

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

FERRAMENTA PARA GERENCIAMENTO E
MONITORAMENTO DE ESPAÇO EM UNIDADES DE
ARMAZENAMENTO

RICARDO COLZANI

BLUMENAU
2016

RICARDO COLZANI

**FERRAMENTA PARA GERENCIAMENTO E
MONITORAMENTO DE ESPAÇO EM UNIDADES DE
ARMAZENAMENTO**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Alexander Roberto Valdameri, Mestre - Orientador

**BLUMENAU
2016**

**FERRAMENTA PARA GERENCIAMENTO E
MONITORAMENTO DE ESPAÇO EM UNIDADES DE
ARMAZENAMENTO**

Por

RICARDO COLZANI

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente: _____
Prof. Alexander Roberto Valdameri, Mestre – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, Doutor – FURB

Membro: _____
Prof. Gabriele Jennrich Bambineti, Especialista – FURB

Blumenau, 7 de dezembro de 2016

Dedico este trabalho à minha família que sempre me apoiou durante toda trajetória acadêmica.

AGRADECIMENTOS

A Deus, por ter me guiado até aqui.

À minha família e amigos pelo total apoio recebido.

Ao meu mestre-orientador Alexander Roberto Valdameri, por toda dedicação em me orientar e por confiar no meu trabalho.

RESUMO

Atualmente a necessidade por espaço de armazenamento de dados é latente, haja vista a grande utilização de recursos de áudio, vídeo, texto entre outros. No entanto, gerenciar os dispositivos de armazenamento em um microcomputador requer análise e ações corretivas e preventivas. Este trabalho apresenta uma ferramenta para monitoramento e gerenciamento dos espaços ocupados nas unidades de armazenamento. A ferramenta possibilita a identificação de arquivos que estão consumindo a maior parte de espaço no disco, monitora diretórios enviando alertas automaticamente e com base em dados históricos realiza projeções de espaços ocupados nas unidades de armazenamento nos próximos dias, meses ou anos. Para implementação da ferramenta foi utilizada a linguagem C#, os dados são armazenados no Microsoft SQL Server e para persistência dos dados foi utilizado o Entity Framework. O desenvolvimento proposto foi realizado e todos os objetivos foram atingidos de forma satisfatória.

Palavras-chave: Monitoramento. Gerenciamento de espaço. Espaço em disco. Projeção de espaço ocupado.

ABSTRACT

Currently the need for data storage space is latent, due to the great use of audio, video, text and other resources. However, managing the storage devices in a microcomputer requires analysis and corrective and preventive actions. This work presents a tool for monitoring and managing the spaces occupied in the storage units. The tool enables the identification of files that are consuming the most disk space, monitors directories automatically sending alerts and based on historical data makes projections of occupied spaces in the storage units in the next days, months or years. For the implementation of the tool was used the C # language, storing the data using Microsoft SQL Server and for data persistence the Entity Framework was used. The proposed development was carried out and all objectives were satisfactorily achieved.

Key-words: Monitoring. Space management. Disk Space. Projection of occupied space.

LISTA DE FIGURAS

Figura 1 – Tela da ferramenta Disk Analyzer do CCleaner	18
Figura 2 – Tela inicial do software WinDirStat	19
Figura 3 – Tela de configuração de alerta do software watchDirectory.....	20
Figura 4 – Diagrama de casos de uso do ator Usuário	25
Figura 5 – Casos de uso do ator Sistema.....	26
Figura 6 – Estrutura do projeto.....	27
Figura 7 – Tela de análise de diretórios.....	37
Figura 8 – Tela de monitoramento	39
Figura 9 – Tela de unidades.....	40
Figura 10 – Tela configurações	41

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais	23
Quadro 2 – Requisitos não funcionais.....	24
Quadro 3 – Trecho de código do método <code>TarefasArquivos_DoWork</code>	29
Quadro 4 – Trecho de código dos métodos <code>ArquivosNosDiretorios</code> e <code>ArquivosNosSubDiretorios</code>	30
Quadro 5 – Trecho do código do método <code>AdicionaSeDiferente</code>	31
Quadro 6 – Trecho de código do método <code>TarefasGeral_DoWork</code>	31
Quadro 7 – Função <code>AdicionaSeNaoAdicionado</code>	32
Quadro 8 – Função <code>Enviar</code> da classe <code>FuncoesEmail</code>	33
Quadro 9 – Função <code>ProjecaoUnidadeCheia</code>	35
Quadro 10 – Comparativo entre o trabalho desenvolvido e os trabalhos correlatos	42
Quadro 11 - Descrição do caso de uso <code>Selecionar</code> diretórios.....	48
Quadro 12 - Descrição do caso de uso <code>Obter</code> informações de arquivos e diretórios	48
Quadro 13 - Descrição do caso de uso <code>Ordenar</code> lista de arquivos	49
Quadro 14 - Descrição do caso de uso <code>Cadastrar</code> conta de e-mail	49
Quadro 15 - Descrição do caso de uso <code>Enviar</code> e-mail	49
Quadro 16 - Descrição do caso de uso <code>Manter</code> arquivo.....	50
Quadro 17 - Descrição do caso de uso <code>Aplicar</code> filtro na listagem de arquivos	50
Quadro 18 - Descrição do caso de uso <code>Configurar</code> agendamento de envio de arquivos existentes dos diretórios.....	51
Quadro 19 - Descrição do caso de uso <code>Acessar</code> local do arquivo.....	51
Quadro 20 - Descrição do caso de uso <code>Configurar</code> alerta para tamanho livre dos dispositivos.....	52
Quadro 21 - Descrição do caso de uso <code>Enviar</code> alerta de tamanho livre do dispositivo.....	52
Quadro 22 - Descrição do caso de uso <code>Configurar</code> alerta para tamanho do diretório.....	53

Quadro 23 - Descrição do caso de uso Enviar alerta de tamanho do diretório	53
Quadro 24 - Descrição do caso de uso Armazenar tamanho dos dispositivos conectados	53
Quadro 25 - Descrição do caso de uso Fazer projeção de ocupação do espaço dos dispositivos.....	54

LISTA DE ABREVIATURAS E SIGLAS

C# – CSharp

HD – Hard Disk

IDE – Integrated Development Environment

Json – JavaScript Object Notation

SQL – Structured Query Language

SSD – Solid State Drive

TI – Tecnologia da informação

UC – *User Case*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 MONITORAMENTO DO AMBIENTE DE TI.....	14
2.2 ARQUIVOS	15
2.2.1 Estrutura de arquivos.....	15
2.2.2 Atributos de arquivos	16
2.2.3 Operações com arquivos	16
2.3 TRABALHOS CORRELATOS.....	17
2.3.1 CCleaner.....	17
2.3.2 WinDirStat	18
2.3.3 WatchDirectory.....	19
3 DESENVOLVIMENTO.....	21
3.1 LEVANTAMENTO DE INFORMAÇÕES.....	21
3.2 ESPECIFICAÇÃO	22
3.2.1 Especificação dos requisitos	22
3.2.2 Requisitos não funcionais	23
3.2.3 Diagrama de casos de uso	24
3.3 IMPLEMENTAÇÃO	26
3.3.1 Técnicas e ferramentas utilizadas.....	26
3.3.2 Operacionalidade da implementação	36
3.4 RESULTADOS E DISCUSSÕES.....	41
4 CONCLUSÕES.....	44
4.1 EXTENSÕES	45
REFERÊNCIAS	46
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO.....	48

1 INTRODUÇÃO

Com o passar dos anos, a capacidade de armazenamento dos dispositivos como o Hard Disk (HD), o pen drive e cartões de memória, aumentou significativamente. Segundo Brito (2012), atualmente um usuário comum possui computador com capacidade de armazenamento aproximado de até 250GB. Já para um usuário mais avançado, essa capacidade aumenta significativamente, em torno de 500GB ou mais. Isso porque quanto maior a capacidade de armazenamento do computador, maior é o espaço para instalar programas, armazenar músicas, vídeos e fotos.

Além do espaço, os usuários também procuram um dispositivo de armazenamento que seja rápido, principalmente no meio corporativo. Higa (2012) afirma que por muitos anos os avanços foram significativos apenas na capacidade de armazenamento, enquanto a velocidade foi se mantendo sempre a mesma. Mas esse cenário mudou com o avanço da tecnologia e popularização do disco de estado sólido, também conhecidos como Solid State Drive (SSD). Com o disco SSD, o ganho em desempenho chega a ser três vezes maior (YNTERNIX TEC, 2015). Outro fator que precisa ser levado em consideração é o custo de armazenamento. O SSD fica em desvantagem nesse quesito, visto que o preço por Gigabyte (GB) chega a ser cinco vezes maior que um disco rígido (HIGA, 2012).

Diante deste cenário, vê-se uma necessidade de controle do espaço nos dispositivos de armazenamento. Com vistas a este controle é que este trabalho propõe o desenvolvimento de uma ferramenta de gerenciamento de arquivos, com execução de tarefas simultâneas e automáticas, auxiliando na economia tempo do usuário.

1.1 OBJETIVOS

O objetivo desse trabalho é o desenvolvimento de uma ferramenta para auxiliar no gerenciamento de espaço nos discos e dispositivos do computador no sistema operacional Windows.

Os objetivos específicos do trabalho proposto são:

- a) disponibilizar um mecanismo para monitoramento de diretórios e unidades de armazenamento;
- b) disponibilizar o gerenciamento das configurações para envio de alertas automáticos;
- c) disponibilizar o acesso a dados sobre os arquivos e diretórios armazenados nas unidades de armazenamento.

1.2 ESTRUTURA

Esta monografia está dividida em capítulos. O primeiro refere-se à introdução e aos objetivos do trabalho. O segundo capítulo apresenta a fundamentação teórica do monitoramento no ambiente de TI, o que são arquivos, qual sua estrutura, seus atributos, as operações que podem ser realizadas com os mesmos e os trabalhos correlatos. No terceiro é demonstrada a especificação dos requisitos e casos de uso da ferramenta, assim como a implementação e a operacionalidade da implementação. Por fim, o quarto capítulo refere-se a conclusão obtida com o desenvolvimento do trabalho e quais extensões podem ser aplicadas futuramente.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos relacionados ao trabalho, tais como a estrutura de arquivos, atributos de arquivos, operações com arquivos e os trabalhos correlatos encontrados ao longo das pesquisas.

2.1 MONITORAMENTO DO AMBIENTE DE TI

Atualmente a TI desempenha um papel estratégico na gestão de muitas empresas. Cada vez mais a infraestrutura de TI precisa oferecer serviços para garantir a constante disponibilidade das aplicações (G1, 2015). Não é uma tarefa fácil garantir essa disponibilidade sem uma ferramenta de monitoramento (Pretel, 2015). Ter apenas um suporte reativo pode consumir diversas horas de trabalho na busca da causa de algum problema não tratado de forma proativa.

Conforme Mijatovic (2010, p. 1), o significado da palavra monitoramento é:

Observar em determinado período de tempo, se as condições de um objeto/equipamento estão dentro dos padrões. Difere de rastreamento que acompanha a ida e vinda de um objeto ou equipamento. O monitoramento não acompanha sua ida e vinda, somente verifica sua condição se é aceitável ou não aos padrões que se espera dele.

O monitoramento explicado por Mijatovic (2010) pode ser aplicado para a área de TI auxiliando na análise de diversos componentes de armazenamento da infraestrutura (SHRIVASTAVA; SOMAZUNDARAM, 2009). Conforme a organização cresce, o monitoramento da infraestrutura de armazenamento auxilia na otimização dos recursos e precisa acompanhar o crescimento da empresa.

Qualquer parte da infraestrutura de uma organização precisa ter o menor *downtime* possível. Para isso, segundo (SHRIVASTAVA; SOMAZUNDARAM, 2009) é necessário a implementação de ferramentas que atendam essa necessidade e satisfaçam os níveis de serviço de forma proativa. Toda a infraestrutura precisa ser monitorada e gerenciada, não apenas os itens individualmente, devido a interdependência dos componentes.

Com um monitoramento adequado, os benefícios obtidos destacados pela Equipe GD (2015) são:

- a) identificação de falhas com mais agilidade: em um ambiente com múltiplos componentes, é mais difícil encontrar o foco de um problema quando ele ocorre;
- b) correção de problemas em menos tempo: o monitoramento além de identificar e prever problemas, também oferece uma correção mais eficiente. Alguns softwares oferecem *scripts* que oferecem a solução para falhas assim que acontecem, por

exemplo;

- c) prevenção e minimização de falhas: quanto mais rápida a identificação de um problema, menor é a chance de um efeito cadeia, que induz a uma parada total do sistema;
- d) identificação do impacto no negócio: monitorar ambientes de TI permite uma estimativa mais completa do impacto das falhas na organização. Tornando mais fácil a alocação de equipes na resolução de um eventual problema;
- e) redução de custos: com um monitoramento constante, a empresa passa menos tempo procurando a solução de problemas e o sistema tem um menor *downtime*, consequentemente diminuindo os custos;
- f) aumento da eficiência: um ambiente com menos falhas e menos tempo de parada é um ambiente muito mais eficiente.

