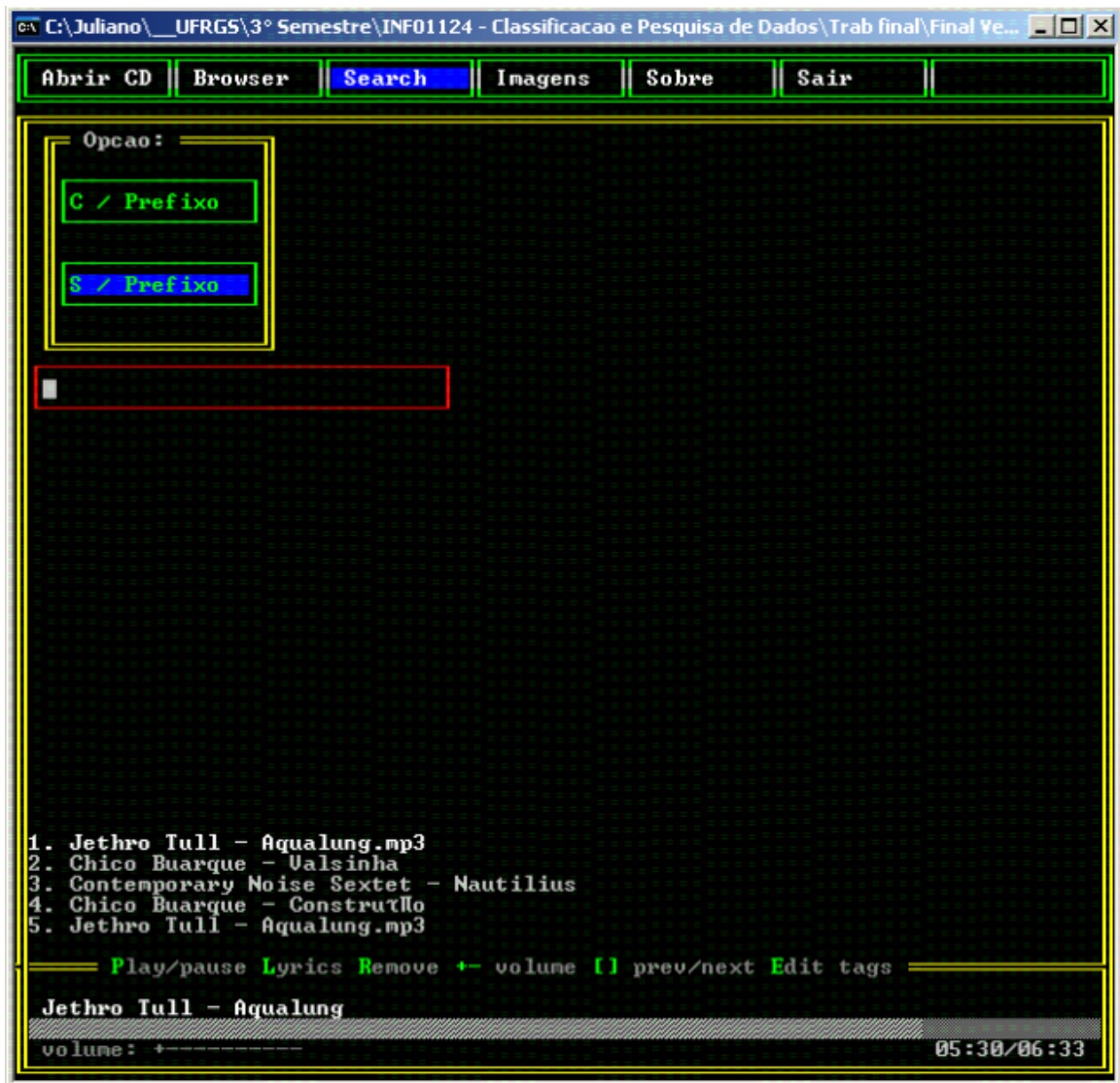


Ja para o browser, eh apresentada uma interface semelhante a anterior, mas agora mostrando tambem os arquivos, os tipos (diretorio/arquivo) e o tamanho dos arquivos. O usuario percorre a arvore de diretorios como desejar, e ao selecionar um arquivo sao apresentadas as 3 opcoes mostradas na figura, adicionar a playlist atual, abrir como imagem, e voltar ao browser:



