

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Desenvolvimento de Aplicativos para a Bilhetagem Eletrônica em Trens

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação na disciplina
DAS 5511: Projeto de Fim de Curso*

Thiago Israel Ramalho Bacic

Florianópolis, Fevereiro de 2013

Desenvolvimento de Aplicativos para a Bilhetagem Eletrônica em Trens

Thiago Israel Ramalho Bacic

Esta monografia foi julgada no contexto da disciplina
DAS5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. Dr. Eng. Rômulo Silva de Oliveira

Assinatura do Orientador

Agradecimentos

Aos meus pais Ivan e Iasmirine e à minha irmã Karoline, sem eles nada teria sido possível.

À minha namorada Thaili, pela companhia nos momentos bons e pelo apoio nos difíceis.

Aos meus amigos, em especial para a Priscila e à galera de Aachen, que ajudaram a fazer de mim quem eu sou hoje.

Ao Arleston e ao Eng. Wislann, meus orientadores na Radix, e ao Prof. Rômulo, meu orientador na UFSC, pela ajuda que me deram para desenvolver este relatório.

Dr. Eng. Alberto Pavim, aos Engs. Rafael e Renato da Greylogix, à Prof. Analúcia e ao Prof. Armando do Labmetro, pelas oportunidades que me deram de estagiar, o que me proporcionou um grande crescimento profissional e pessoal.

A todos meus professores, pelo conhecimento que me foi passado.

Aos colegas na SuperVia, pela ajuda no meu PFC e pelas horas de descontração.

Ao Prof. Augusto, pela ajuda em optar pelo curso de automação e por todo auxílio que sempre prestou.

Resumo

Este relatório descreve o trabalho realizado de Julho de 2012 a Janeiro de 2013 na empresa Radix Engenharia e Software, em um projeto realizado na SuperVia (empresa que opera, desde 1º de novembro de 1998, o serviço de trens urbanos em 11 municípios da Região Metropolitana do Rio de Janeiro.). O trabalho foi realizado na área de desenvolvimento de software, no escopo da disciplina Projeto de Fim de Curso, do curso de Engenharia de Controle e Automação da Universidade Federal de Santa Catarina.

A Radix é uma empresa brasileira de projetos que atua nos ramos de Engenharia e de Tecnologia da Informação (TI). Criada em Abril de 2010, já recebeu os prêmios de Melhor Empresa para se Trabalhar no Rio de Janeiro e no Brasil, concedido pelo Instituto Great Place to Work. As principais atividades da Radix na SuperVia são garantir o funcionamento correto dos equipamentos de bilhetagem eletrônica e assegurar que os arquivos com registros de passagens sejam enviados à empresa de cartões RioCard, para que ela faça o ressarcimento à SuperVia.

O objetivo deste projeto foi desenvolver programas que auxiliassem e automatizassem alguns procedimentos no Setor de Bilhetagem Eletrônica da SuperVia. Entre eles estão aplicativos para (1) configurar validadores (equipamentos que controlam as catracas); (2) testar a comunicação dos equipamentos com os servidores; (3) verificar se o ressarcimento da RioCard está sendo feito corretamente; (4) separar um grande volume de arquivos de passagens, removendo duplicidades e encontrando corrompimentos, para solicitar o reprocessamento à RioCard; e (5) exibir indicadores de desempenho de equipamentos e de técnicos.

Para o desenvolvimento foram utilizadas as linguagens (1) C#, no ambiente de desenvolvimento Microsoft Visual Studio, utilizando a plataforma “.NET”, e (2) Java, no desenvolvimento de um aplicativo para Android. Os resultados obtidos foram: (1) uma grande redução do tempo de configuração de validadores, de verificação do ressarcimento e de análise dos dados de desempenho; (2) a possibilidade de analisar melhor a comunicação dos validadores; e (3) um retorno financeiro para a SuperVia, com a separação dos arquivos.

Abstract

This report describes the work developed from July 2012 till January 2013 at Radix Engineering and Software company, in a project for SuperVia (the company that operates, since November 1st, 1998, the urban train services in 11 municipalities of Rio de Janeiro Metropolitan Region). The work area was software development, in the scope of the “End of Course Project” discipline, from Control and Automation Engineering from the Federal University of Santa Catarina.

Radix is a Brazilian company that works with engineering and information technology. Founded on April 2010, it already received the title of Best Company to Work in Rio de Janeiro and in Brazil. The title was granted by the Great Place to Work Institute. Radix’s main activities in SuperVia are to ensure that all equipments for electronic ticketing are working fine, and to guarantee that all files containing transactions from passengers are send to the card company, RioCard, so that they can reimburse SuperVia.

The objective of this project was to develop programs to assist and automate certain procedures in the Electronic Ticketing Sector of SuperVia. Among them, the following applications were developed: (1) to configure validators (equipments that control the turnstiles); (2) to test the equipments’ communication with the servers; (3) verify if the reimbursement was successful; (4) to separate a huge amount of transaction files, removing duplicities and finding corrupt ones, to request the reprocessing to RioCard; and (5) to show performance indicators of equipments and technicians.

To develop the applications, the following programming languages were used: (1) C#, on Microsoft Visual Studio development environment, using the “.NET” Framework, and (2) Java, to develop the Android application. The results achieved in this project were: (1) a great reduction of time spend to configure validators, to check the reimbursement and to analyze performance data; (2) the possibility to analyze the validator’s communication; and (3) a financial return for SuperVia, by separating the transaction files.

Sumário

Agradecimentos.....	3
Resumo	4
Abstract	5
Sumário	6
Capítulo 1: Introdução	8
1.1: Motivação e definição do problema.....	8
1.2: Empresa do projeto: Radix Engenharia e Software	9
1.3: Empresa cliente: SuperVia.....	9
1.4: Problemas e soluções propostas	10
1.5: Organização deste relatório	11
Capítulo 2: Setor de Bilhetagem Eletrônica	13
2.1: Equipamentos	13
2.2: Suporte à Bilhetagem Eletrônica.....	17
2.3: Homologação.....	20
2.4: Comunicação e Ressarcimento	20
2.5: Importação	22
2.6: RAT-Site	23
Capítulo 3: Descrição dos problemas no SBE	26
3.1: Configuração de validadores.....	26
3.2: Ressarcimento	28
3.3: Análise da comunicação	32
3.4: Reenvio de arquivos Bin	33
3.5: Análise de indicadores do SBE	33
3.6: Considerações finais.....	35

Capítulo 4: Especificação dos novos aplicativos	36
4.1: Configuração de validadores: ValparaMod.exe	36
4.2: Ressarcimento	37
4.3: Ferramentas análise de comunicação	39
4.4: Separação de arquivos	39
4.5: Indicadores do SBE	40
4.6: Considerações finais.....	41
Capítulo 5: Desenvolvimento dos novos aplicativos.....	42
5.1: Processo de desenvolvimento	42
5.2: Ferramentas para o desenvolvimento.....	43
5.3: Configurador do Valparam	45
5.4: Ressarcimento	49
5.5: Ferramentas análise de comunicação	63
5.6: Separador de arquivos bins	67
5.7: Indicadores do SBE	70
5.8: Considerações finais.....	74
Capítulo 6: Resultados	75
6.1: Configuração de validadores.....	75
6.2: Ressarcimento	75
6.3: Ferramentas de análise de comunicação	76
6.4: Separador de arquivos bins	77
6.5: Indicadores do SBE	77
Capítulo 7: Conclusões e Perspectivas	79
Bibliografia:.....	81

Capítulo 1: Introdução

1.1: Motivação e definição do problema

Um dos acontecimentos mais marcantes das últimas décadas, que alterou de forma drástica a vida de grande parte da população mundial, foi o advento da informática. Neste momento histórico, conhecido como Terceira Revolução Industrial, diversas tarefas que sempre foram realizadas por seres humanos estão sendo desenvolvidas, de forma mais ágil e confiável, por sistemas computacionais.