2.2 ARQUIVOS

Tudo que é armazenado nos discos rígidos são arquivos e é com base nos arquivos que é possível analisar o espaço ocupado ou livre dos diretórios e unidades. Segundo Oliveira, Carissimi e Toscani (2009), o Windows possui um sistema próprio para gerenciar arquivos chamado NTFS. Além do NTFS, o Windows oferece suporte ao FAT (MS-DOS e Windows 3.1), FAT32 (Windows 95, 98, Millenium), HPFS(OS/2), formatos para CD-ROM e UDF. Esta seção é destinada a descrever a estrutura, os atributos e as operações que podem ser realizadas com os arquivos.

2.2.1 Estrutura de arquivos

Segundo Silberschatz, Galvin e Gagne (2004), a estrutura de arquivos é dividida em duas partes: uma responsável pelos arquivos propriamente ditos, que contém os dados armazenados; e a outra parte que é referente à estrutura de diretórios, que é responsável por guardar e fornecer as informações de todos os arquivos do sistema. Stallings (2001, tradução nossa) esclarece que os diretórios também são arquivos, mantidos pelos sistemas operacionais e acessíveis por rotinas de gerenciamento de arquivos.

“Arquivo” possui um conceito muito geral, pois quem define como serão os registros, *bits* e *bytes* de cada arquivo, é o seu próprio criador. Ou seja, cada software pode criar o arquivo de um jeito diferente, que pode ser em variados formatos como: texto, números, linhas de programação, cálculos, planilhas, imagens etc. (SILBERSCHATZ; GALVIN; GAGNE, 2004).

Segundo Machado e Maia (2007), a identificação do arquivo é feita através do nome. Esse nome é composto por duas partes: a primeira é a parte do nome propriamente dito; e a segunda parte é a extensão do arquivo, que torna possível a identificação do conteúdo do arquivo. Por exemplo, um arquivo de texto possui a extensão TXT, enquanto um arquivo executável possui a extensão EXE.

2.2.2 Atributos de arquivos

Segundo Tanenbaum (2009), além do nome e extensão, os sistemas operacionais armazenam outras informações de cada arquivo. Nome, data de criação, data de modificação, tamanho, caminho e formato são algumas das informações armazenadas, que podem ser chamadas de atributos.

Machado e Maia (2007) afirmam que parte dos atributos dos arquivos podem ser modificados pelos softwares, como data/hora de criação. Outros atributos são modificados pelo próprio sistema operacional, como é o caso do tamanho do arquivo e a data da última alteração. Existem ainda atributos que o próprio usuário pode modificar, como é o caso da proteção do arquivo e senha de acesso.

Segundo Silberschatz, Galvin e Gagne (2004), entre os atributos pode-se destacar o nome, que é uma forma de facilitar ao usuário identificar cada arquivo. O identificador é o nome legível apenas pelo sistema. O tipo é fundamental para sistemas que trabalham com tipos diferentes de arquivos. Já o tamanho, indica o quanto ocupa em disco e é armazenado em bytes. A proteção define quem pode editar o arquivo. O proprietário, que segundo Stallings (2001, tradução nossa) é quem tem o total controle do arquivo, podendo permitir/negar acesso a outros usuários.

2.2.3 Operações com arquivos

Para armazenar e recuperar informações dos arquivos, os sistemas operacionais oferecem diferentes operações que podem ser realizadas pelos softwares (TANENBAUM, 2009). As principais operações que podem ser utilizadas para manipular as informações dos arquivos nos sistemas operacionais são:

- a) abertura dos arquivos. A operação “*open*” é responsável por abrir o arquivo e colocar os seus principais atributos em memória, para tornar mais rápido o acesso a esses atributos;
- b) leitura dos arquivos. A operação “*read*” é responsável pela leitura dos *bytes* do arquivo;

- c) apagar um arquivo. A operação “*delete*” é utilizada para apagar o arquivo totalmente do sistema. Dessa forma, será liberado o espaço que esse arquivo ocupa em disco.

Essas operações podem ser executadas pelo sistema operacional ao mesmo tempo em arquivos diferentes. Para isso, o software deve utilizar o conceito de programação multitarefa, que permite a execução de uma tarefa em um *thread* separado, enquanto o aplicativo principal continua respondendo.

2.3 TRABALHOS CORRELATOS

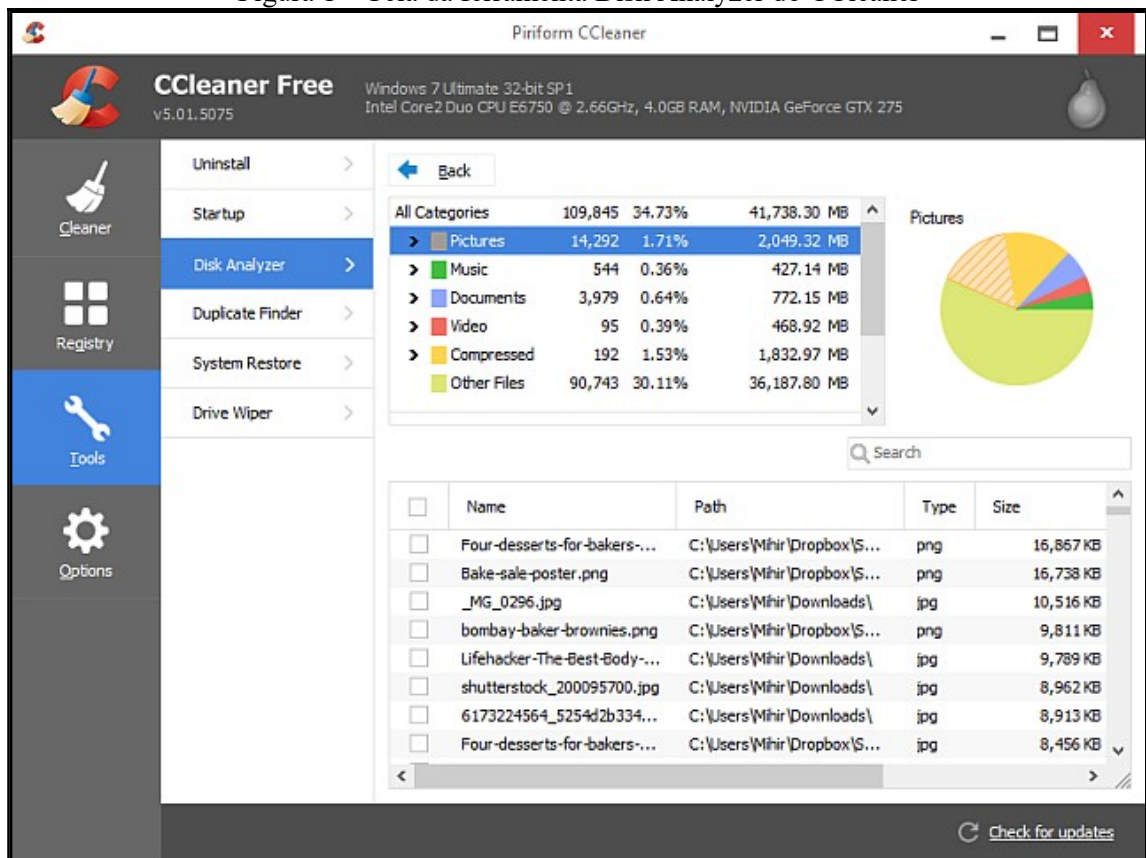
Nas pesquisas realizadas foram encontrados alguns utilitários semelhantes ao proposto. A partir da análise realizada são listadas as características e particularidades de cada um.

2.3.1 CCleaner

CCleaner é um sistema desenvolvido pela empresa Piriform (PIRIFORM, 2009). É um software voltado para a limpeza de arquivos desnecessários de um computador com sistema operacional Windows. Com ele é possível realizar uma limpeza de disco, de arquivos temporários, de registros do Windows que não são utilizados e desinstalar softwares.

A função do CCleaner que está relacionada ao trabalho proposto é chamada de Disk Analyzer. Essa função do software analisa os discos selecionados pelo usuário e exibe uma lista com os arquivos existentes. Não é possível selecionar um diretório específico, apenas analisar um disco completamente. Na Figura 1 é apresentada a tela da função Disk Analyzer do sistema CCleaner durante a edição da lista de arquivos, após ser realizada uma análise dos arquivos existentes no disco.

Figura 1 – Tela da ferramenta Disk Analyzer do CCleaner



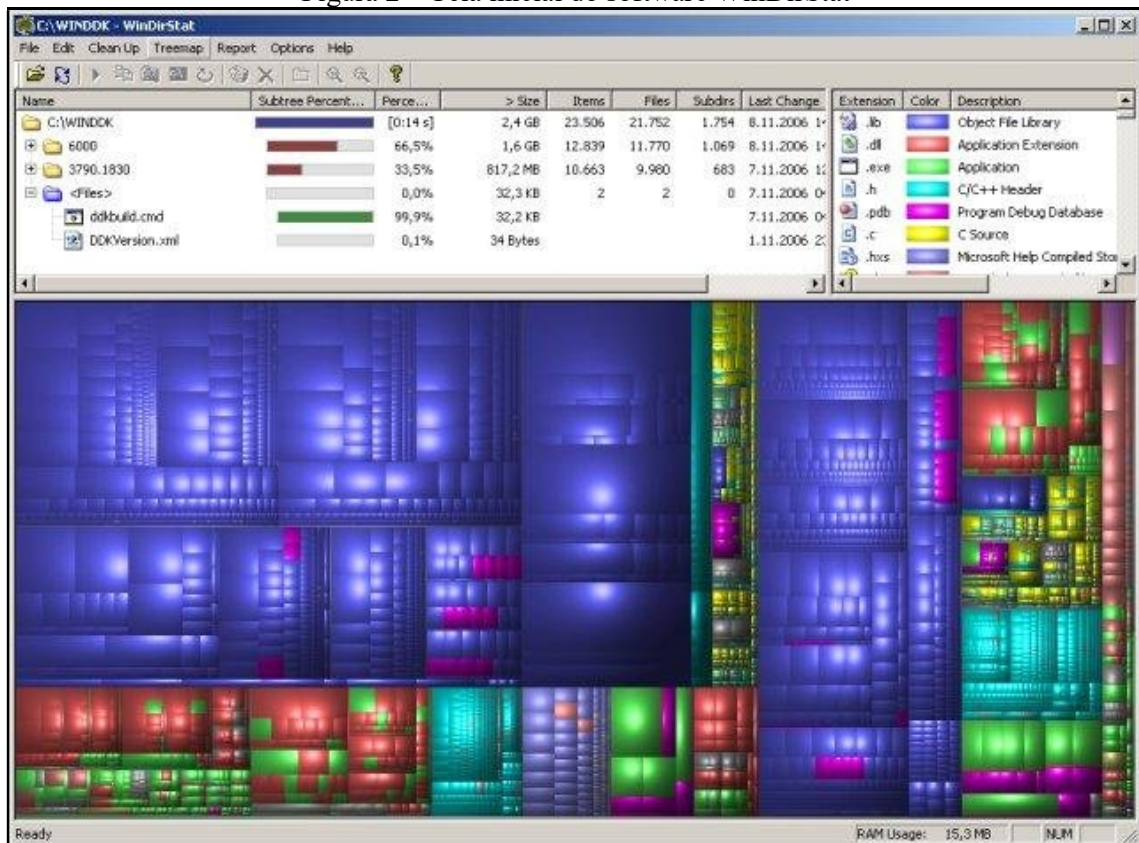
Fonte: Patkar (2014).

2.3.2 WinDirStat

O WinDirStat é um projeto *Open Source* desenvolvido na linguagem de programação C++ para Windows, que tem o objetivo de mostrar ao usuário as estatísticas de uso do disco e também exibe a posição física dos arquivos no Disco Rígido (Sourceforge, 2014). Ele permite que o usuário selecione um diretório específico ou uma unidade para análise.

Na Figura 2 é apresentada a janela principal do software WinDirStat após uma análise do disco no diretório C:\Windows. Na tela são mostradas as informações dos diretórios analisados, dos arquivos e a respectiva posição no HD.

Figura 2 – Tela inicial do software WinDirStat



Fonte: Sourceforge (2014).

2.3.3 WatchDirectory

O WatchDirectory foi desenvolvido para automatizar as tarefas repetitivas (GdP Software, 2016, tradução nossa). É uma ferramenta paga, desenvolvida pela empresa GdP Software e compatível com o sistema operacional Windows.