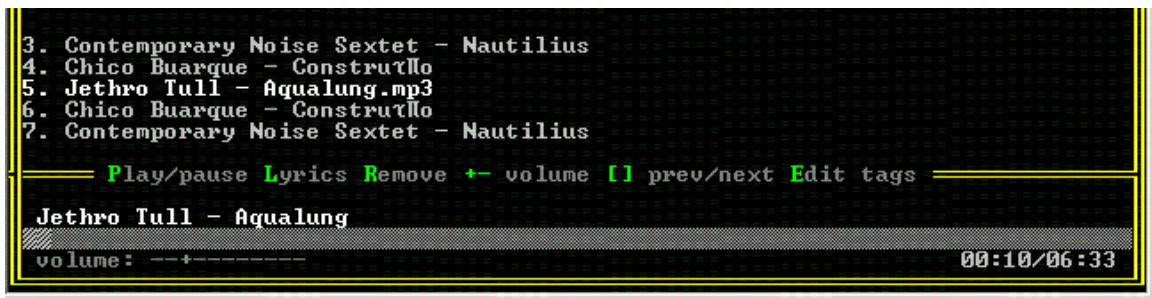


A funcionalidade de busca por arquivos/diretorios se apresenta como mostra a figura. E' possivel fazer buscas por 2 metodos diferentes, um somente pelo caminho, e outro que retorna todos os prefixos da string digitada (por exemplo, digitando um path ate uma pasta, sao apresentados todos os arquivos contidos nela).

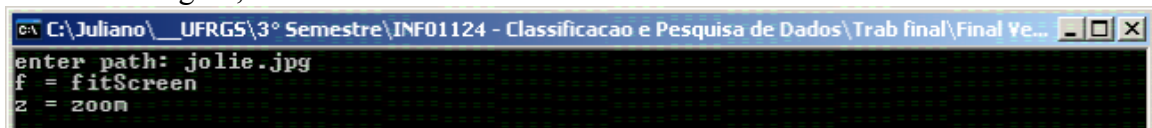


O componente do player de musicas eh integrado ao restante do trabalho, e colocado na posicao inferior da tela. Os comandos listados tem teclas de atalho destacadas em verde, e sao acionaveis e reconhecidas em qualquer momento do programa, seja no browser seja na copia dos arquivos. Da playlist atual sao mostrados 5 itens, com o item que esta sendo executado no momento no meio. A lista esta programada para ter tamanho indefinido, e realiza scroll up e down quando os botoes correspondentes sao pressionados. O player em si eh feito totalmente em uma thread de sistema separada, de forma a nao interferir no restante do processamento. A barra de progresso e o tempo da musica sao tambem atualizados de segundo em segundo.





Ao selecionar no menu o item “Imagens”, o usuário é levado a tela para inserir o caminho da imagem que será aberta pelo visualizador, e logo após é pedido o modo de exibição da figura, “fitscreen” ou “zoom”.



A foto a seguir mostra claramente os modos de exibição, em relação à imagem original (esquerda). O modo mostrado na gravação de cima à direita indica que o visualizador se encontra em “fitscreen”, já o de baixo mostra o modo “zoom”:



## Observacoes sobre o trabalho

Com tudo isso dito, existiram diversos problemas durante o desenvolvimento. O código começou a ser implementado utilizando o software Code::Blocks (e inicialmente utilizando apenas estruturas comuns e não classes), e mais tarde foi feita a transição para o DevC++, pois algumas bibliotecas extras, como a DevIL, não funcionavam corretamente (ou eram complicadas demais de instalar) com o primeiro. Nessa transição ocorreram inúmeros entraves que atrasaram significativamente o trabalho. O grupo verificou que alguns pontos são bastante distintos nos 2 compiladores, como a necessidade de definição de limites para arrays de ponteiros no Dev, o que fez com que se precisasse modificar boa parte da implementação já testada e funcional do Code::Blocks. Outra complicação a partir daí foi o início de erros inesperados em alguns pontos do código, em especial nas chamadas aos destrutores de objetos: sempre que uma janela era encerrada, vinha uma mensagem de erro, que o grupo não conseguiu de forma alguma descobrir a razão, devido aos remendos feitos durante a transformação para compilação em DevC++.

Como foi discutido brevemente durante a apresentação, algumas funcionalidades não foram implementadas a tempo. Em termos gráficos, de diversos pontos o que ficou mais significativamente de fora foi o uso do mouse como entrada. Foram feitos apenas testes sobre isso mas no fim se decidiu que seria demasiadamente complexo incrementar o programa com essa feature tão próximo ao tempo de entrega. Em se tratando de diretórios e arquivos, era de interesse da dupla fazer um navegador semelhante ao explorer.exe do windows, com direito a todas as funções padrão de deleção, criação, movimento e renomeação de arquivos e pastas, além de uma visão geral da estrutura de diretórios. Somando-se a isso também o grupo pretendia criar um menu de contexto que permitiria abrir os formatos suportados em seus devidos players, ou no caso de arquivos de áudio, de adicioná-los a uma playlist.