A capacidade de processamento e de armazenamento de dados de computadores possibilita aos programas realizarem, em poucos segundos, tarefas que um ser humano poderia levar meses para concluir. Muitas dessas tarefas são realizadas utilizando um procedimento padrão e repetitivo, o que pode torná-la tediosa. Um “software” tem a capacidade de evitar a ocorrência de diversos erros que seres humanos não conseguiriam, por causa de suas limitações.

Devido a todas suas vantagens, a utilização de sistemas de informação tem crescido consideravelmente, em empresas de todos os ramos. Na SuperVia, empresa responsável pelos trens do Rio de Janeiro, não é diferente.

Este projeto de fim de curso se encaixa no escopo de um projeto maior desenvolvido pela empresa Radix Engenharia e Software para a SuperVia. A Radix foi contratada pela SuperVia para gerenciar todos os procedimentos relativos à bilhetagem eletrônica da empresa.

Neste projeto de fim de curso, diversos aplicativos foram desenvolvidos para auxiliar os funcionários do Setor de Bilhetagem Eletrônica da SuperVia (SBE) no desempenho de suas tarefas diárias. As ferramentas criadas visam a redução de erros e do tempo de execução de várias tarefas realizadas pelos funcionários do SBE.

Entre essas tarefas destacam-se a configuração de equipamentos, a análise de falhas e o controle do recebimento de dinheiro da empresa de cartões eletrônicos.

1.2: Empresa do projeto: Radix Engenharia e Software

A Radix é uma empresa brasileira que atua nos ramos de engenharia e tecnologia da informação. Foi fundada em abril de 2010, com sede no Rio de Janeiro e um escritório em Belo Horizonte [1].

No ano de 2011, recebeu as certificações ISO 9001, ISO 14001 e OHSAS 18001 [1]. No mesmo ano, foi eleita a 2ª Melhor Empresa de Tecnologia para se Trabalhar no Brasil, a Melhor Empresa para se Trabalhar no estado do Rio de Janeiro e a Melhor Empresa para se Trabalhar no Brasil – pequenas e médias empresas, pelo Great Place to Work Institute [2]. Em 2012, foi eleita a 4ª Melhor Empresa para se Trabalhar no Brasil e a Melhor Empresa para se Trabalhar no Rio de Janeiro [2].

A empresa oferece a seus funcionários uma grande variedade de atividades, como massagem, coral, futebol, vôlei, sorteio de ingressos para eventos culturais e esportivos, trilhas, entre outros.

Ela patrocina eventos esportivos e equipes acadêmicas de várias universidades brasileiras, possui projetos com cunho social e faz campanhas internas para doação de sangue, roupa e alimentos.

A alta qualidade dos serviços prestados pela Radix tem feito a empresa conseguir projetos com empresas de diferentes portes, tanto no Brasil quanto no exterior. Alguns dos clientes para os quais a Radix já prestou serviço são Petrobras, Rede Globo, Companhia Siderúrgica Nacional, Queiroz Galvão e Aker Solutions [3].

1.3: Empresa cliente: SuperVia

A SuperVia é a empresa vencedora da licitação para operação e manutenção da malha ferroviária da região metropolitana do Rio de Janeiro [4]. Iniciou as operações em 1998, tendo a concessão de 25 anos, renovável por igual período de tempo. Transporta quase 10 milhões de pessoas por mês, pelas 89 estações em operação nos treze municípios localizados na malha. É também responsável pelos teleféricos do Complexo do Alemão.

O Setor de Bilhetagem Eletrônica (SBE), onde a Radix atua, é responsável pela manutenção dos equipamentos de bilhetagem eletrônica, pela homologação dos novos “firmwares” para os equipamentos e pelo controle do ressarcimento da empresa de cartões à SuperVia.

Além disso, um funcionário da Radix é responsável por todo o desenvolvimento de software para o SBE. Nos seis meses deste Projeto de Fim de Curso, o autor deste relatório teve essa função. Ele realizou melhorias em softwares já existentes e desenvolveu programas novos, que serão descritos neste relatório.

1.4: Problemas e soluções propostas

A seguir, são descritos problemas e soluções propostas. Estas soluções foram implementadas pelo autor deste trabalho e são descritas mais detalhadamente nos capítulos seguintes.

1.4.1: Configuração de equipamentos

A configuração dos equipamentos era feita utilizando um programa pouco amigável ao usuário, além de apresentar diversos defeitos. Um software novo foi desenvolvido para desempenhar esta função.

1.4.2: Comunicação

Todos os dias, a comunicação de vários equipamentos com os servidores do SBE não é concluída com êxito. Para analisar isso, foram criados alguns aplicativos que testam a comunicação, manualmente ou automaticamente, e que fornecem informações claras.

1.4.3: Ressarcimento

Quando um passageiro utiliza um cartão de integração da empresa RioCard nas catracas da SuperVia, a transação é registrada em um arquivo. Estes arquivos são coletados e enviados à RioCard, que deve ressarcir à SuperVia.

Pela Lei do Bilhete Único, a integração entre dois meios de transporte no Rio de Janeiro tem um limite de valor a ser cobrado. O segundo meio de transporte

utilizado em uma integração deverá cobrar um valor reduzido, caso exceda o limite. A diferença é ressarcida pelo Governo.

Para realizar a verificação da falta de arquivos e a comparação entre o que foi pago e o que deveria ter sido pago, alguns aplicativos foram desenvolvidos.

1.4.4: Reprocessamento de arquivos

A SuperVia solicitou à RioCard que todos os arquivos desde 2008, quando o atual sistema começou a operar, fossem reprocessados, para verificar a existência de alguma pendência nos pagamentos. O novo sistema da RioCard também fará o pagamento de arquivos corrompidos. Para conseguir juntar todos estes arquivos, separando os corrompidos, foi criado um aplicativo que, mesmo com um grande volume de dados, consegue manter o uso de memória em um nível baixo.

1.4.5: Indicadores do SBE

Um dos sistemas já operantes no SBE possui uma quantidade elevada de informações interessantes em seu banco de dados, mas que não são utilizadas da melhor forma.

Foi criado um software que mostra indicadores de desempenho de equipamentos e de técnicos, além de relacionar problemas com suas possíveis causas, mostrando possíveis soluções.

1.5: Organização deste relatório

Este relatório é composto por 7 capítulos, sendo este primeiro a introdução.

O Capítulo 2: descreve conceitos, procedimentos, softwares implementados antes deste Projeto de Fim de Curso e tarefas de outros membros da equipe.

No Capítulo 3: são expostos todos os problemas que foram atacados pelo autor deste relatório neste Projeto de Fim de Curso.

O capítulo seguinte (Capítulo 4:) explica conceitualmente como os novos softwares criados pelo autor devem operar, de modo a resolver os problemas descritos no Capítulo 3:.

O Capítulo 5: mostra como o autor deste trabalho implementou os novos softwares. São apresentadas descrições, diagramas e telas da interface gráfica para facilitar o entendimento.