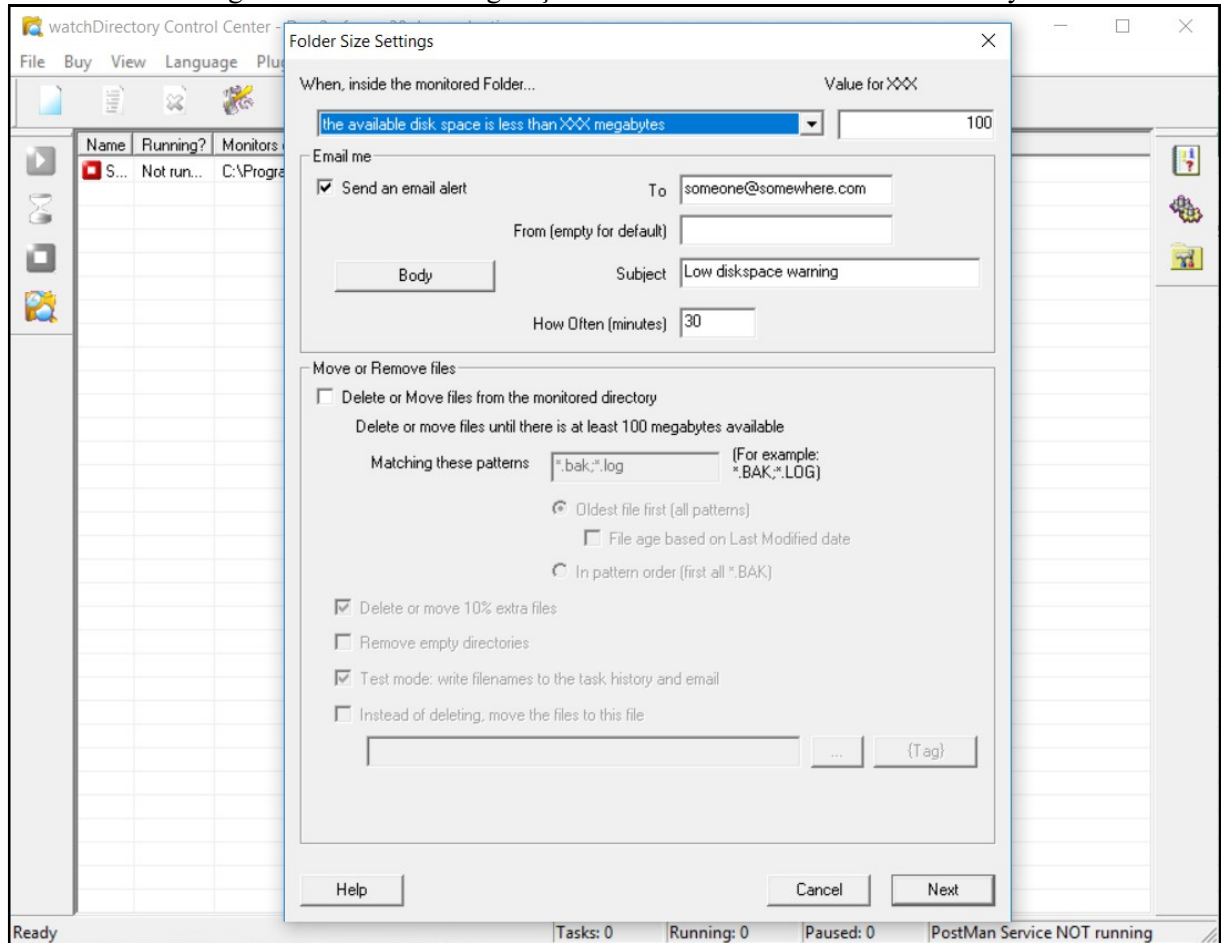
A ferramenta permite configurar cópias automáticas de arquivos, compactações automáticas, monitoramento de quantidade de arquivos em um diretório, tamanho ocupado pelo diretório e tamanho livre da unidade de armazenamento, fazendo o envio de alertas por e-mail. Todas as tarefas de monitoramento são executadas através de *threads*, podendo fazer muitas tarefas simultaneamente em vários núcleos do processador (WatchDirectory, 2016, tradução nossa).

Sempre que é detectada uma mudança em algum arquivo do computador, o aplicativo compara os arquivos com os filtros de eventos configurados. Caso alguma configuração esteja habilitada e atenda as mudanças dos arquivos, a ferramenta executa um *thread* com a função configurada.

Na Figura 3 mostra-se a tela de configuração para monitoramento do tamanho de uma pasta específica. Na configuração realizada a ferramenta enviará um e-mail para o endereço

someone@somewhere.com quando o espaço disponível na unidade for menor do que 100 megabytes.

Figura 3 – Tela de configuração de alerta do software watchDirectory



Fonte: do autor.

3 DESENVOLVIMENTO

Neste capítulo é abordado o processo para o desenvolvimento da ferramenta proposta. A seção 3.1 aborda a descrição e explanação do trabalho proposto. A seção 3.2 aborda as especificações utilizadas no desenvolvimento da ferramenta. A seção 3.3 apresenta o desenvolvimento da ferramenta. A seção 3.4 aborda os resultados e discussões obtidos com a ferramenta.

3.1 LEVANTAMENTO DE INFORMAÇÕES

A ferramenta desenvolvida neste trabalho tem como principal objetivo auxiliar o usuário do sistema operacional Windows no gerenciamento de espaço dos discos e diretórios existentes no computador, visando o aproveitamento de tempo do usuário. Para isso, o sistema exibe quais arquivos estão ocupando mais espaço no disco e automaticamente envia e-mails com o atual estado do disco ou diretório, conforme configurações realizadas.

A ferramenta permite que o usuário selecione quais diretórios deseja analisar. Os diretórios podem estar em um disco rígido do computador, em um computador da rede local ou algum disco removível. Com base na seleção realizada, a ferramenta faz o levantamento de todos os arquivos e subdiretórios existentes, e armazena esses dados em uma base de dados. Caso o usuário já tenha feito o levantamento de arquivos anteriormente, a ferramenta permite que ele continue utilizando as funções do sistema com esses dados, enquanto a ferramenta faz o novo levantamento de arquivos.

Com os dados dos diretórios e arquivos devidamente armazenados, o usuário pode selecionar filtros e ordenações dos arquivos. Os filtros e ordenações possíveis são: nome, caminho, extensão, data de criação, data de modificação e tamanho.

Enquanto o sistema exibe uma listagem de arquivos na tela, o usuário pode excluir, abrir o local em que esse arquivo se encontra ou então abrir diretamente o arquivo com o software padrão definido no Sistema Operacional Windows do usuário.

Outra funcionalidade da ferramenta é permitir que o usuário configure monitoramentos específicos, como para receber por e-mail informações sobre os diretórios e unidades de armazenamento configurados. As opções de monitoramentos que o usuário pode configurar são:

- a) determinada quantidade de arquivos atingida;
- b) determinado espaço total ocupado;
- c) lista dos últimos arquivos alterados;

- d) lista dos maiores arquivos do diretório;
- e) lista dos últimos arquivos criados;
- f) determinado espaço livre da unidade atingido;
- g) determinado espaço ocupado da unidade atingido.

Com o uso contínuo da ferramenta, são armazenados dados referentes as unidades conectadas ao computador. Com base nesses dados, a ferramenta faz projeções de ocupação de espaço, exibindo estatísticas ao usuário de quantos dias até o dispositivo estar com seu tamanho total ocupado.

3.2 ESPECIFICAÇÃO

Nesta seção são apresentados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) junto aos casos de uso da ferramenta desenvolvida. Para criar os diagramas de casos de uso foi utilizado o Enterprise Architect.

3.2.1 Especificação dos requisitos

O Quadro 1 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, a vinculação de cada um com o(s) caso(s) de uso associado(s).

Quadro 1 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir ao usuário selecionar vários diretórios do HD.	UC01
RF02: O sistema deverá permitir ao usuário selecionar vários diretórios da rede local.	UC01
RF03: O sistema deverá permitir ao usuário selecionar vários diretórios de dispositivos de armazenamento removível (Pen drive, HD externo, cartão de memória etc.).	UC01
RF04: O sistema deverá varrer os diretórios selecionados, armazenar os dados apurados e exibir a lista dos arquivos na tela.	UC02
RF05: O sistema deverá permitir ao usuário o uso das funcionalidades da ferramenta enquanto o sistema faz o levantamento de arquivos dos diretórios selecionados.	UC02
RF06: O sistema deverá permitir ao usuário ordenar a lista de arquivos pelos atributos: nome, data de criação, data de modificação, tamanho, caminho e extensão.	UC03
RF07: O sistema deve permitir ao usuário configurar uma conta de e-mail.	UC04
RF08: O sistema deverá permitir o envio de e-mail com a listagem de arquivos.	UC05
RF09: O sistema deve permitir ao usuário acessar o arquivo existente na lista.	UC06
RF10: O sistema deve permitir ao usuário excluir o arquivo analisado do computador.	UC06
RF11: O sistema deve permitir ao usuário acessar o local que o arquivo está armazenado.	UC10
RF12: O sistema deve permitir a exportação da lista de arquivos com as informações no formato texto.	UC07
RF13: O sistema deve permitir ao usuário aplicar filtros na exibição dos arquivos.	UC08
RF14: O sistema deverá permitir ao usuário configurar o agendamento de envio de e-mail com os dados dos arquivos existentes em determinados diretórios.	UC09
RF15: O sistema deverá enviar automaticamente os e-mails com os arquivos existentes nos diretórios configurados.	UC17
RF16: O sistema deverá permitir ao usuário configurar o agendamento de envio de e-mail quando um dispositivo atingir um tamanho livre específico.	UC11
RF17: O sistema deverá enviar automaticamente os e-mails quando um dispositivo atingir um tamanho livre específico configurado.	UC12
RF18: O sistema deverá permitir ao usuário configurar o agendamento de envio de e-mail quando um diretório atingir um tamanho específico.	UC13
RF19: O sistema deverá alertar ao usuário quando um diretório atingir um tamanho específico.	UC14
RF20: O sistema deverá diariamente armazenar os dados do tamanho ocupado dos dispositivos existentes.	UC15
RF21: O sistema deverá fazer uma projeção de quando os dispositivos estarão completamente cheios e enviar ao usuário.	UC16

Fonte: do autor.

3.2.2 Requisitos não funcionais

O Quadro 2 lista os requisitos não funcionais previstos para o sistema.

Quadro 2 – Requisitos não funcionais

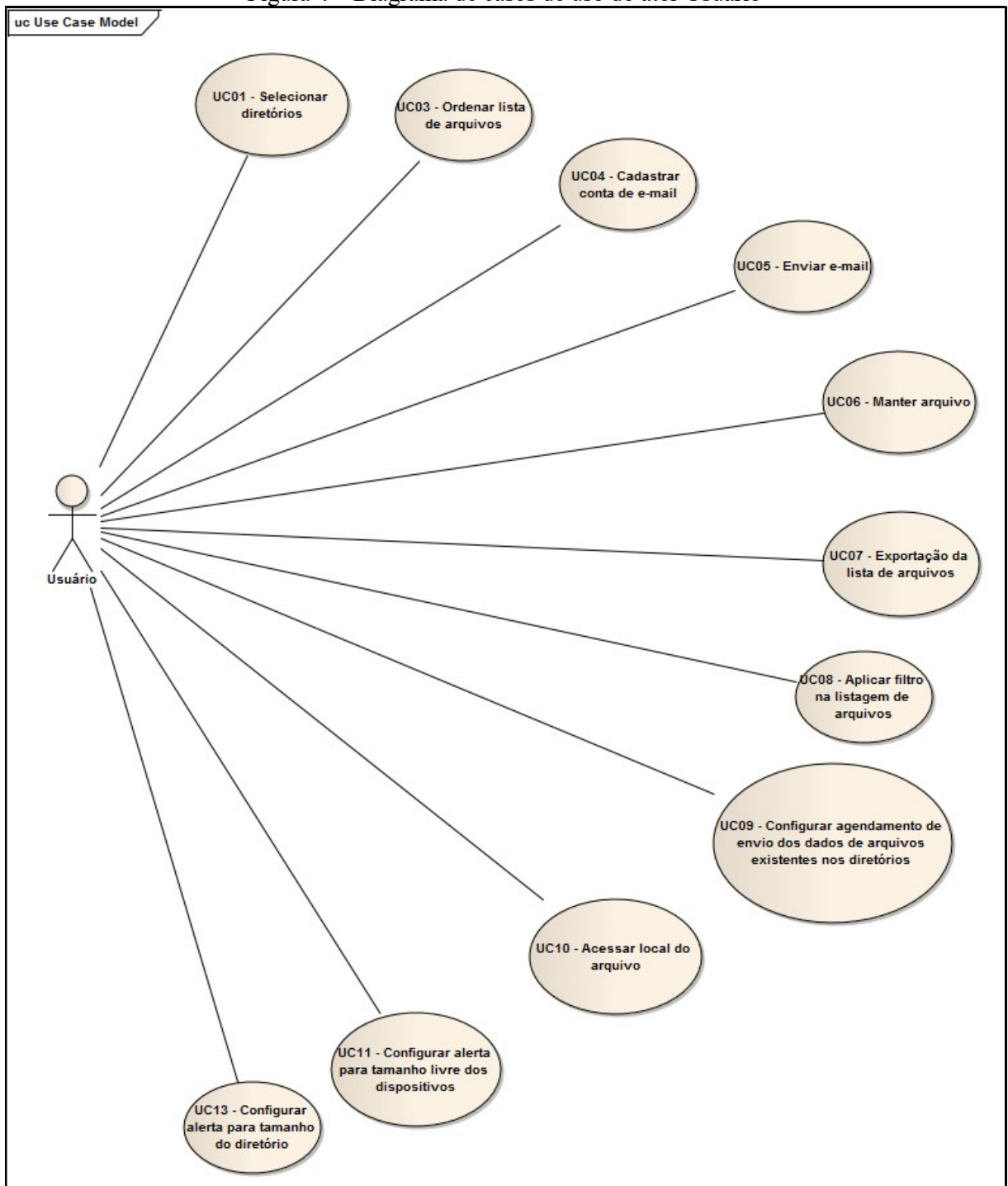
Requisitos Não Funcionais
RNF01: O sistema deverá ser desenvolvido em C# utilizando a ferramenta Visual Studio 2015.
RNF02: O sistema deverá ser acessível no Windows 7 ou superior.
RNF03: O sistema operacional precisa ter instalado o Framework 4.5.
RNF04: O sistema deverá utilizar o banco de dados Microsoft SQL Server.