Das estruturas que o grupo lembra de ter ideias para implementar e que acabaram não sendo feitas, podemos listar:

- Menu de contexto no browser, ao invés de uma janela com botões;
- Dropdowns para o menu principal, apresentando funções secundárias;
- Interface mais amigável para o visualizador de imagens;
- Maior quantidade de efeitos gráficos e também efeitos na música, nos respectivos “players”;
- Correção de bugs no decorrer do programa como um todo, como cores incorretas nas listboxes, destrutores das janelas que fazem o programa terminar com erro, etc;

- Implementação de um listbox mais dinâmico, com campos para ordenação automática, barra de rolagem horizontal e buffer ilimitado (a implementação atual está limitada ao tamanho máximo de um screen buffer);
- Para a API, implementação do restante dos componentes comuns do Windows, como radio buttons, labels, checkboxes, treelists (importante para a visualização das pastas no browser), frames, etc;
- Confeção de uma janela especial multi-thread para indicar a cópia dos arquivos dos CDs, com tempo de duração estimado, barra de progresso e indicador do arquivo atual sendo copiado;
- Possibilidade de utilizar input via mouse, com funcionalidades como arrastar e redimensionar componentes da interface gráfica;
- Execução da busca mencionada nas explicações do programa feitas anteriormente – não houve implementação final da parte da interface de buscas, mas todas estruturas de busca (TRIES) e funções de “retrieval” foram implementadas e acompanham o código.

## Observações pessoais

### **Juliano:**

Gostaria aqui de reiterar parte do que foi descrito no decorrer do relatório, e admitir que uma parcela esmagadora dos problemas encontrados foram todos da minha parte. Em todos os casos, ou os problemas foram gerados por implementações mal feitas, ou por erros decorrentes da mudança de compilador durante a confecção do programa. Além disso, boa parte do que foi feito por mim acabou sendo transparente durante a apresentação do projeto, e por isso grande porção das funcionalidades finais acabaram sendo implementadas pelo Cristiano, como o visualizador de imagens e o tocador de músicas. Por diversas vezes, eu fui o “bottleneck” no desenvolvimento do projeto, pois o Cristiano estava com as funções prontas e eu estava tentando corrigir bugs com as implementações das classes para a interface gráfica.

Por essas e outras razões (muitas vezes até preguiça e desânimo da minha parte, por não aguentar mais mexer com o código fonte e me deparar com erros e mais erros), acredito sinceramente que ele deva merecer uma parcela maior de conceito do que eu com relação ao que foi feito, e fico disposto a ceder parte da minha avaliação para isso. Gostaria muito que isso fosse levado em consideração ao fechar nossos conceitos sobre o trabalho.

Já em termos de como me senti e o que aprendi de concreto com o trabalho, queria dizer que não foi uma experiência em geral positiva, na minha opinião. Fiquei bastante estressado e chateado por as vezes não conseguir fazer aquilo que sei que muitos de meus colegas (inclusive o próprio Cristiano) tem bastante facilidade, como implementar coisas simples de maneira rápida e prática. Por diversas vezes, acabei indo por um caminho completamente desnecessário para depois alguém vir e me perguntar “mas porque não fizestes assim?”.

Quando o trabalho começou a ser desenvolvido, eu tinha várias ideias para a funcionalidade da interface como um todo, mas deixei uma grande porção disso a desejar,

tanto por não ter conhecimento necessário da linguagem como por não ter as ideias que levariam a criação dessas funcionalidades.

Além disso, o que realmente aprendi tenho a impressão que poderia ter sido compreendido em poucas horas de leitura sobre as bibliotecas do windows, tanto na parte de threads, como nas funções da wincon.h do console. As vezes penso que não devia ter começado o projeto dessa forma, mas sim ter lutado apenas para ganhar a maior nota possível, e utilizado bibliotecas prontas que fizessem aquilo que eu acabei implementando “no braco”. É impressionante que com 1500-2000 linhas de código, parece que se tenha feito tão pouco em termos práticos, e é exatamente essa a sensação que tenho agora com o projeto “terminado”.

O que me vem à mente que tenha sido realmente útil foi ter uma experiência prática de como modularizar um programa em termos de código. Acabei dividindo o problema em diversos arquivos .cpp e .h, e ainda no final acabei percebendo que a divisão que fiz era insuficiente, ocasionando diversos problemas tanto para achar erros, quanto para organizar o código e as funções.

#### **Cristiano:**

Esse foi um projeto bem complicado, mas achei bem divertido também. Mexi com coisas com as quais nunca tinha mexido antes, desenvolvi ainda mais minhas habilidades com interfaces em console, mexi um pouco com orientação a objetos, e fiz muita, muita pesquisa na internet.

Tínhamos altas pretensões, que quando comparadas ao que realmente foi implementado apresenta uma grande diferença. Mas mesmo assim fiquei bastante feliz com o produto final, já que atendeu bem à essência do que era nossa proposta!

Mas o que mais me ensinou nisso tudo foi com certeza o trabalho em grupo, já que foi meu primeiro projeto de software de tal porte que não desenvolvi sozinho. Em segundo lugar, devo dizer que foi bastante interessante pôr em prática alguns dos conceitos ensinados em aula, já que a cadeira de Classificação e Pesquisa de Dados é bem pouco voltada à prática no sentido de acúmulo de experiência com a utilização das estruturas e conceitos teóricos.