No Capítulo 6: são apresentados os resultados obtidos com a implementação dos novos programas.

O último capítulo (Capítulo 7:) contém as conclusões que o autor obteve neste Projeto de Fim de Curso. É mostrada uma síntese do que foi feito e dos resultados obtidos, bem como propostas para trabalhos futuros.

Capítulo 2: **Setor de Bilhetagem Eletrônica**

O Setor de Bilhetagem Eletrônica (SBE) é responsável por garantir o correto funcionamento dos equipamentos de bilhetagem eletrônica da SuperVia e por verificar que o ressarcimento da RioCard e do Governo está sendo feito corretamente. Suas funções são de extrema importância para a empresa de trens, por tratarem de uma parcela considerável de sua entrada de dinheiro.

Para que todos os procedimentos realizados no SBE sejam feitos de forma mais rápida e com maior confiabilidade, alguns softwares auxiliam na realização de diversas tarefas.

A seguir serão descritos conceitos e processos do SBE, que serão importantes para entender o contexto do projeto.

2.1: Equipamentos

A bilhetagem eletrônica apresenta diversos equipamentos que funcionam em conjunto para que os clientes comprem a passagem, consigam viajar, e para garantir que o dinheiro entre na conta da SuperVia.

É feito um controle de entrada e saída de equipamentos no SBE, sendo possível verificar onde ele se encontra e seu estado de funcionamento.

2.1.1: Catraca

As catracas são equipamentos que liberam a passagem apenas para pessoas com permissão de acesso. No caso dos trens, esta permissão é obtida por quem tiver crédito em um cartão inteligente.

2.1.2: Cartão inteligente

O cartão inteligente é um cartão magnético onde são inseridos os créditos para que um passageiro possa ter acesso aos trens da SuperVia. Os tipos de cartão que podem ser utilizados na SuperVia são:

1. Unipass (Figura 2.1): cartão unitário, pode ser utilizado apenas uma vez.



Figura 2.1: Cartão UniPass (Fonte: http://www.supervia.com.br/tarifas_cartoes.php)

2. Multipass (Figura 2.2): cartão múltiplo, pode ser utilizado várias vezes. É interessante para viajar em grupo, sendo que um único cartão pode ser utilizado por todos, e para quem viaja frequentemente.



Figura 2.2: Cartão Multipass (Fonte: http://www.supervia.com.br/tarifas_cartoes.php)

3. RioCard: cartão de integração entre diferentes meios de transporte. Seu uso é exclusivo para o deslocamento casa-trabalho-casa. É o principal tipo de cartão utilizado nos trabalhos realizados neste projeto. Existem 4 tipos de cartões de integração:
 - a. Bilhete Único Estadual (BUE, Figura 2.3): cartão unitário utilizado para integração entre dois meios de transporte, em diferentes cidades na região da grande Rio de Janeiro.



Figura 2.3: Bilhete estadual unitário (Fonte: <https://www.cartaoriocard.com.br/vt/visitante/sobre/Sobre.do>)

- b. Bilhete Único Carioca (BU, Figura 2.4): cartão unitário utilizado na integração entre dois meios de transporte dentro da cidade do Rio de Janeiro. Possui um valor abaixo do BUE.



Figura 2.4: Bilhete unitário carioca (Fonte: <https://www.cartaoriocard.com.br/vt/visitante/sobre/Sobre.do>)

- c. Vale Transporte Convencional (VTC, Figura 2.5): cartão pessoal recarregável que pode ser utilizado tanto para integração estadual quanto municipal.



Figura 2.5: Cartão VTC (Fonte: <https://www.cartaoriocard.com.br/vt/visitante/sobre/vtc.jsp>)

- d. Vale Transporte Rápido (VTR, Figura 2.6): cartão pessoal não recarregável que pode ser utilizado tanto para integração estadual quanto municipal.



Figura 2.6: Cartão VTR (Fonte: <https://www.cartaoriocard.com.br/vt/visitante/sobre/Sobre.do>)

- e. Cartão Expresso (Figura 2.7): cartão múltiplo (pode ser usado por um grupo) para se usar em um meio de transporte.



Figura 2.7: Cartão Expresso (Fonte: <https://www.cartaoriocard.com.br/vt/visitante/sobre/Sobre.do>)

2.1.3: Ponto-de-Venda (PDV)

PDVs são equipamentos utilizados para inserir crédito em cartões. O bilheteiro encosta o cartão no local adequado no PDV e digita o crédito a ser inserido. Todas as transações de crédito inserido em cartões são armazenadas em um arquivo. Para que isso ocorra, é necessário “abrir a jornada do PDV”, o que colocará o equipamento em operação e criará um novo arquivo de transações. No final do dia esta jornada deve ser fechada, finalizando o arquivo, o que adicionará uma assinatura no final do arquivo com o total de transações. A jornada também poderá ser fechada para realizar a manutenção de um equipamento, e reaberta quando o problema for corrigido.

2.1.4: Validador

Validador é um equipamento, localizado na parte superior de uma catraca, que efetua o débito do cartão e libera a passagem na catraca. Há algumas placas que realizam processamento para diferentes tipos de cartão, como é o caso da placa “blackbox”, fornecida pela RioCard para ler os seus cartões.

Para que ocorra uma transação com um cartão reutilizável (Multipass ou RioCard), o cartão deve ser aproximado do validador. Com os unitários, o cartão é inserido em uma fenda, onde a leitura do cartão ocorrerá. A passagem será liberada e o cartão será armazenado num compartimento do validador para uso posterior. Os procedimentos de abertura e fechamento de jornada, e de registro de transações em arquivos, também são realizados nos validadores. Para isso, é necessário utilizar um

cartão “abre-fecha”, que armazenará os arquivos de transações. No final do dia, após o último fechamento, o bilheteiro (funcionário responsável pela venda de bilhetes e pela abertura e fechamento de jornadas de equipamentos) leva o cartão “abre-fecha” a um PDV na estação, que irá imprimir uma filipeta (papel com o total de cada tipo de transação). Os valores dos diferentes tipos de transações são somados pelo bilheteiro e preenchidos no Recibo de Arrecadação por Turno (RAT), que será entregue ao Setor de Arrecadação.

2.1.5: Totem

É a carcaça metálica onde são instalados a catraca e o validador.

2.2: Suporte à Bilhetagem Eletrônica

O SBE é responsável por dar suporte para os equipamentos citados anteriormente. O SABE (Sistema de Atendimento à Bilhetagem Eletrônica), é o sistema implantado na SuperVia, pela Radix, para realizar o controle do suporte. Ele foi desenvolvido em PHP, baseado no software livre “Ocomon” (utilizado para realizar o cadastro, acompanhamento, controle e consulta de ocorrências [5]), com algumas adaptações feitas para se adequar às necessidades do setor.

Um dos módulos do SABE, chamado de “InvMon”, é onde ocorre o cadastramento dos equipamentos e a atualização do local onde eles se encontram (a estação onde ele está operando).

O SABE se baseia no conceito de “chamado” (ou “ocorrência”), que é a denominação dada a problemas de qualquer natureza, relativos à bilhetagem eletrônica, que devem ser resolvidos pelo SBE.

Quando ocorre um problema em um validador ou em um PDV em qualquer das estações da SuperVia, um funcionário da respectiva estação deve acessar o SABE pela “intranet” da SuperVia e abrir um chamado, especificando o equipamento e o erro que está ocorrendo.