Fonte: do autor.

3.2.3 Diagrama de casos de uso

Na Figura 4 é apresentado o Diagrama de Caso de Uso (UC) do autor Usuário. O UC01 - Selecionar diretórios refere-se à seleção dos diretórios que posteriormente serão utilizados para levantar os dados dos arquivos existentes, que serão exibidos em forma de lista na tela. O usuário poderá ordenar essa lista conforme o UC03 - Ordenar lista de arquivos, aplicar filtros conforme UC08 - Aplicar filtro na listagem de arquivos e acessar local dos arquivos conforme o UC10 - Acessar local do arquivo. Além das operações com arquivos, o usuário poderá criar monitoramentos específicos para os diretórios, conforme referem-se os casos de uso UC09 - Configurar agendamento de envio de arquivos existentes nos diretórios, UC11 - Configurar alerta para tamanho livre dos dispositivos e UC13 - Configurar alerta para tamanho do diretório. As descrições expandidas dos principais casos de uso estão no apêndice A.

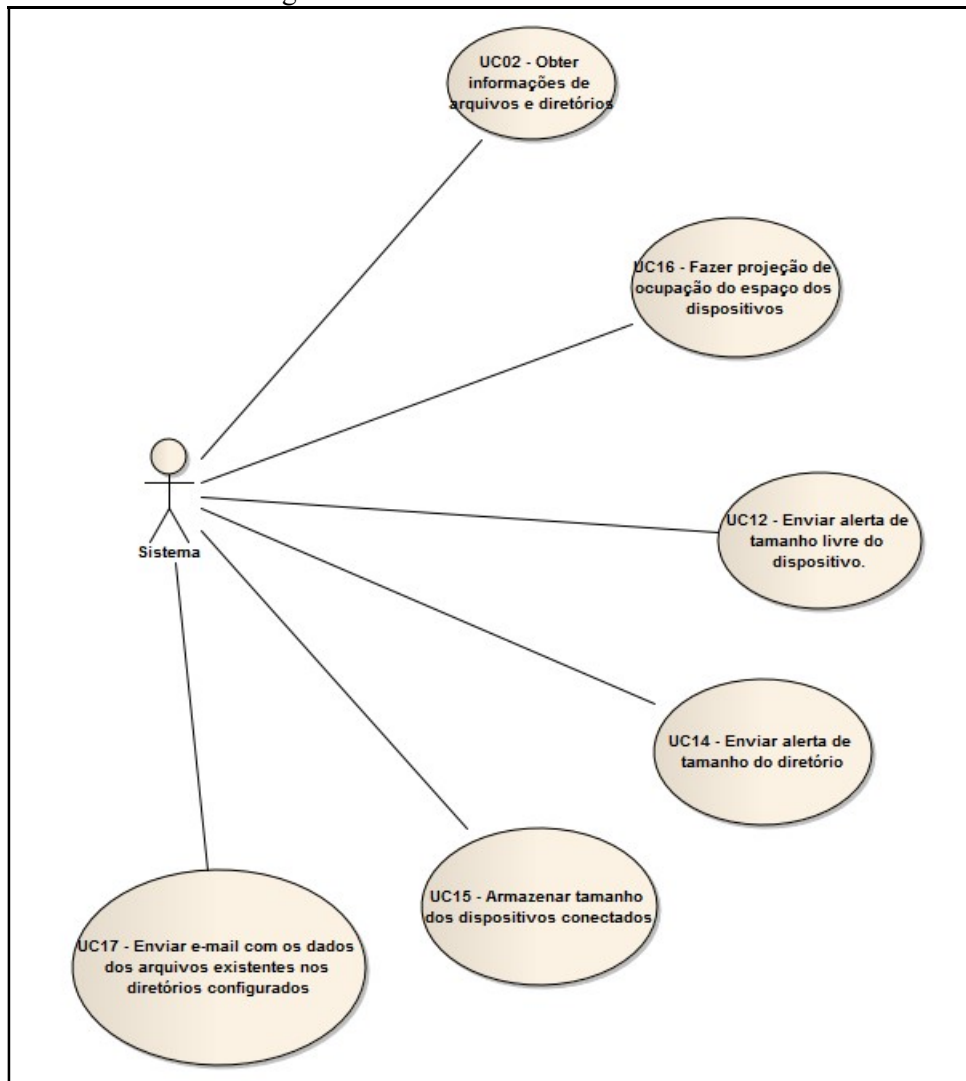
Figura 4 – Diagrama de casos de uso do ator Usuário



Fonte: do autor.

Após o usuário realizar as configurações de monitoramento da ferramenta, o sistema fará de fato o monitoramento constante e enviará os alertas para o e-mail configurado. Na Figura 5 são apresentados os casos de uso referentes ao ator sistema.

Figura 5 – Casos de uso do ator Sistema



Fonte: do autor.

3.3 IMPLEMENTAÇÃO

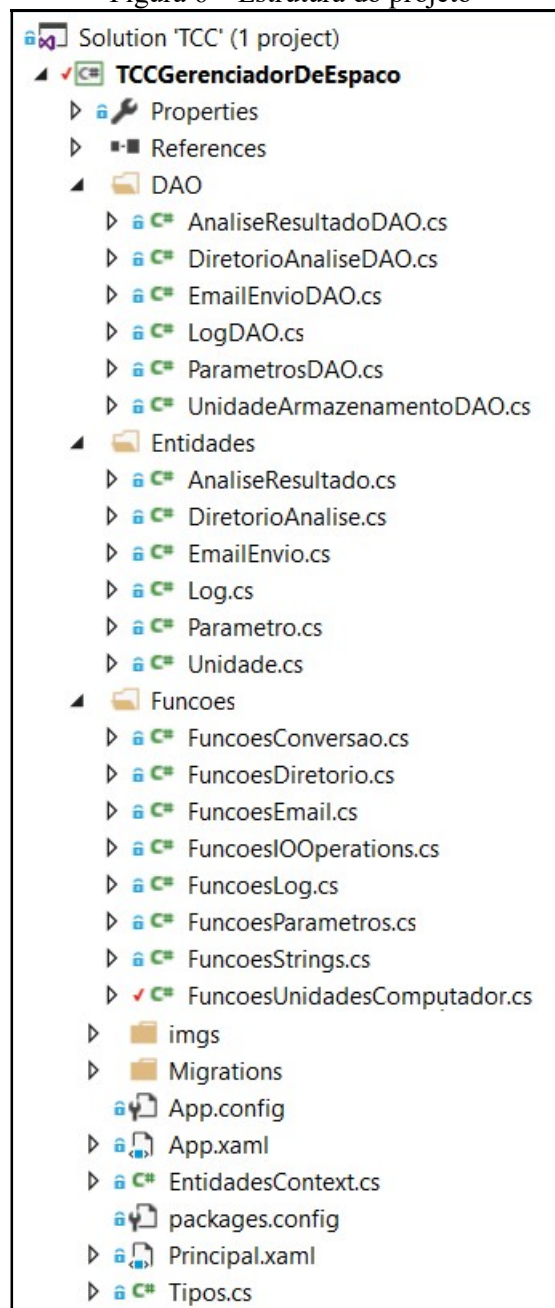
Nesta seção são descritas as informações sobre o desenvolvimento da ferramenta, como é a sua operacionalidade, quais foram os resultados obtidos e a comparação da ferramenta desenvolvida com os trabalhos correlatos.

3.3.1 Técnicas e ferramentas utilizadas

A ferramenta foi desenvolvida na linguagem C# através da Integrated Development Environment (IDE) Visual Studio Community 2015. Os dados serão armazenados no banco de dados Microsoft SQL Server. Para a persistência dos dados no SQL Server foi utilizada a ferramenta Entity Framework, nativa do Visual Studio. Para maior agilidade na gravação e seleção dos dados dos arquivos, a lista de arquivos existentes nos diretórios é convertida para o formato Json, e para isso é utilizada a biblioteca open source Json.net.

A estrutura do projeto está dividida em 3 pastas principais, conforme mostra a Figura 6, facilitando o entendimento e a manutenção dos códigos. A pasta `DAO` contém os métodos e regras para acesso ao banco de dados. Dentro da pasta `Entidades` estão todas as entidades existentes no banco de dados e mantidas pelo Entity Framework. Na pasta `Funcoes` estão todas as regras de negócio da ferramenta.

Figura 6 – Estrutura do projeto



Fonte: do autor.

O levantamento de arquivos existentes nos diretórios selecionados é feito em segundo plano, de modo que o usuário continue utilizando a ferramenta enquanto o sistema verifica se existe alguma atualização de arquivo nos diretórios. Para isso, o método

`TarefasArquivos_DoWork` existente na classe do formulário `Principal` é executado em *background*, conforme mostra-se no Quadro 3. Todos os arquivos existentes nos diretórios selecionados serão adicionados à variável privada `listaArquivos`, que é utilizada para mostrar a lista ao usuário. Essa lista é convertida para o padrão `Json` e adicionada à variável `listaArquivosJson`, que é adicionada ao banco de dados. Se a `listaArquivosJson` for diferente do último registro adicionado ao banco de dados, significa que algum arquivo possui alteração, então a variável `PvNovoLevantamentoDisponivel` será colocada como verdadeiro, significando que a lista de arquivos exibida na tela precisa ser atualizada.

Quadro 3 – Trecho de código do método `TarefasArquivos_DoWork`

```

void TarefasArquivos_DoWork(object sender, DoWorkEventArgs e)
{
    try
    {
        List<string> diretorios = new List<string>();
        DiretorioAnaliseDAO diretoriosDAO = new
DiretorioAnaliseDAO();
        foreach (string diretorio in
diretoriosDAO.RetornaTodosDiretoriosSoCaminhos())
        {
            diretorios.Add(diretorio.ToString());
        }
        Funcoes.Diretorio dir = new Funcoes.Diretorio();
        List<Tipos.Arquivo> listaArquivos = new
List<Tipos.Arquivo>();
        listaArquivos = dir.ArquivosNosDiretorios(diretorios,
true);

        string listaArquivosJson = "";

        try
        {
            listaArquivosJson =
JsonConvert.SerializeObject(listaArquivos);
        }
        catch (System.OutOfMemoryException ex)
        {
            System.Windows.Forms.MessageBox.Show("Quantidade de
arquivos existentes nos diretórios muito grande. Selecione diretórios mais
específicos!");
            return;
        }
        AnaliseResultado resultado = new AnaliseResultado();
        resultado.Resultado = listaArquivosJson;
        AnaliseResultadoDAO analiseDAO = new AnaliseResultadoDAO();
        if (analiseDAO.AdicionaSeDiferente(resultado) == true)
        {
            PvNovoLevantamentoDisponivel = true;
        }

        analiseDAO.ExcluiRegistrosAntigos();
    }
    catch (Exception ex)
    {
        Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
    }
}

```

Fonte: do autor.

Conforme mostrado no Quadro 4, o método `ArquivosNosDiretorios`, existente na classe `Diretorio`, é responsável por fazer o levantamento e retornar todos os arquivos existentes na raiz e nos subdiretórios do diretório passado como parâmetro. São percorridos todos os subdiretórios existentes no diretório parâmetro, e em cada subdiretório, são percorridos os arquivos existentes. Caso o usuário não tenha permissão para acessar um arquivo ou diretório, a rotina está preparada para ignorar essa falta de permissão e seguir adiante. Para cada arquivo do tipo `FileInfo` encontrado nos subdiretórios, será criado um

objeto do tipo `Arquivo` que receberá os dados do objeto `FileInfo`. Todos os objetos do tipo `Arquivo` são adicionados à uma lista privada na classe `Diretorio` denominada `listaArquivos`, que será retornada ao final de todo processo.