Dependendo do nível de urgência, o SBE tem um determinado tempo para resolver o problema, conforme acordado na SLA (“Service-Level Agreement”) entre a Radix e a SuperVia.

Os níveis de prioridade são:

1. Baixo (30 horas): quando o problema não impede o uso normal do equipamento. Ex.: problemas no visor do validador
2. Emergencial (6 horas): Quando todos os validadores da estação estão com defeito. Neste caso, não havendo validadores em operação, o débito dos cartões não ocorrerá. Então, os portões da SuperVia são abertos e os passageiros entram na estação de graça. Isto representa uma grande perda para a SuperVia. Este tipo de chamado é mais comum em estações que possuem apenas um validador, onde os chamados serão sempre emergenciais.
3. Médio (24 horas)/Alto(12 horas): caso o chamado não seja de prioridade baixa nem emergencial, é feito um cálculo para se calcular a variável de prioridade (W). Este cálculo é feito multiplicando a distância até o local, o fluxo de passageiros no horário de abertura de chamado e o percentual de equipamentos defeituosos. Se o valor de W estiver acima de um nível estipulado, a prioridade será alta. Senão, ela será média.

Quando o chamado chega no SBE, delega-se a responsabilidade para se atender o chamado a um dos técnicos.

Inicialmente, dependendo do problema que estiver ocorrendo, o técnico pode tentar resolvê-lo por telefone, conversando com os funcionários da estação. Caso não seja possível, configura-se um validador ou PDV do estoque para operar em uma determinada posição em uma estação e o técnico efetuará a troca. O traslado até as estações é feito pelos próprios trens da SuperVia, sendo que a viagem para as estações mais distantes pode levar horas.

A Figura 2.8 mostra algumas posições de validadores na Central do Brasil:



Figura 2.8: Validadores da Central do Brasil

Quando o equipamento retorna ao SBE, o técnico em eletrônica fará uma análise da causa do problema e irá corrigi-lo. Ele verifica se há problema com fios ou se alguma placa está com defeito. Há vários fios e placas em estoque, para que o reparo seja feito o mais rápido possível. Quando o equipamento estiver funcionando corretamente, ele retorna ao estoque do SBE, e estará pronto para ser configurado para outra estação.

A Figura 2.9 mostra a tela de chamados pendentes no SBE:

SABE
Sistema de Atendimento à Bilhetagem Eletrônica

supervia
Trens Urbanos

Home | **Ocorrências** | Inventário | Admin | Usuário: admin | Logoff

Ocorrências

- Início
- Abrir Chamado
- Busca Rápida
- Busca Detalhada
- Mural
- Relatórios

SABE - Sistema de Atendimento à Bilhetagem Eletrônica - Módulo de Ocorrências Segunda-feira, 21

Não existem ocorrências pendentes para o usuário admin.

Existem 5 ocorrências pendentes no sistema. Ordenar por área e número do chamado (padrão).

N.º / Área	Equipamento	Problema	Contato Telefone	Local	Responsável	Status Prioridade	Tempo Decorrido	SLA	Resposta
11819 SBE	VALIDADOR	Não carrega Blackbox	Douglas - 24097 9210	Japeri Validador 1 RAMAL JAPERI	Marcos - Ata Service	Aguardando atendimento Prioridade Média	12:12	24h	●
11826 SBE	PDV	Erro de Impressão	Bil. Daniela, mat. 24606 9745	Madureira PDV 42 RAMAL DEODORO	Danielle Santana da Silva	Aguardando atendimento Prioridade Alta	8:11	12h	●
11833 SBE	WOLPAC	Catraca Parando fora do ponto	Bil. Valéria, mat. 25291 9492	Cascadura Catraca 3 RAMAL DEODORO	Danielle Santana da Silva	Aguardando atendimento Prioridade Alta	2:29	12h	●
11834 SBE	VALIDADOR	Não Engolindo cartão	Bil. Carla, mat. 25807 9745	Madureira Validador 2 RAMAL DEODORO	Danielle Santana da Silva	Aguardando atendimento Prioridade Alta	0:51	12h	●
11835 SBE	VALIDADOR	Validador apagado	Bil. Rosalina, mat. 24336 9863	Honório Gurgel Validador 1 RAMAL BELFORD ROXO	Danielle Santana da Silva	Aguardando atendimento Prioridade Emergencial	0:30	6h	●

Figura 2.9: Tela de chamados abertos no SABE

2.3: Homologação

Quando o fabricante dos equipamentos lança uma nova versão do “firmware” para os validadores e para os PDVs, a equipe de homologação do SBE é responsável por realizar os testes para verificar se existem problemas que possam atrapalhar o funcionamento correto deles. Um “firmware” com defeito que seja instalado em todos os equipamentos da bilhetagem eletrônica pode trazer um grande prejuízo para a SuperVia.

2.4: Comunicação e Ressarcimento

Quando um passageiro compra cartões multipass ou unipass, o dinheiro entra diretamente para a SuperVia. Por outro lado, quando a recarga é feita no RioCard, o dinheiro não entrará até que a SuperVia solicite. Para que isso ocorra, é necessário manter um registro de todas as transações com RioCard nos validadores da SuperVia.

Esses registros são gravados em arquivos nos validadores e enviados por comunicação GPRS (General Packet Radio Service). O GPRS permite o acesso via rádio para estações móveis, como celulares e “laptops”, e a comutação de pacotes, que são enviados em uma rede compartilhada, entrando em uma fila de envio quando passam por um nodo [6].

Para que as mensagens cheguem nos servidores, é utilizada uma APN (Access Point Name), que é uma ligação entre uma rede de celular e uma rede de computadores.

O destino das mensagens são os 3 servidores de coleta (nucsrv28, nucsrv29 e nucsrv30), que criam uma “thread” (tarefa que pode ser realizada concorrentemente com outras). Antigamente, havia apenas um servidor de coleta (o nucsrv28). Porém, verificou-se que havia uma sobrecarga de validadores tentando comunicar com o servidor, o que prejudicava a comunicação.

Embora a divisão em três servidores tenha melhorado a comunicação, problemas na rede, nos servidores, no sinal da operadora de telefonia móvel ou nos próprios validadores podem impedir que algum arquivo chegue.

Para obter os arquivos que não foram enviados, é designado aos atendentes (funcionários responsáveis pela coleta dos arquivos), que eles se desloquem até a estação para obter os dados.

A obtenção do arquivo pelos atendentes pode ser feita de 3 formas:

1. Forçar manualmente o validador a comunicar. Isso é feito nos equipamentos que não enviaram arquivos no dia anterior.
2. Transferir os arquivos para um pendrive e levá-los ao SBE. Este procedimento é realizado em validadores onde os dados não foram enviados pela comunicação forçada. Os arquivos com 2 dias de atraso são coletados dessa forma.
3. Trocar o cartão SD do validador e levá-lo ao SBE. Esta troca é feita quando o atraso chega a 3 dias, pois nem a comunicação forçada nem o uso dos pendrives foram efetivos na obtenção de todos os arquivos.

A menos que o SD esteja corrompido, a obtenção dos arquivos por este método é garantida.

Os arquivos que chegam ao SBE são manipulados pelo aplicativo SAGA (Sistema de Apoio ao Gerenciamento de Arquivos), desenvolvido pela Radix, que adiciona um registro dos arquivos em um banco de dados, no momento do recebimento.

Diariamente, o SAGA envia os arquivos à RioCard e o pagamento é feito. A RioCard disponibiliza todos os dias, em um site, uma lista de todos os pagamentos feitos. O acesso a esse site é feito utilizando um nome de usuário e uma senha. É necessário digitar um “captcha” (teste de Turing público completamente automatizado para diferenciação de computadores e humanos) [7], o que impossibilita a automatização da obtenção dos dados de pagamento.

Para acompanhar a comunicação diária, um programa com interface Web chamado SOUBE (Sistema de Operações e Utilidades para a Bilhetagem Eletrônica), também desenvolvido pela Radix, mostra uma tabela listando os validadores e PDVs que comunicaram, dia-a-dia. Um gráfico também é gerado, mostrando a porcentagem de equipamentos que comunicaram.

2.5: Importação

Os arquivos que foram registrados no banco de dados pelo SAGA (com exceção dos arquivos de transações da RioCard), passam por um processo de importação. Um programa chamado “Interface”, desenvolvido pelo fabricante dos equipamentos, importa os valores das transações para o banco de dados da SuperVia, separando arquivos corrompidos ou com problemas de leitura (o que geralmente ocorre quando há erro de data ou hora).

Outro programa no SBE chamado AGUIA (Aplicativo de Gerenciamento Único de Importação de Arquivos) atualiza o banco de dados do SAGA, informando o “status de importação” (lido, corrompido ou falha de processamento). O AGUIA agrupa os arquivos com problema e os envia para o fabricante, que irá corrigi-los e enviá-los de volta ao SBE, para que sejam importados corretamente.

2.6: RAT-Site

O setor de Arrecadação da SuperVia é responsável por digitar os dados dos RATs em um programa, que popula um banco de dados. Os dados digitados são confrontados com os dados de pagamento utilizando um programa chamado SBE_Diff. O resultado é transferido para uma planilha do Excel que converte os dados em uma tabela, listando a diferença entre o que foi digitado e que foi pago para todos os validadores, dia-a-dia, no período especificado.

Havendo diferenças, um funcionário deve verificar o que provocou o erro. Ele deve pintar as células onde há diferenças de acordo com o respectivo erro, conforme a Tabela 2.1.

Cor	Problema	Ação a ser tomada
Azul Escuro	O arquivo chegou no SBE	Pedir a coleta do arquivo a um atendente
Cinza	O arquivo ou o cartão SD está corrompido	Enviar SD à empresa fabricante dos validadores
Verde	O arquivo foi enviado mas não foi pago	Entrar em contato com RioCard e reenviar o arquivo
Amarelo	O Setor de Arrecadação digitou o valor incorretamente	Pedir para que a Arrecadação corrija a digitação
Vermelho	Arquivo no SBE, mas não enviado à RioCard	Aguardar o pagamento que deve ser realizado no dia seguinte

Tabela 2.1: Cores do RAT-Site

Há uma formatação condicional na planilha, que colore todas as células onde o valor é menor ou igual a zero com a cor azul claro. Desta forma, as células positivas (onde supostamente falta pagamento de algum arquivo) ficam em evidência, na cor branca, ficando mais fácil localizá-las para colori-las. A Figura 2.10 mostra um exemplo da planilha do RAT-SITE:

Estacao	Estacao Site	Carro_seq	15/12/12	16/12/12	17/12/12	18/12/12	19/12/12
44	NOVA IGUAÇU	1	0	0	0	0	-40
		2	0	0	0	0	-50
		3	0	0	0	0	-10
		4	0	0	0	0	-20
		5	0	0	0	0	-30
		6	-18	0	18	0	-40
		7	-20	0	20	0	-50
		8	-5	0	5	0	-10
45	COMENDADOR SOARES	1	0	0	0	0	-20
		2	0	0	0	0	-30
		3	0	0	2	10	-40
		4	0	0	0	0	-50
		5	0	0	0	0	-10
46	AUSTIN	1	0	0	0	0	-20
		2	0	0	0	0	-30
		3	0	0	0	0	-40
		4	20	0	0	0	-50
		5	0	0	0	0	-10
		6	0	0	0	0	-20
47	QUEIMADOS	1	0	0	0	0	-30
		2	0	0	0	0	-40
		3	0	0	-30	0	-50
		4	0	0	0	0	-10
		5	0	0	0	0	-20
		6	0	0	0	0	-30
		7	0	0	0	0	-40
		8	0	0	0	66	-50
		9	0	0	0	0	-10
49	Japeri	1	0	0	0	0	-40
		2	0	0	0	0	-50
		3	0	0	0	0	-10
		4	0	0	30	0	-20
52	Jacarezinho	1	0	0	0	0	-10
		2	0	0	0	0	-20
53	Del Castilho	1	0	0	0	0	-30
54	Pilares	1	0	0	0	0	-40
55	Tomas Coelho	1	0	0	0	0	-50
56	Cavalcante	1	0	0	0	40	-10
57	M.de Madureira	1	0	0	0	0	-20
		2	0	0	0	0	-30

Figura 2.10: Tabela do RAT-Site:

A verificação é feita analisando todas as células positivas na planilha. A análise das negativas não é necessária, pois nunca é pago um valor acima do devido. As células negativas representam erros de digitação ou arquivos pagos que ainda não foram digitados.

Inicialmente, verifica-se a existência de arquivos do validador na data do erro. Havendo falta de arquivo, pinta-se a célula de azul. Caso não haja falta, deve-se abrir os arquivos no programa ListaArqLog, da Riocard. Se não for possível abrir algum arquivo, o que indica corrompimento, pinta-se a célula de cinza. Se os arquivos estiverem íntegros, soma-se o número de transações registradas em cada um deles. Se o valor total pago for igual ao valor dos arquivos, o pagamento foi feito corretamente. Então, houve erro de digitação, e a célula é pintada de amarelo.

Se a diferença entre o valor digitado e pago for igual ao valor de um arquivo que ainda não foi enviado, mas está disponível para envio, a célula é pintada de vermelho. Por fim, se o arquivo foi enviado à RioCard mas ainda não foi pago, pinta-se de verde. Neste caso, é necessário contatar a RioCard e reenviar o arquivo.

Em certas ocasiões, um arquivo com transações é pago como se estivesse zerado. A célula também é pintada de verde e um chamado deve ser aberto na central de atendimento da RioCard, para que o pagamento correto seja feito.

Pela SLA, não pode haver validadores sem arquivos dos últimos 3 dias (com exceção dos fins de semana). Esta planilha é mostrada nas reuniões semanais entre a Radix e a SuperVia, para mostrar que as tarefas estão sendo executadas corretamente.

No período deste Projeto de Fim de Curso, a verificação do ressarcimento pela RioCard ficou sob a responsabilidade do autor deste relatório.

Capítulo 3: Descrição dos problemas no SBE

O SBE possuía diversas ferramentas para auxiliar nas tarefas de bilhetagem. Porém, há muitos pontos a serem melhorados. Foram criados novos sistemas para reduzir erros e tempo de execução de tarefas, utilizando a linguagem C#, no Visual Studio, e Java (ver seção 5.2:). A seguir, são descritos os problemas solucionados pelo autor deste relatório. Serão mostrados, nos próximos capítulos, a descrição conceitual, a implementação e os resultados associados a estes problemas.