Quadro 4 – Trecho de código dos métodos `ArquivosNosDiretorios` e `ArquivosNosSubDiretorios`

```
public List<Tipos.Arquivo> ArquivosNosDiretorios(List<string> diretorios,
bool varrerSubdiretorios)
{
    foreach (string dir in diretorios)
    {
        ArquivosNoDiretorio(dir);

        if (varrerSubdiretorios)
        {
            ArquivosNosSubDiretorios(dir);
        }
    }
    return listaArquivos;
}
private void ArquivosNosSubDiretorios(DirectoryInfo sDir)
{
    try
    {
        foreach (DirectoryInfo d in sDir.GetDirectories())
        {
            try
            {
                foreach (System.IO.FileInfo f in d.GetFiles())
                {
                    listaArquivos.Add(fileInfoParaArquivo(f));
                }
                ArquivosNosSubDiretorios(d);
            }
            catch (Exception ex)
            {
                Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
            }
        }
    }
    catch (System.Exception excpt)
    {
        Funcoes.Log.GravaLog(excpt.Message, Tipos.Log.Erro);
    }
}
```

Fonte: do autor.

Na classe `AnaliseResultadoDAO`, responsável por fazer a persistência dos dados referentes as análises de arquivos existentes nos diretórios, o método `AdicionaSeDiferente` é responsável por adicionar um resultado do levantamento de arquivos no banco de dados, caso esse resultado seja diferente do último adicionado. Conforme mostrado no Quadro 5, se o resultado do levantamento de arquivos for igual ao último levantamento realizado, significa que nenhum arquivo foi modificado, logo, não precisa ser adicionado ao banco de dados.

Quadro 5 – Trecho do código do método `AdicionaSeDiferente`

```

public bool AdicionaSeDiferente(AnaliseResultado resultado)
{
    if (resultado.Horario == new DateTime())
    {
        resultado.Horario = DateTime.Now;
    }

    var existentes =
contexto.AnaliseResultados.OrderByDescending(a =>
a.Horario).FirstOrDefault();

    if (existentes == null){}
    else if (existentes.Resultado == resultado.Resultado)
    {
        return false;
    }

    contexto.AnaliseResultados.Add(resultado);
    contexto.SaveChanges();
    return true;
}

```

Fonte: do autor.

Em um intervalo de tempo configurado pelo usuário, o método `TarefasGeral_DoWork` é executado em segundo plano. Conforme ilustrado no Quadro 6, o método `TarefasGeral` é responsável pelas funções que precisam ser executadas de tempos em tempos, como é o caso do método `UnidadeHistoricoArmazenamentoRegistra`, que faz o registro do tamanho ocupado por cada unidade de armazenamento existente no computador. Também é responsável por executar a função `AgendamentosCriaEmail`, que envia os e-mails de alertas pendentes.

Quadro 6 – Trecho de código do método `TarefasGeral_DoWork`

```

void TarefasGeral_DoWork(object sender, DoWorkEventArgs e)
{
    try
    {
        UnidadeHistoricoArmazenamentoRegistra();
        Funcoes.Email email = new Funcoes.Email();
        email.AgendamentosCriaEmailSeNecessario();
        email.EnviaEmailsPendentes();
    }
    catch (Exception ex)
    {
        Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
    }
}

```

Fonte: do autor.

Para cada unidade de Disco Rígido, mapeamento da rede ou removível (pen drive e cartão de memória) existente no computador, a ferramentas cria uma chave única baseada nos dados da própria unidade. Essa chave é utilizada para registrar o histórico do espaço usado e espaço livre. Conforme mostra o Quadro 7, esse registro é feito a cada doze horas. Esse

histórico é utilizado pela ferramenta para exibir ao usuário uma estimativa de quando a unidade ficará cheia, para que seja possível controlar essa situação.

Quadro 7 – Função AdicionaSeNaoAdicionado

```
public bool AdicionaSeNaoAdicionado(Unidade unidadeAdicionar)
{
    try
    {
        var unidades = contexto.Unidades.FirstOrDefault(d =>
d.RegistroDataAdicionado > DateTime.Now.AddHours(-12) && d.ChaveUnica ==
unidadeAdicionar.ChaveUnica);
        if (unidades != null)
        {
            return false;
        }
        unidadeAdicionar.RegistroDataAdicionado = DateTime.Now;
        Adiciona(unidadeAdicionar);
        return true;
    }
    catch (Exception ex)
    {
        Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
        return false;
    }
}
```

Fonte: do autor.

Conforme for configurada, a ferramenta monitora e envia por e-mail alertas se um diretório ou unidade está em determinada situação. No Quadro 8, tem-se a rotina para preenchimento das variáveis com os dados configurados e o envio do e-mail. Para que a rotina funcione, é necessário que o usuário configure uma conta de e-mail que está habilitada ao envio via software.

Quadro 8 – Função Enviar da classe FuncoesEmail

```

public bool Enviar(string[] emailsDestinatarios, string titulo, string
mensagem)
    {
        try
        {
            System.Net.Mail.SmtpClient client = new
System.Net.Mail.SmtpClient();
            client.Host = Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailSMTPServidor);
            client.Port =
int.Parse(Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailSMTPPorta));

            if (Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailSSLAtivar) != "1")
                client.EnableSsl = false;
            else
                client.EnableSsl = true;

            client.UseDefaultCredentials = false;
            string usuario = Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailUsuario);
            string senha = Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailSenha);
            client.Credentials = new
System.Net.NetworkCredential(usuario, senha );

            MailMessage mail = new MailMessage();
            string emailEndereco =
Funcoes.Parametros.RetornaParametroValor(
ParametroChave.ConfigGeraisEmailConta);
            mail.Sender = new
System.Net.Mail.MailAddress(emailEndereco, "Monitoramento de espaço");
            mail.From = mail.Sender;

            foreach (string email in emailsDestinatarios)
            {
                mail.To.Add(new MailAddress(email, ""));
            }

            mail.Subject = titulo;
            mail.Body = mensagem;
            mail.IsBodyHtml = true;
            mail.Priority = MailPriority.High;
            client.Send(mail);
            return true;
        }
        catch (System.Exception ex)
        {
            Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
            return false;
        }
    }

```

Fonte: do autor.

Com o uso contínuo da ferramenta, diariamente são armazenados dados de espaço ocupado pelas unidades para que seja realizada uma projeção de espaço ocupado. Conforme mostrado no Quadro 9, a rotina `ProjecaoUnidadeCheia` é responsável pelo cálculo da

projeção. Com os dados dos últimos trinta dias a ferramenta calcula a média diária de espaço ocupado pela unidade. O espaço livre é dividido pela média diária, onde obtém-se a quantidade de dias até a unidade estar totalmente cheia.

Quadro 9 – Função ProjecaoUnidadeCheia

```

public static Tipos.Projecao ProjecaoUnidadeCheia(Entidades.Unidade
unidade, int qtdDiasListaProjecao)
{
    try
    {
        DAO.UnidadeArmazenamentoDAO dao = new
DAO.UnidadeArmazenamentoDAO();
        DateTime dataInicial = DateTime.Now.Date.AddDays(-30);
        List<Entidades.Unidade> historico =
dao.RetornaHistorico(unidade.ChaveUnica, dataInicial);

        if (historico.Count <= 1)
        {
            return new Tipos.Projecao();
        }

        var primeiroRegistro = historico.FirstOrDefault(f =>
f.RegistroDataAdicionado >= dataInicial);
        var ultimoRegistro = historico.OrderByDescending(o =>
o.RegistroDataAdicionado).FirstOrDefault();
        double diferencaDias =
(ultimoRegistro.RegistroDataAdicionado.Date -
primeiroRegistro.RegistroDataAdicionado.Date).TotalDays;
        double somaEspacos = 0;
        double espacoOcupadoAnterior = 0;
        foreach (var h in historico)
        {
            if (espacoOcupadoAnterior == 0)
            {
                espacoOcupadoAnterior = h.EspacoOcupado;
            }
            else
            {
                somaEspacos += h.EspacoOcupado -
espacoOcupadoAnterior;
                espacoOcupadoAnterior = h.EspacoOcupado;
            }
        }

        double media = somaEspacos / diferencaDias;

        if (media <= 0)
        {
            return new Tipos.Projecao();
        }

        int qtdDiasFaltaEncher =
(int)(unidade.EspacoLivreTotalBytes / media);
        int qtdDiasProjecao = (int)qtdDiasFaltaEncher /
qtdDiasListaProjecao;
        Tipos.Projecao projecao = new Tipos.Projecao();
        projecao.ChaveUnica = unidade.ChaveUnica;
        projecao.Letra = unidade.LetraUnidade;
        projecao.QtdDiasFaltantesParaEncherUnidade =
qtdDiasFaltaEncher;
        projecao.MediaDiariaMB =
Funcoes.Conversao.BytesToMegabytes((long)media, 3);
        projecao.Historico = new
List<Tipos.ProjecaoDataEEspacoOcupado>();
        projecao.Historico.Add(new Tipos.ProjecaoDataEEspacoOcupado

```

```

{ Data = DateTime.Now, EspacoOcupadoMB =
Funcoes.Conversao.BytesToGigabytes(ultimoRegistro.EspacoOcupado, 2) });

        for (int i = 1; i < qtdDiasListaProjecao - 1; i++)
        {
            projecao.Historico.Add(new
Tipos.ProjecaoDataEEspacoOcupado
            {
                Data = DateTime.Now.AddDays(qtdDiasProjecao * i),
                EspacoOcupadoMB =
Funcoes.Conversao.BytesToGigabytes((long)(ultimoRegistro.EspacoOcupado + (i
* qtdDiasProjecao * media)), 2)
            });
        }

        projecao.Historico.Add(new Tipos.ProjecaoDataEEspacoOcupado
        {
            Data = DateTime.Now.AddDays(qtdDiasFaltaEncher),
            EspacoOcupadoMB =
Funcoes.Conversao.BytesToGigabytes((long)(ultimoRegistro.EspacoOcupado +
qtdDiasFaltaEncher * media), 2)
        });
        return projecao;
    }
    catch (Exception ex)
    {
        Funcoes.Log.GravaLog(ex.Message, Tipos.Log.Erro);
        return new Tipos.Projecao();
    }
}

```

Fonte: do autor.

3.3.2 Operacionalidade da implementação

Nesta subseção apresentam-se as telas do sistema com uma apresentação sobre seu uso e suas funcionalidades.

3.3.2.1 Análise de diretórios

Na Figura 7 é exibida a tela de Análise de diretórios. Na análise de diretórios o usuário seleciona quantos diretórios forem necessários e a ferramenta fará um levantamento de todos os arquivos existentes nesses diretórios. Quando todos os diretórios foram percorridos e os dados dos arquivos estão disponíveis para o usuário, será mostrada uma mensagem em vermelho, alertando que a lista pode ser atualizada. Após o usuário iniciar o levantamento, sempre que algum dos arquivos da lista for alterado, a ferramenta mostrará o alerta para que o usuário atualize a lista mostrada. Isso é possível porque a ferramenta faz em *background* comparações entre os dados exibidos na lista e os existentes nos diretórios, sem interromper o uso da ferramenta.

A lista dos arquivos pode ser ordenada por qualquer um dos atributos mostrados, clicando no título da coluna que se deseja ordenar. É possível que o usuário filtre a lista dos

arquivos, para exibir apenas arquivos com determinado nome, caminho, tamanho e data de alteração. Para excluir ou abrir o diretório de um arquivo, basta o usuário clicar com o botão direito em cima do arquivo desejado e selecionar a opção.

Figura 7 – Tela de análise de diretórios

Nome	Tamanho (MB)	Criação	Modificação	Extensão	Bytes	Caminho
12470006.MOV	1136.53	26/06/16 12:47	26/06/16 13:08	.MOV	1191738181	D:\Ricardo\fotos
17240024.MOV	344.02	20/11/16 23:09	12/01/14 17:27	.MOV	360726251	D:\Ricardo\fotos
17180022.MOV	344.02	20/11/16 23:09	12/01/14 17:21	.MOV	360627947	D:\Ricardo\fotos
17210023.MOV	344.02	20/11/16 23:09	12/01/14 17:24	.MOV	360562411	D:\Ricardo\fotos
21190011.MOV	167.39	16/09/16 20:03	25/07/16 21:23	.MOV	246909753	D:\Ricardo\outr
E05 - Versão 2 - orientações.RTF	167.33	16/09/16 20:02	09/11/16 22:17	.RTF	194731815	D:\Ricardo\FURE
10490001.MOV	167.33	16/09/16 20:03	29/07/16 10:52	.MOV	175554281	D:\Ricardo\outr
11310015.MOV	167.39	16/09/16 20:03	29/07/16 11:34	.MOV	175521513	D:\Ricardo\outr
11160010.MOV	167.39	16/09/16 20:03	29/07/16 11:19	.MOV	175521513	D:\Ricardo\outr
11040006.MOV	167.33	16/09/16 20:03	29/07/16 11:07	.MOV	175455977	D:\Ricardo\outr
11130009.MOV	167.33	16/09/16 20:03	29/07/16 11:16	.MOV	175455977	D:\Ricardo\outr
11430019.MOV	167.23	16/09/16 20:03	29/07/16 11:46	.MOV	175357673	D:\Ricardo\outr
11070007.MOV	167.2	16/09/16 20:03	29/07/16 11:10	.MOV	175324905	D:\Ricardo\outr
11010005.MOV	167.17	16/09/16 20:03	29/07/16 11:04	.MOV	175292137	D:\Ricardo\outr
10580004.MOV	167.17	16/09/16 20:03	29/07/16 11:01	.MOV	175292137	D:\Ricardo\outr
10550003.MOV	167.14	16/09/16 20:03	29/07/16 10:58	.MOV	175259369	D:\Ricardo\outr
11190011.MOV	167.11	16/09/16 20:03	29/07/16 11:22	.MOV	175226601	D:\Ricardo\outr
11100008.MOV	167.11	16/09/16 20:03	29/07/16 11:13	.MOV	175226601	D:\Ricardo\outr

Extensao	Ocupado	Quantida
Total	15986.67	31559
.jpg	2569.09	10202
.gif	2.44	3889
.jar	330.9	2489
.js	28.98	1956
.JPG	2499.38	1644
.html	10.81	1552

Fonte: do autor.

A Figura 7 mostra o levantamento de arquivos realizado nos diretórios “D:\Ricardo”, “C:\Users\Ricardo\Music” e “H:\BACKUPTCC”. Após o levantamento, todos os arquivos existentes nesses diretórios estão listados e ordenados pelo tamanho, em ordem decrescente. A lista com o “Resumo dos arquivos” mostra a soma do tamanho ocupado por cada extensão de arquivo, em ordem decrescente de espaço ocupado. Como a ferramenta identificou que algum arquivo que está na lista foi alterado externamente, o alerta “Alguns arquivos foram alterados. Atualize a lista!” é exibido. Quando o usuário achar necessário, basta clicar no botão Atualizar lista, que a lista de arquivos será atualizada e o alerta será ocultado.

3.3.2.2 Monitoramento

A tela de monitoramento permite que o usuário faça configurações para que a ferramenta envie alertas por e-mail. Assim que a ferramenta identificar que os arquivos existentes no diretório configurado se enquadraram em um dos alertas, será enviado um e-mail informando ao usuário. O mesmo e-mail é enviado a cada 12 horas, até que o alerta não seja mais necessário.

Conforme é mostrado na Figura 8, podem ser configurados alertas para diretórios específicos e para unidades de armazenamento, tantos quanto necessários. Para os diretórios, é possível que sejam realizadas as seguintes configurações:

- a) quantidade de arquivos maior que: enviará alerta quando a quantidade de arquivos existentes no diretório ultrapassar o número informado;
- b) espaço total ocupado maior que: enviará alerta quando o espaço total ocupado pelos arquivos do diretório ultrapassaram o número informado;
- c) lista dos X últimos arquivos alterados: enviará uma lista com os dados dos últimos arquivos alterados no diretório;
- d) lista dos X maiores arquivos: enviará uma lista com os dados dos maiores arquivos existentes no diretório;
- e) lista dos X últimos arquivos criados: enviará uma lista com os dados dos últimos arquivos criados no diretório;
- f) considerar apenas arquivos da extensão: em todos os alertas que serão enviados apenas serão considerados arquivos da extensão definida;
- g) considerar arquivos dos subdiretório: marcando essa opção, serão considerados os arquivos dos subdiretórios para o envio dos alerta. Caso contrário, serão considerados apenas arquivos da raiz do diretório;
- h) assunto do e-mail: é o assunto do e-mail dos alertas que serão enviados.
- i) e-mail destino: é o e-mail que receberá os alertas.

Para as unidades de armazenamento, é possível que sejam configurados os seguintes alertas:

- a) espaço livre menor que: enviará um alerta quando a unidade estiver com o espaço livre menor que o configurado;
- b) espaço ocupado maior que: enviará um alerta quando a unidade estiver com o espaço ocupado maior que o configurado;
- c) x dias para a unidade estar totalmente ocupada: enviará um alerta quando a função de projeção de espaço identificar que a unidade estará completamente cheia em x dias;
- d) projeção a cada x dias: enviará um alerta a cada x dias com a projeção do espaço ocupado pela unidade de armazenamento;

Figura 8 – Tela de monitoramento

The screenshot shows a software interface for monitoring files, directories, and storage units. It is divided into two main sections: 'Diretório' (Directory) and 'Unidades' (Units).

Diretório Section:

- Directory path: C:\Users\Ricardo\Documents
- Quantity of files greater than: 6000
- Total space occupied greater than (MB): 10000
- Lists of 15 items: últimos arquivos alterados, maiores arquivos, últimos arquivos criados
- Consider only files of extension: (empty)
- Considerar arquivos dos subdiretórios
- Assunto do e-mail: Monitoramento da pasta Documents
- E-mail destino: ricardo.colzani@hotmail.com
- Buttons: Novo, Salvar, Excluir

Unidades Section:

- Dispositivo/Unidade: C -
- Projeção a cada: 3 dias
- Espaço livre menor que (GB): 30
- Assunto do e-mail: Monitoramento da unidade C
- E-mail destino: ricardo.colzani@hotmail.com
- Espaço ocupado maior que (GB): 0
- 8 dias para a unidade estar totalmente ocupada
- Buttons: Novo, Salvar, Excluir

Fonte: do autor.

A Figura 8 mostra a tela de configuração de monitoramentos com duas configurações, uma do diretório “C:\Users\Ricardo\Documents” e a outra configuração referente a unidade C. A ferramenta detecta automaticamente quando deve alertar por e-mail, com base nas configurações realizadas pelo usuário.

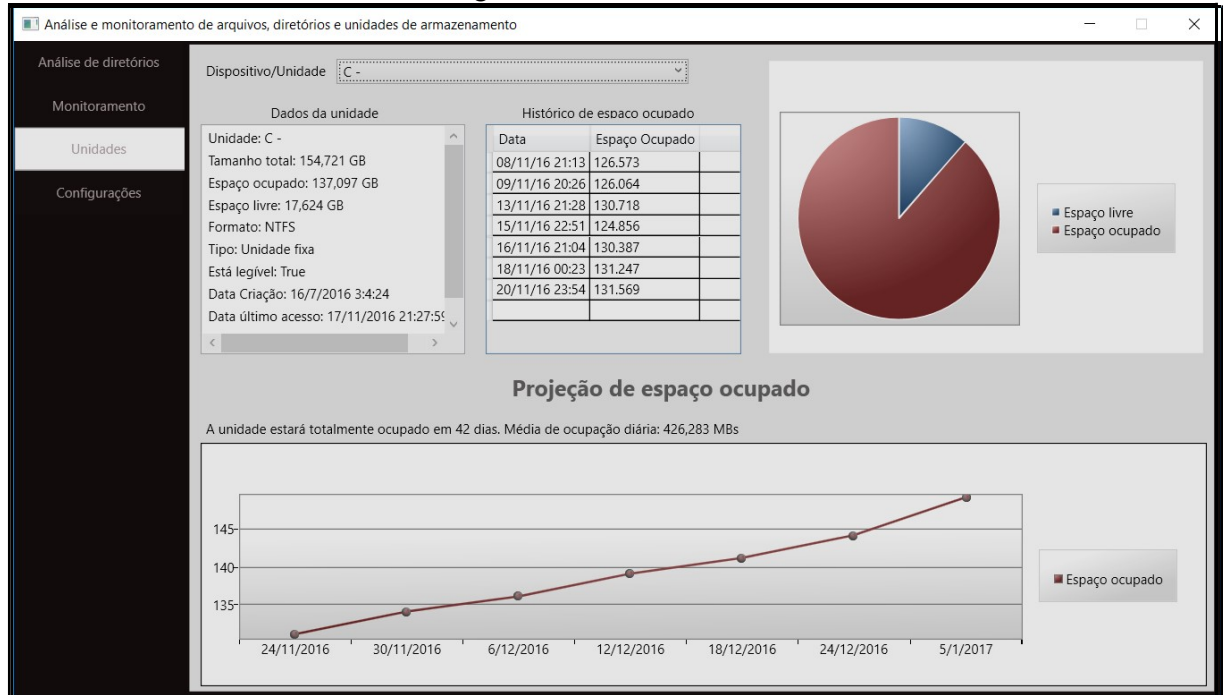
3.3.2.3 Unidades

A tela de unidades fornece ao usuário dados de todas as unidades existentes no computador. São detectados e mostrados ao usuário HDs, pen drives, unidades de CD, unidades mapeadas da rede e qualquer outra unidade de armazenamento conectada ao computador.

Conforme mostrado na Figura 9, a tela é dividida em duas partes, uma com os dados atuais da unidade e outra parte com a projeção de espaço ocupado. Nos dados da unidade a ferramenta lista um resumo da unidade selecionada, com o nome, tamanho total, espaço livre, espaço ocupado, formato, tipo, data de criação, data do último acesso e data da última gravação. Na lista com o histórico de espaço ocupado são listados os dados armazenados pela ferramenta nos últimos trinta dias, com a data e o espaço ocupado pela unidade. Pode-se ver qual espaço a unidade estava ocupando em cada um dos dias que a ferramenta foi utilizada. Também é exibido um gráfico em forma de pizza com com os dados atuais de espaço livre e espaço ocupado da unidade, para uma fácil visualização do usuário.

Para realizar a projeção de espaço ocupado, a ferramenta utiliza os dados que foram armazenados nos últimos trinta dias. A ferramenta utiliza esses dados para fazer uma média de aumento diário, com base no aumento diário é calculado a quantidade de dias faltantes para atingir o tamanho total da unidade e esses dias são mostrados no gráfico em forma de linha.

Figura 9 – Tela de unidades



Fonte: do autor.

Na Figura 9 são mostrados os dados da unidade C. O histórico de espaço ocupado é mostrado do dia 08/11/2016 até o dia 20/11/2016 e com base nesses dados foi realizada a projeção de espaço ocupado. Conforme mostrado no gráfico em linha, a unidade C estará com o espaço total cheio aproximadamente no dia 05/01/2017.

3.3.2.4 Configurações

Na tela de configurações são feitas as configurações gerais da ferramenta e mostrados os eventos que ocorreram em *background*. Conforme mostrado na Figura 10, é possível configurar o intervalo de tempo em que a rotina de monitoramento será checada e o intervalo que a lista de arquivos será levantada. O usuário também pode configurar um tamanho mínimo de arquivo, para que no levantamento de arquivos só sejam considerados arquivos maiores do que o configurado e não cause uma possível falta de memória, que acontece se a lista de arquivos ultrapassar oitocentos mil arquivos. É necessário que o usuário configure os dados de uma conta de e-mail, que será utilizada para realizar o envio dos e-mails de monitoramentos.

Figura 10 – Tela configurações

Gerais

Timer de monitoramento minutos
 Tamanho mínimo de arquivo (MB)

Levantamento de arquivos a cada segundos

Conta de E-mail

E-mail conta
 Servidor SMTP
 Usuário
 Porta SMTP
 Senha
 SSL

Eventos ocorridos

Descrição	Horário
E-mail enviado para monitoramentoarquivos@gmail.com com o títuloMonitoramento da pasta Documents	25/11/16 00:03
E-mail inserido na fila de envio. Para: ricardo.colzani@hotmail.com. TítuloMonitoramento da pasta Documents	25/11/16 00:03
Alerta agendado para envio: Últimos arquivos criados	25/11/16 00:03
Alerta agendado para envio: Últimos arquivos alterados	25/11/16 00:03
Alerta agendado para envio: Maiores arquivos do diretório	25/11/16 00:03
E-mail enviado para monitoramentoarquivos@gmail.com com o títulomusicas espaço	24/11/16 22:46
E-mail inserido na fila de envio. Para: ricardo.colzani@hotmail.com. Títulomusicas espaço	24/11/16 22:46
Alerta agendado para envio: Últimos arquivos criados	24/11/16 22:46
Alerta agendado para envio: Últimos arquivos alterados	24/11/16 22:46

Fonte: do autor.

3.4 RESULTADOS E DISCUSSÕES

A ferramenta desenvolvida possibilitou o gerenciamento dos arquivos e o monitoramento dos diretórios, que eram os objetivos principais. A linguagem C# possui todas as bibliotecas necessárias para manipulação de arquivos e envio de e-mails no Windows, tornando o desenvolvimento das funções da ferramenta mais fácil.

O desenvolvimento da ferramenta foi realizado conforme as necessidades e problemas que o autor observou durante um período de experiência em atendimentos de telessuporte. Vários dos problemas relatados pelos usuários eram relacionados à falta de espaço nas unidades de armazenamento e dificuldades em ter uma gestão de suporte proativa, no sentido de diminuir essa problemática. Os dados que a ferramenta apresenta ao usuário e a possibilidade de configurar alertas por e-mail elimina quase por completo a necessidade de um suporte reativo.

Durante o desenvolvimento e testes, foram encontrados alguns problemas. Após estudos e modificações, todas as dificuldades encontradas foram corrigidas. Um deles foi a tentativa de uso de um banco de dados embarcado, que inicialmente foi cogitada a ideia de usar o SQLite, já que o armazenamento de dados não seria o foco da ferramenta. Mas ao longo no desenvolvimento de um protótipo, notou-se a dificuldade de integração com o Visual Studio. E, para contornar a situação, optou-se pelo SQL Server, que integra

nativamente com o Visual Studio. A performance da ferramenta não foi comprometida com essa alteração.