3.1: Configuração de validadores

O arquivo “Valparam.ini” é utilizado para a configuração dos validadores. Nele estão contidas diversas informações para que os validadores operem corretamente, como mostrado na Figura 3.1:

Linha no Valparam	Definição
[800797594]	CPG: o número do validador físico (o equipamento em si)
CARRO = 574	Número do validador no sistema (associado a uma posição em uma estação. Ex.: Gramacho 7)
LINHA = 8654	O ramal da estação
APN = *****	APN da SuperVia
LOGIN_APN = *****	Usuário da APN
SENHA_APN = *****	Senha da APN
SERVIDOR_CONEXAO = **.*.*.*.*	Servidor com o qual o validador comunica
PORTA_CONEXAO = *****	A porta que é aberta no servidor de conexão
LOGIN_SERVIDOR = *****	Login do servidor
SENHA_SERVIDOR = *****	Senha do servidor
SERVIDOR_TEMPO = **.*.*.*.*	Servidor de onde o validador atualiza o horário (no caso, o nucsrv28)
PORTA_TEMPO = *****	A porta que é aberta no servidor de tempo
RAMAL = 20	O ramal da estação
ESTACAO = 79	O número da estação no sistema
IP_LOCAL = *.*.*.*.*.*	O endereço IP do validador
NETMASK = *.*.*.*.*.*	Máscara de rede, que informa qual parte do IP é da rede pública e qual é da rede local
GATEWAY = 0.0.0.0	Endereço do roteador que se comunica com a rede pública
LIMITE_USO = 30	Tempo, em minutos, entre usos consecutivos de um cartão vale transporte RioCard
SETA_DATA = 0	Nova data para o validador
SETA_HORA = 0	Nova hora para o validador
ATIVA_LOG_FETRANSPOR = NAO	Opção de ativar ou não um log do funcionamento

Figura 3.1: Parâmetros do Valparam

O arquivo Valparam (validador – parâmetro) possui estes 21 parâmetros configurados para todos os validadores. Caso haja um erro na formatação, como colchete, espaço ou quebra de linha em local indevido, a comunicação dos validadores com o SBE estará comprometida. Antigamente, este arquivo era alterado pelo bloco de notas, sendo que erros ocasionados por falha do operador travavam a comunicação.

Este procedimento foi substituído por um programa há alguns anos, onde era possível buscar um validador, informando o ramal, a estação e o validador. Embora o número de falhas tenha reduzido drasticamente, o programa ainda apresentava diversos erros que impediam o correto funcionamento do sistema.

Para que apenas um usuário acessasse o arquivo do Valparam, era criado um arquivo temporário que impedia que o programa fosse aberto em outro computador. Este arquivo era excluído ao fechar o programa. Porém, ocasionalmente, o arquivo não era removido, travando o acesso ao programa.

Outro problema que ocorria era o corrompimento do arquivo se o programa travasse ou se ocorresse uma falha na rede (o que não é incomum).

Além disso, o usuário encontrava mais problemas: era necessário memorizar o ramal onde a estação se encontrava, para buscar o validador; e os dados do parâmetro “Linha” e “Ramal” precisavam estar equivalentes, pois ambos se referem ao ramal.

Para solucionar estes problemas, pensou-se, inicialmente, em alterar o programa que já existia. Porém, o código fonte estava difícil de compreender e a interface gráfica de desenvolvimento estava muito confusa, como pode ser visto na Figura 3.2:

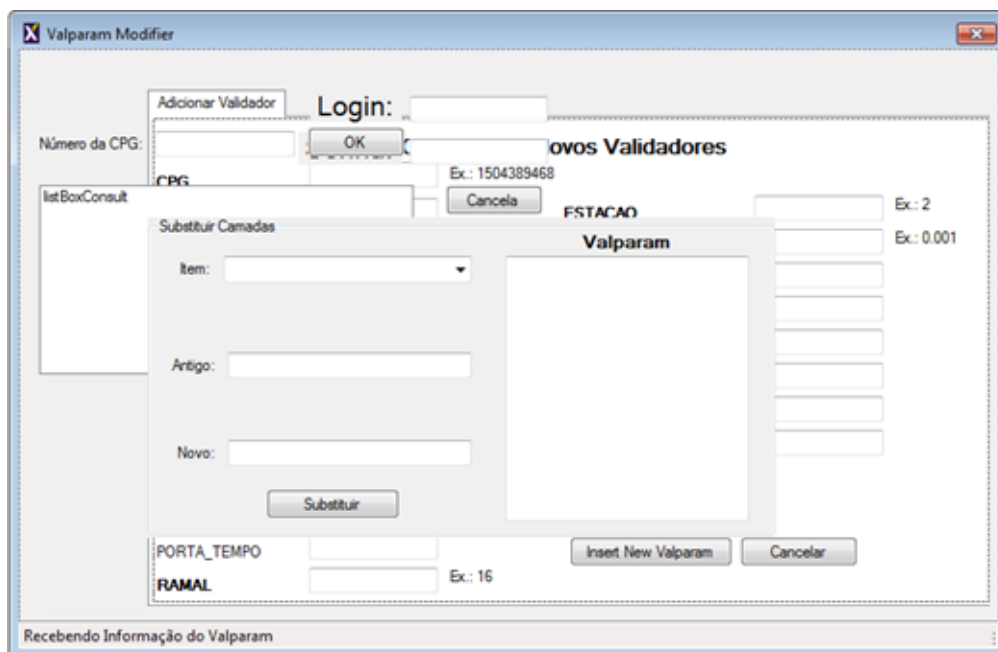


Figura 3.2: Tela de desenvolvimento do antigo modificador de valparam

Então, um novo programa, com uma interface mais amigável ao operador e uma maior robustez, foi desenvolvido.

Este novo software deveria ser capaz de evitar problemas ocasionados por acesso simultâneo, como descrito a seguir: se dois usuários acessarem o programa ao mesmo tempo, ambos terão a mesma versão do arquivo valparam. Porém, se um deles fizer uma alteração, ela não estará presente na versão do outro. Quando o segundo realizar sua alteração, a mudança feita pelo primeiro será perdida. Então, é necessário que o programa apresente um sistema de exclusão mútua, impedindo que mais de um usuário acesse o programa ao mesmo tempo.

3.2: Ressarcimento

Os problemas listados abaixo estão relacionados ao ressarcimento, tanto da RioCard quanto do Governo.