Outra dificuldade encontrada foi no uso da biblioteca Json.NET. No momento de converter uma lista de objetos para o formato Json, se a lista de arquivos for muito extensa, a biblioteca entra na exceção de falta de memória. Nos testes realizados, acontece essa exceção quando a lista de arquivos ultrapassa oitocentos mil arquivos. Nesse caso, foi criado um parâmetro nas configurações da ferramenta para selecionar no levantamento de arquivos apenas arquivos superiores ao tamanho mínimo especificado. Dessa forma, arquivos pequenos, que seriam irrelevantes para o usuário, são ignorados pela ferramenta, e o problema é contornado. Essa situação ocorre apenas quando a lista de arquivos existentes nos diretórios é mostrada para o usuário. No caso dos alertas por e-mail, nada é afetado, pois todo o processo é baseado diretamente na lista de objetos, sem necessidade da conversão para Json.

A Figura 11 mostra o desempenho da ferramenta na função responsável por fazer o levantamento da lista de arquivos. O computador utilizado nos testes é um modelo DELL – Vostro 3560, com processador Core i5 (3ª geração) e 4GB de memória RAM. As unidades de armazenamento utilizadas são descritas na Figura 11.

Figura 11 – Desempenho da ferramenta desenvolvida

Levantamento dos dados de todos os arquivos existentes em um diretório selecionado			
	Quantidade	Tempo(segundos)	Arquivos por segundo
Disco Local - Seagate 500GB - 7200 RPM	36594	1,96	18670
HD Externo - Toshiba 1 TB - USB 3.0	30592	2,01	15219
Pasta mapeada da rede - velocidade da conexão: 100 MBs	36594	6,94	5272

Fonte: do autor.

O Quadro 10 apresenta uma comparação entre as principais funcionalidades da ferramenta desenvolvida e dos trabalhos correlatos. Com relação aos trabalhos correlatos, eles possuem algumas das funcionalidades da ferramenta desenvolvida, mas apenas isoladamente.

Quadro 10 – Comparativo entre o trabalho desenvolvido e os trabalhos correlatos

	Trabalho desenvolvido	CCleaner	WinDirStat	WatchDirectory
Levantar arquivos de diretório específico	X		X	X
Levantar arquivos de uma unidade específica	X	X	X	X
Selecionar vários diretórios no levantamento de arquivos existentes	X			
Excluir arquivos	X	X	X	X
Monitoramento de diretório	X			X
Projeção do espaço ocupado da unidade de armazenamento	X			

Fonte: do autor.

Nenhum dos trabalhos possui todas as funções que a ferramenta desenvolvida possui, como é mostrado no Quadro 10. Todos os trabalhos comparados possuem a funcionalidade de fazer o levantamento de arquivos existentes permitindo a exclusão desses arquivos, mas cada um com suas particularidades. Apenas o trabalho desenvolvido tem a opção de selecionar vários diretórios e exibir todos os arquivos existentes nesses diretórios. Os demais permitem selecionar apenas um diretório por vez.

A ferramenta WatchDirectory é bem semelhante ao trabalho proposto quando se refere a monitoramento de diretórios. Nas duas ferramentas é possível configurar monitoramentos que enviarão alertas por e-mail quando um diretório excedeu uma quantidade de arquivos, excedeu um tamanho ocupado e quando uma unidade de armazenamento possui determinado tamanho disponível ou ocupado.

Apenas o trabalho proposto possui a funcionalidade de projetar o espaço que será ocupado nas unidades de armazenamento, com base nos dados históricos que são armazenados conforme o uso da ferramenta.

4 CONCLUSÕES

O objetivo de desenvolver uma ferramenta que facilitasse o monitoramento e gerenciamento de espaço nas unidades de armazenamento foi alcançado. A ferramenta realiza monitoramentos constantemente e envia alertas por e-mail ao usuário. Mostra ao usuário quais arquivos estão consumindo espaço nos diretórios. É capaz de fazer uma projeção de espaço ocupado nas unidades com base no histórico armazenado.

A ferramenta foi testada por dez dias seguidos e mostrou-se eficiente no gerenciamento de espaço. Utilizando as funções da ferramenta foi possível realizar uma limpeza de forma rápida e precisa nas unidades de armazenamento, assim como identificar o que realmente está ocupando espaço. Sempre que necessário, a ferramenta enviou alertas para o e-mail configurado, auxiliando o efetivo gerenciamento de espaço sem a necessidade de um controle manual.

Com os testes, também foi constatado que a funcionalidade de projeção de espaço ocupado nas unidades de armazenamento se mostra mais efetiva a medida que a ferramenta é usada constantemente. Com isso, é recomendável que a ferramenta seja utilizada diariamente, armazenando dados do histórico de espaço ocupado pelas unidades e, podendo assim, fazer uma projeção mais precisa.

Quando a lista de arquivos é muito extensa, encontrou-se dificuldade no momento de converter o objeto para uma *string* e armazenar no banco de dados, quando a lista de arquivos existentes ultrapassa 800 mil arquivos. Nesse caso, a ferramenta faz o tratamento dessa exceção e exibe uma mensagem ao usuário, avisando dessa situação. Para contornar essa situação, foi criada a configuração de tamanho mínimo de arquivo, que pode ser configurada pelo usuário e faz com que a ferramenta descarte os arquivos menores, que seriam irrelevantes para o usuário.

Os softwares e bibliotecas utilizados se mostraram muito eficientes. O Visual Studio utiliza o .NET Framework, que possui várias bibliotecas nativas para se trabalhar com arquivos e diretórios, o que facilitou o desenvolvimento nesse sentido. A biblioteca Json.net se mostrou muito ágil e de fácil uso para serialização e deserialização de objetos.

O Entity Framework em conjunto com o SQL Server facilitou o desenvolvimento e manutenção dos códigos para persistência dos dados. Facilitou principalmente a escrita dos comandos, visto que possibilitou a persistência dos dados com orientação a objetos.

Além de a ferramenta apresentar os resultados esperados, todo o desenvolvimento foi de grande valia para o autor, expandindo o conhecimento na linguagem C# e no .NET

Framework, assim como no desenvolvimento utilizando o Entity Framework para persistência dos dados, antes desconhecido.

4.1 EXTENSÕES

São sugeridas as seguintes extensões para continuidade da ferramenta para gerenciamento e monitoramento do espaço nos unidades de armazenamento:

- a) consultar dados das unidades de armazenamento em um Web Site em tempo real;
- b) enviar alertas *push* e SMS para smartphones com os dados dos monitoramentos;
- c) receber comandos remotamente através de Web Site ou smartphone, para criação de monitoramentos específicos e obtenção de dados dos arquivos existentes nas unidades.

REFERÊNCIAS

- BRITO, Felipe. **Qual o tamanho de HD ideal para o seu PC.** [S.l.], 2012. Disponível em: <<http://www.zoom.com.br/pc-computador/deumzoom/qual-o-tamanho-de-hd-ideal-para-o-seu-pc>>. Acesso em 25 mar. 2016.
- EQUIPE GD. **Entenda por que monitorar ambientes de TI.** [S.l.], 2015. Disponível em: <<https://gdsolutions.com.br/monitoramento-de-ti/entenda-por-que-monitorar-ambientes-de-ti>>. Acesso em: 21 out. 2016.
- G1. **Monitoramento de TI traz agilidade para os negócios.** [S.l.], 2015. Disponível em: <<http://g1.globo.com/economia/especial-publicitario/embratel/pense-inovacao/noticia/2015/01/monitoramento-de-ti-traz-agilidade-para-os-negocios.html>>. Acesso em: 20 out. 2016.
- GDP SOFTWARE. **Automate your repetitive tasks.** [S.l.], 2008. Disponível em: <<http://www.watchdirectory.net/index.html>>. Acesso em: 10 out. 2016.
- HIGA, Paulo. **Tudo o que você precisa saber sobre SSDs.** [S.l.], 2012. Disponível em: <<https://tecnoblog.net/108784/ssd-tudo-sobre/>>. Acesso em 25 mar. 2016.
- MACHADO, Francis B.; MAIA, Luiz P.; **Arquitetura de sistemas operacionais.** 4. ed. Rio de Janeiro:LTC, 2007.
- MEDEIROS, Higor. **Programação com Threads: Parte I.** [S.l.], 2012. Disponível em: <<http://www.devmedia.com.br/programacao-com-threads-parte-i/6152>>. Acesso em 1 abr. 2016.
- MIJATOVIC, Fredy. **Significado de Monitoramento.** [S.l.], 2010. Disponível em:<<http://www.dicionarioinformal.com.br/monitoramento>>. Acesso em 19 out. 2016.
- OLIVEIRA, Rômulo S.; CARISSIMI, Alexandre S.; TOSCANI, Simão S. **Sistemas Operacionais.** Porto Alegre: Bookman, 2010. 373 p.
- PATKAR, Mihir. **The best windows cleaning tool gets a Disk Analyzer.** [S.l.], 2014. Disponível em: <<http://usetechology.org/tag/best-windows-cleaning-tool-gets-disk-analyzer/>>. Acesso em 30 mar. 2016.
- PIRIFORM. **What is CCleaner.** [S.l.], 2009. Disponível em: <<http://www.piriform.com/docs/ccleaner/introducing-ccleaner/what-is-ccleaner>>. Acesso em 25 mar. 2016.
- PRETEL, Jorge. **A importância do monitoramento no ambiente de TI.** [S.l.], 2015. Disponível em:<<https://www.profissionaisti.com.br/2015/09/a-importancia-do-monitoramento-no-ambiente-de-ti/>>. Acesso em: 21 out. 2016.
- SHRIVASTAVA, Alok; SOMAZUNDARAM, G.. **Armazenamento e gerenciamento de informações: Como armazenar, gerenciar e proteger informações digitais.** Porto Alegre: Bookman, 2009. 476 p.
- SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. **Fundamentos de sistemas operacionais.** 6. ed. Tradução Elisabete do Rego Lins. Rio de Janeiro: LTC, 2004.
- SOURCEFORGE. **Windows Directory Statistics.** [S.l.], 2014. Disponível em: <<https://sourceforge.net/projects/windirstat/>>. Acesso em 30 mar. 2016.
- STALLINGS, William. **Operating systems: internals and design principles.** 4. Ed. Upper Saddle River: Prentice Hall, 2001.

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. 3. ed. Tradução Ronaldo A. L. Gonçalves, Luís A. Consularo, Luciana do Amaral Teixeira. São Paulo: Pearson Prentice Hall, 2009.

WINDIRSTAT. **Windows Directory Statistics**. [S.l.], 2013. Disponível em: <<https://windirstat.info/>>. Acesso em 25 de mar. 2016.

YNTERNIX YT. **SSD x HD**: Saiba a diferença, as vantagens e desvantagens. [S.l.], 2015. Disponível em: <<http://www.ynternix.com/ssd-x-hd-saiba-diferenca-vantagens-e-desvantagens/>>. Acesso em 25 mar. 2016.

APÊNDICE A – Descrição dos Casos de Uso

Nos quadros abaixo tem-se o detalhamento dos casos de uso exibidos no diagrama apresentado na subseção 3.2.3.

No Quadro 11 apresenta-se o caso de uso *Selecionar diretórios*:

Quadro 11 - Descrição do caso de uso *Selecionar diretórios*

UC01	Selecionar diretórios
Descrição	É necessário que o usuário selecione um conjunto de diretórios. Com base nesse conjunto, o sistema fará o levantamento dos arquivos e subdiretórios. O sistema buscará todos os arquivos e subdiretórios existentes nesses diretórios.
Ator	Usuário
Fluxo principal	Na tela principal, o sistema permite que o usuário faça a seleção de diretórios. O usuário seleciona ou escreve o caminho dos diretórios que deseja. O sistema valida se o usuário tem permissões para acessar os diretórios.
Cenário – Exceção	Se, no passo 3, o usuário não tem permissão para acessar um dos diretórios informados, o sistema deve apresentar a mensagem: "Caminho inexistente ou inacessível!".
Pós-condição	Sistema fará o levantamento dos arquivos existentes nos diretórios.

Fonte: do autor.

No Quadro 12 apresenta-se o caso de uso *Obter informações de arquivos e diretórios*:

Quadro 12 - Descrição do caso de uso *Obter informações de arquivos e diretórios*

UC02	Obter informações de arquivos e diretórios
Descrição	O sistema fará uma varredura nos diretórios selecionados para localizar os arquivos e subdiretórios existentes. Esses dados serão armazenados.
Ator	Sistema
Pré-condição	Usuário ter informado no mínimo um diretório acessível.
Fluxo principal	O sistema verifica se o usuário tem permissão para acessar todos os diretórios que foram informados. Através de um <i>thread</i> secundário, o sistema inicia a verificação dos arquivos existentes nos diretórios informados. Ao final do levantamento, o sistema armazena os dados. O sistema sugere ao usuário atualizar a listagem de arquivos que ele o usuário está editando.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui permissão para acessar algum diretório, o sistema deve ignorar esse diretório e informar ao usuário.
Pós-condição	Sistema gravará os dados encontrados e informará ao usuário.