3.2.1: Listagem de arquivos faltantes

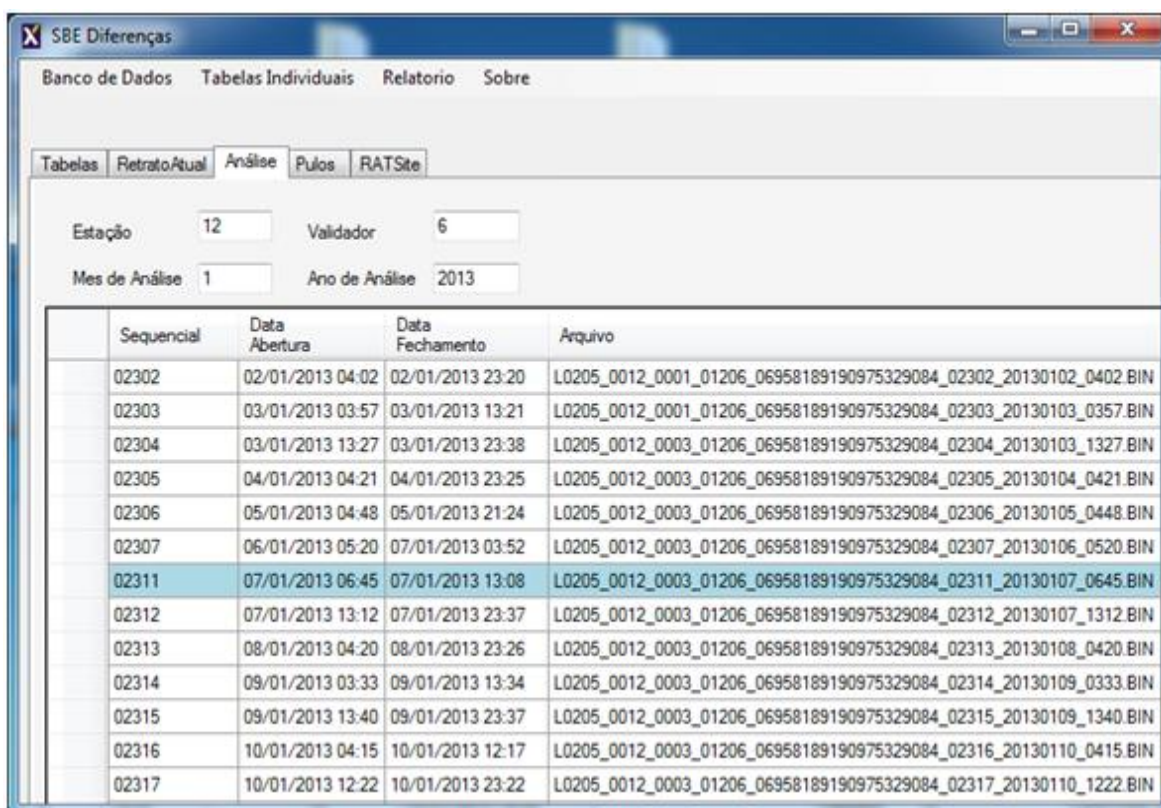
Um sistema já operante no SBE, o SOUBE (Sistema de Operações e Utilidades para a Bilhetagem Eletrônica), verifica diariamente se um validador comunicou ou não. Um e-mail contendo a lista de validadores que não comunicaram

é enviado automaticamente ao responsável pela coleta de arquivos. Ele irá informar aos atendentes de cada ramal, os validadores em que eles devem forçar a comunicação, obter os arquivos por pen-drive ou buscar o cartão SD.

Há duas informações importantes que este sistema não informa: (1) pulo sequencial em um validador (por exemplo, os arquivos de sequencial 1, 2 e 4 chegaram ao SBE, enquanto que o arquivo 3 não chegou); e (2) há quantos dias o validador não comunica.

O problema do pulo sequencial só era resolvido quando a planilha do RAT-Site era feita, ocorrendo uma diferença entre o valor pago e o valor digitado. Porém, era difícil distinguir o que era um pulo no sequencial e um erro de digitação. Para isso, era necessário utilizar o SBE_Diff para listar os arquivos de um validador. Dessa forma, era possível visualizar o pulo no sequencial.

Na Figura 3.3 observa-se a tela de busca no SBE_DIFF. Há um pulo no sequencial, indo do arquivo 2307 para o 2311. No caso, o pulo foi uma falha do equipamento, ou seja, os arquivos não existem.



The screenshot shows a software window titled "SBE Diferenças". It has a menu bar with "Banco de Dados", "Tabelas Individuais", "Relatorio", and "Sobre". Below the menu is a tabbed interface with "Tabelas", "RetratoAtual", "Análise", "Pulos", and "RATSite" tabs. The "Análise" tab is active. There are four input fields: "Estação" (12), "Validador" (6), "Mes de Análise" (1), and "Ano de Análise" (2013). Below these fields is a table with the following columns: "Sequencial", "Data Abertura", "Data Fechamento", and "Arquivo". The table contains 16 rows of data, with the row for sequencial 02311 highlighted in blue. The data shows a gap between sequencial 02307 and 02311.

Sequencial	Data Abertura	Data Fechamento	Arquivo
02302	02/01/2013 04:02	02/01/2013 23:20	L0205_0012_0001_01206_06958189190975329084_02302_20130102_0402.BIN
02303	03/01/2013 03:57	03/01/2013 13:21	L0205_0012_0001_01206_06958189190975329084_02303_20130103_0357.BIN
02304	03/01/2013 13:27	03/01/2013 23:38	L0205_0012_0003_01206_06958189190975329084_02304_20130103_1327.BIN
02305	04/01/2013 04:21	04/01/2013 23:25	L0205_0012_0003_01206_06958189190975329084_02305_20130104_0421.BIN
02306	05/01/2013 04:48	05/01/2013 21:24	L0205_0012_0003_01206_06958189190975329084_02306_20130105_0448.BIN
02307	06/01/2013 05:20	07/01/2013 03:52	L0205_0012_0003_01206_06958189190975329084_02307_20130106_0520.BIN
02311	07/01/2013 06:45	07/01/2013 13:08	L0205_0012_0003_01206_06958189190975329084_02311_20130107_0645.BIN
02312	07/01/2013 13:12	07/01/2013 23:37	L0205_0012_0003_01206_06958189190975329084_02312_20130107_1312.BIN
02313	08/01/2013 04:20	08/01/2013 23:26	L0205_0012_0003_01206_06958189190975329084_02313_20130108_0420.BIN
02314	09/01/2013 03:33	09/01/2013 13:34	L0205_0012_0003_01206_06958189190975329084_02314_20130109_0333.BIN
02315	09/01/2013 13:40	09/01/2013 23:37	L0205_0012_0003_01206_06958189190975329084_02315_20130109_1340.BIN
02316	10/01/2013 04:15	10/01/2013 12:17	L0205_0012_0003_01206_06958189190975329084_02316_20130110_0415.BIN
02317	10/01/2013 12:22	10/01/2013 23:22	L0205_0012_0003_01206_06958189190975329084_02317_20130110_1222.BIN

Figura 3.3: Tela de busca de validador do SBE_DIFF

Sem a informação do tempo que um validador não comunica, não era possível priorizar um validador que estivesse atrasado há mais tempo, pois seria inviável verificar nos e-mails, todos os dias, quais validadores não estavam comunicando.

Foi criado um novo programa, que automaticamente verifica se há pulos e que informa os validadores que não estão comunicando.

3.2.2: Busca de cartões SD

O SBE possui uma quantidade limitada de cartões SD para serem inseridos nos validadores. Quando um cartão SD é retirado do validador, seu conteúdo é copiado para um computador. O programa SAGA é responsável por copiar apenas os arquivos “BIN” (arquivos da RioCard) para o servidor 31 utilizando uma máquina virtual com Linux (que é o sistema operacional utilizado pelo fabricante dos validadores). Após ser descarregado, o cartão SD vai para uma fila de análise. Ele somente pode ser liberado para formatação quando houver certeza de que todos os arquivos estão presentes no servidor.

É importante que esta análise seja feita para garantir que falhas no descarrego do SD, no procedimento de cópia do SAGA ou um erro humano (por exemplo, o funcionário esquecer de descarregar o SD) não acarrete na perda dos arquivos, o que significa prejuízo para a SuperVia.

O procedimento para realizar a análise consistia em verificar se havia pulo utilizando o SBE_Diff. Não havendo pulo, o SD poderia ser formatado. Caso contrário, era necessário aguardar a digitação e o pagamento, fazer o RAT-Site e verificar se há diferença.