Fonte: do autor.

No Quadro 13 apresenta-se o caso de uso *Ordenar lista de arquivos*:

Quadro 13 - Descrição do caso de uso Ordenar lista de arquivos

UC03	Ordenar lista de arquivos
Descrição	É permitido que o usuário ordene a lista de arquivos que está na tela inicial pelos atributos: nome, extensão, caminho, data da criação, data de modificação, tamanho e formato. Dessa forma, pode identificar, por exemplo, qual o maior arquivo de todos os exibidos na tela.
Ator	Usuário
Pré-condição	Ter algum arquivo na lista de exibição de arquivos.
Fluxo principal	Usuário clica em um dos atributos dos arquivos exibidos. O sistema faz a ordenação desses arquivos. O sistema exibe os arquivos ordenados.
Pós-condição	A lista de arquivos será ordenada conforme a ação do usuário.

Fonte: do autor.

No Quadro 14 apresenta-se o caso de uso Cadastrar conta de e-mail:

Quadro 14 - Descrição do caso de uso Cadastrar conta de e-mail

UC04	Cadastrar conta de e-mail
Descrição	O usuário precisará cadastrar todos os dados de uma conta de e-mail para que os e-mails sejam enviados a partir dessa conta.
Ator	Usuário
Fluxo principal	Usuário preenche o nome da conta. Usuário preenche o endereço de e-mail. Usuário preenche a senha do e-mail. Usuário preenche o endereço SMTP. Usuário preenche a porta do SMTP. Sistema valida e grava as informações.
Cenário – Exceção	Se, no passo 6, o sistema encontrou alguma inconsistência nas validações, exibirá mensagem para o usuário fazer a correção
Pós-condição	Cadastrará o e-mail no sistema, permitindo que os alertas agendados sejam enviados a partir dessa conta.

Fonte: do autor.

No quadro 7 apresenta-se o caso de uso Enviar e-mail:

Quadro 15 - Descrição do caso de uso Enviar e-mail

UC05	Enviar e-mail
Descrição	O sistema permite que o usuário faça o envio por e-mail da listagem dos arquivos exibidos na tela para um e-mail definido pelo usuário. Os arquivos serão enviados na ordenação em que eles estão apresentados na tela.
Ator	Usuário
Pré-condição	O usuário precisa ter cadastrado um e-mail no sistema.
Fluxo principal	Usuário clica para fazer o envio dos arquivos por e-mail. Sistema verifica se a conta de e-mail está cadastrada. Sistema exibe o último e-mail de destinatário utilizado para o envio. Usuário altera o e-mail do destinatário se achar necessário. Sistema envia o e-mail.
Cenário – Exceção	Se, no passo 2, o sistema verificar que não existe nenhum e-mail cadastrado, é exibida a mensagem: "O e-mail não está configurado".
Pós-condição	O e-mail será enviado com a listagem de arquivos em anexo.

Fonte: do autor.

No Quadro 16 apresenta-se o caso de uso Manter arquivo:

Quadro 16 - Descrição do caso de uso Manter arquivo

UC06	Manter arquivo
Descrição	O sistema permitirá que o usuário exclua ou abra qualquer arquivo existente na lista. Se o usuário optar por excluir, esse arquivo será excluído do disco em que se encontra.
Ator	Usuário
Pré-condição	Ter algum arquivo na lista de arquivos. O arquivo selecionado ainda existir no disco.
Fluxo principal	O usuário seleciona arquivo(s) da lista. O usuário clica com o botão direito e escolher qual ação deseja: abrir ou excluir. O sistema verifica se os arquivos ainda existem e estão acessíveis. Sistema executa a operação desejada pelo usuário.
Cenário – Exceção	Se, no passo 3 o sistema identificar que um dos arquivos não existe mais, exibe uma mensagem ao usuário.
Pós-condição	Os arquivos são excluídos. Os arquivos são abertos.

Fonte: do autor.

No Quadro 17 apresenta-se o caso de uso Aplicar filtro na listagem de arquivos:

Quadro 17 - Descrição do caso de uso Aplicar filtro na listagem de arquivos

UC08	Aplicar filtro na listagem de arquivos
Descrição	Depois que foi feito o levantamento dos arquivos, o sistema permitirá que o usuário aplique filtro nesses arquivos. Para caso o usuário queira visualizar apenas os arquivos de determinado nome, extensão, data de modificação ou tamanho.
Ator	Usuário
Pré-condição	Ter algum arquivo na lista de arquivos.
Fluxo principal	Usuário abre a tela de filtros. Usuário preenche as informações com os filtros que deseja aplicar. (nome, extensão, data de modificação e tamanho). Sistema valida as informações. Sistema aplica o filtro na lista de arquivos.
Cenário – Exceção	Se, no passo 3, o sistema identificar algum inconsistência no preenchimento dos filtros, exibirá mensagem para o usuário corrigir o preenchimento.
Pós-condição	A lista de arquivos será atualizada e só serão mostrados os arquivos que se enquadram nos filtros aplicados.

Fonte: do autor.

No Quadro 18 apresenta-se o caso de uso Configurar agendamento de envio de arquivos existentes dos diretórios:

Quadro 18 - Descrição do caso de uso Configurar agendamento de envio de arquivos existentes dos diretórios

UC09	Configurar agendamento de envio de arquivos existentes nos diretórios
Descrição	O usuário poderá configurar um agendamento para que o sistema envie a listagem de arquivos encontrados com base nos diretórios configurados.
Ator	Usuário
Pré-condição	Ter um e-mail de remetente configurado, para ser utilizado no envio.
Fluxo principal	Usuário seleciona/digita os diretórios. Usuário digita a quantidade de maiores arquivos que deseja receber. Usuário seleciona se deseja receber os dados diariamente, semanalmente ou mensalmente. Conforme o que foi selecionado no passo 3, o usuário poderá selecionar qual o horário e qual dia deseja receber os dados. Usuário digita qual o e-mail de destino para receber as informações.
Cenário – Exceção	Se, no passo 1, o sistema identificar alguma inconsistência no preenchimento dos diretórios, exibirá mensagem para o usuário corrigir o preenchimento.
Pós-condição	Na data/hora configurada, a lista de arquivos será enviada para o e-mail destino.

Fonte: do autor.

No Quadro 19 apresenta-se o caso de uso Acessar local do arquivo:

Quadro 19 - Descrição do caso de uso Acessar local do arquivo

UC10	Acessar local do arquivo
Descrição	O sistema permitirá que o usuário “abra o local” de qualquer arquivo existente na lista.
Ator	Sistema
Pré-condição	Ter algum arquivo na lista de arquivos. Os arquivos selecionados ainda existirem no disco.
Fluxo principal	O usuário seleciona arquivo(s) da lista. O usuário clica com o botão direito e escolher a ação: abrir local do arquivo. O sistema verifica se os arquivos ainda existem e estão acessíveis. Sistema abre os diretórios onde os arquivos se encontram.
Cenário – Exceção	Se, no passo 2, o sistema identificar que foram selecionados mais de 10 arquivos, o sistema avisa que serão abertas muitas pasta e pergunta se o usuário quer continuar ou não.
Pós-condição	As pastas serão abertas.

Fonte: do autor.

No Quadro 20 apresenta-se o caso de uso Configurar alerta para tamanho livre dos dispositivos:

Quadro 20 - Descrição do caso de uso Configurar alerta para tamanho livre dos dispositivos

UC11	Configurar alerta para tamanho livre dos dispositivos
Descrição	O sistema permitirá que o usuário configure qual o tamanho livre mínimo permitido para cada dispositivo, para ser alertado quando esse tamanho estiver menos que o mínimo. Quando o sistema identificar que o tamanho livre está abaixo ao que foi configurado, será enviado um alerta por e-mail.
Ator	Usuário
Pré-condição	Ter alguma conta de e-mail configurada como remetente.
Fluxo principal	O usuário seleciona quais os dispositivos/discos que deseja. Usuário define um tamanho livre mínimo para o dispositivo/disco. Usuário digita qual o e-mail de destino para receber as informações.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui acesso ao disco/dispositivo, será exibido um alerta.
Pós-condição	O sistema verificará quando algum dispositivo se enquadra na configuração definida.

Fonte: do autor.

No Quadro 21 apresenta-se o caso de uso Enviar alerta de tamanho livre do dispositivo:

Quadro 21 - Descrição do caso de uso Enviar alerta de tamanho livre do dispositivo

UC12	Enviar alerta de tamanho livre do dispositivo.
Descrição	Caso tenha alguma configuração para envio de alerta de tamanho mínimo livre no dispositivo e o dispositivo tenha atingido o tamanho livre configurado, o sistema enviará um alerta ao e-mail configurado.
Ator	Sistema
Pré-condição	Ter configurado algum alerta para tamanho livre dos dispositivos. Ter alguma conta de e-mail configurada como remetente.
Fluxo principal	O sistema verifica se algum dispositivo atingiu o tamanho livre mínimo configurado. Se algum dispositivo atingiu o tamanho livre mínimo, é enviado um e-mail de alerta para o destinatário configurado.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui acesso ao dispositivo, o dispositivo é retirado da lista de configuração.
Pós-condição	Será enviado alerta ao e-mail do remetente configurado.

Fonte: do autor.

No Quadro 22 apresenta-se o caso de uso Configurar alerta para tamanho do diretório:

Quadro 22 - Descrição do caso de uso Configurar alerta para tamanho do diretório

UC13	Configurar alerta para tamanho do diretório
Descrição	O sistema permitirá que o usuário configure os diretórios e qual o tamanho máximo que cada diretório pode ter. Quando o sistema identificar que o tamanho máximo do diretório foi atingido, será enviado um alerta por e-mail.
Ator	Usuário
Pré-condição	Ter alguma conta de e-mail configurada como remetente.
Fluxo principal	O usuário seleciona os diretórios que deseja. Usuário define um tamanho máximo para cada diretório. Usuário digita qual o e-mail de destino para receber as informações.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui acesso ao diretório, será exibido um alerta.
Pós-condição	O sistema verificará quando os diretórios configurados atingirem os tamanhos máximos.

Fonte: do autor.

No Quadro 23 apresenta-se o caso de uso Enviar alerta de tamanho do diretório:

Quadro 23 - Descrição do caso de uso Enviar alerta de tamanho do diretório

UC14	Enviar alerta de tamanho do diretório.
Descrição	Caso tenha alguma configuração para envio de alerta de tamanho máximo do diretório e algum diretório tenha atingido o tamanho máximo, o sistema enviará um alerta ao e-mail configurado.
Ator	Sistema
Pré-condição	Ter configurado algum alerta para tamanho máximo dos diretórios. Ter alguma conta de e-mail configurada como remetente.
Fluxo principal	O sistema verifica se algum diretório atingiu o tamanho máximo configurado. Se algum diretório atingiu o tamanho máximo, é enviado um e-mail de alerta para o destinatário configurado.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui acesso ao diretório, o diretório será retirado da lista de configuração.
Pós-condição	Será enviado alerta ao e-mail do remetente configurado.

Fonte: do autor.

No Quadro 24 apresenta-se o caso de uso Armazenar tamanho dos dispositivos conectados:

Quadro 24 - Descrição do caso de uso Armazenar tamanho dos dispositivos conectados

UC15	Armazenar tamanho dos dispositivos conectados.
Descrição	Uma vez por dia o sistema fará o armazenamento do tamanho ocupado dos dispositivos/discos conectados.
Ator	Sistema
Fluxo principal	O sistema verifica os dispositivos/discos conectados. O sistema armazena o tamanho ocupado de cada dispositivo/disco.
Cenário – Exceção	Se, no passo 1, o sistema identificar que o usuário não possui acesso ao dispositivo/disco, não será armazenado o tamanho desse dispositivo.
Pós-condição	A informação do tamanho do disco/dispositivo será armazenada.

Fonte: do autor.

No Quadro 25 apresenta-se o caso de uso Fazer projeção de ocupação do espaço dos dispositivos:

Quadro 25 - Descrição do caso de uso Fazer projeção de ocupação do espaço dos dispositivos

UC16	Fazer projeção de ocupação do espaço dos dispositivos.
Descrição	Uma vez por dia o sistema fará uma projeção de quando o dispositivo/disco estará totalmente cheio e exibirá ao usuário. Essa projeção será feita com base nos dados históricos armazenados dos dispositivos e será mais correta com o uso contínuo da ferramenta.
Ator	Sistema
Pré-condição	Ter armazenado alguma informação histórica dos dispositivos.
Fluxo principal	Sistema verifica os dados históricos armazenados dos dispositivos. Sistema calcula aproximadamente a data de quando o dispositivo estará completamente cheio. Sistema exibe a informação da data aproximada na tela inicial da ferramenta.
Cenário – Exceção	Se, no passo 1, o sistema identificar que não existem informações históricas armazenadas, não continuará para os passos seguintes.
Pós-condição	A data aproximada de quando o dispositivo estará cheio será exibida na tela inicial da ferramenta.

Fonte: do autor.