Um software foi desenvolvido para buscar os SDs no sistema, facilitando a análise e reduzindo o tempo para atuação sobre o problema.

3.2.3: RAT-Site

3.2.3.1: Confronto entre valor digitado e pago

O confronto entre os valores digitados e pagos, realizado pelo SBE_DIFF, é extremamente lento para um período muito longo (maior que um mês). Tanto os

dados de entrada quanto de saída ficam armazenados em planilhas do Excel. No programa é possível selecionar a data, porém apenas os dados de digitação são filtrados, sendo que todos os dados de pagamento, armazenados na planilha, são utilizados. O único jeito de contornar isso é removendo as linhas de pagamentos antigos.

O modo como o processamento e o armazenamento de dados são feitos foi alterado, tornando-os mais eficientes, utilizando apenas os dados necessários e armazenando as informações em um banco de dados.

3.2.3.2: Pintura da planilha

O procedimento de pintar a planilha do RAT-Site é tedioso e suscetível ao erro humano, além de requerer um tempo de até uma hora de um funcionário por dia. Era necessário buscar as células da planilha onde o valor era positivo (o valor digitado foi maior que o valor pago) e verificar se havia falta de arquivo utilizando o SBE_Diff. Se não houvesse pulo, era necessário abrir o arquivo com o “ListaArqLog” (programa fornecido pela RioCard, para visualizar o número de transações em um arquivo, que é criptografado). Verificava-se se a RioCard não efetuou o pagamento corretamente ou se houve erro de digitação. Era necessário repetir este procedimento para todas as células com erro na planilha do RAT-Site.

A lógica que o funcionário devia utilizar para pintar a planilha corretamente foi implementada em um módulo de software, que foi adicionado ao SBE_Diff.

3.2.4: Diferença BU

No estado do Rio de Janeiro, foi implementada a Lei Nº 5628, de Dezembro de 2009, conhecida como lei do Bilhete Único (BU), que afirma que a integração entre dois meios de transporte nas cidades da região metropolitana do Rio de Janeiro não pode ser superior a um determinado valor [8] (no final de 2012, quando este projeto foi desenvolvido, este valor era de R\$4,95).

Quando um passageiro utiliza a SuperVia como o primeiro meio de transporte, o valor normal é debitado e gravado no arquivo “BIN” no validador. Quando o trem é o segundo meio de transporte, só será debitado e gravado no “BIN”

o valor que completa a tarifa do BU. Por exemplo, se a tarifa do ônibus for de R\$2,75, a tarifa do trem será de R\$2,20, completando o valor de R\$4,95. Como a tarifa da SuperVia (no final de 2012) era de R\$2,90, o governo deve ressarcir a SuperVia o valor de R\$0,70 para esta passagem.

O caso mostrado exemplifica apenas uma transação. Diariamente, milhares de pessoas realizam integrações com o RioCard, sendo o volume de dados do BU muito grande. Um funcionário da SuperVia era responsável por verificar, diariamente, se o governo realizou o ressarcimento corretamente, da mesma forma como é feito com a RioCard.

O procedimento para fazer a conferência consistia em (1) copiar os dados de um site e colar em uma planilha; (2) consultar o banco de dados do BU e colar em outra planilha; e (3) confrontar os dados manualmente. No Excel, era possível fazer o somatório do que foi pago e do que deveria ter sido pago, por data. O processo manual de verificação de arquivos pagos incorretamente era realizado apenas para datas onde as diferenças eram muito altas, pois o volume de dados é muito grande, tornando o procedimento completo inviável.

Foi desenvolvido um programa que faz o confronto automático entre os dados, retornando o validador e a data onde houve erros. Inicialmente, pediu-se que o programa retornasse o arquivo onde ocorreu o problema, mas foi verificado que isto não seria possível, por falta de informações importadas ao banco de dados do BU.

3.3: Análise da comunicação

A comunicação dos servidores de coleta apresenta falhas diariamente. É necessário que o SBE faça algumas análises, para que os problemas sejam encontrados e as correções sejam feitas.

Uma informação que pode ser utilizada para este fim está presente nos servidores de coleta, que produzem um log sobre a comunicação com os validadores. Porém, esses logs estão no formato de texto, sendo difícil fazer uma análise com eles. Um programa foi criado para processar esses logs, disponibilizando as informações contidas neles de forma mais fácil de analisar.

Para testar a comunicação com os servidores de coleta, foram criados dois aplicativos: um para Android, onde seria possível testar, a qualquer momento e de qualquer lugar, a comunicação com os servidores, e outro, para o computador, que mantenha um histórico da comunicação, para que horários de indisponibilidade possam ser identificados.

O último problema de comunicação se refere ao seu forçamento. Quando um atendente realizava este procedimento, não era possível saber se ele chegou ao final, ou se ocorreu uma falha. Para contornar este problema, foi criada uma tela no SOUBE que mostra se um validador já comunicou no dia atual.

3.4: Reenvio de arquivos Bin

A SuperVia e a RioCard determinaram que todos os arquivos “BIN” (que contém dados de transações com os cartões RioCard) , desde que o atual sistema de bilhetagem eletrônica entrou em operação em 2008, fossem reprocessados, para que arquivos que eventualmente estivessem pendentes fossem pagos.

Foi solicitado ao SBE que estes arquivos fossem separados. Porém, eles estão localizados nos backups da SuperVia, em um conjunto de pastas com mais de 1TB em arquivos de vários tipos, sendo muitos deles desnecessários. Além disso, muitos arquivos “BIN” estão repetidos, possuindo até 5 cópias, ou compactados dentro de arquivos ZIP, precisando ser extraídos antes de enviados.

Foi pedido, também, que os arquivos corrompidos fossem separados em outra pasta, pois eles serão processados de uma forma diferente para serem pagos.

No final, deseja-se obter uma pasta com todos os arquivos, sem repetições, e outra pasta com os corrompidos. Um programa foi criado para realizar essa separação. O procedimento de separação será realizado em quatro partes de aproximadamente 250GB.

3.5: Análise de indicadores do SBE

O banco de dados do SBE possui vários dados que são guardados em seu histórico, mas que não são mostrados da melhor forma possível.

Após uma análise de quais desses dados são úteis, pensou em formas de utilizá-los. Um novo aplicativo foi criado para mostrar essas informações, com seus 3 módulos descritos a seguir.

3.5.1: Trocas de validadores

No SABE, os dados de trocas de validadores são mostrados em um gráfico de barra, evidenciando o número de trocas em uma posição na estação por mês. Este método de exibição não é adequado para se fazer uma análise de quais validadores apresentam mais problemas, sendo que eles raramente são utilizados.

Neste novo programa, há uma tela onde é possível visualizar facilmente os períodos de atividade dos validadores. Essa análise é possível tanto por validador físico quanto por posição na estação.

3.5.2: Atendimento de chamados

Quando um técnico atende um chamado em um validador e, após menos de 10 dias, o problema ocorre novamente, tem-se uma “reincidência”. Está especificado, na SLA, que a taxa de reincidência, por técnico, deve ser inferior a 20% do total de chamados atendidos.

No banco de dados do SABE, há um histórico de todos os chamados atendidos. Um dos parâmetros de um chamado é o nome do técnico que o atendeu. Foi criada uma tela que exibe o total de chamados atendidos por técnico, juntamente com a sua taxa de reincidência.

3.5.3: Relação entre problema, causa e solução de problema

Outros 3 parâmetros que podem ser encontrados em um registro de chamado no banco de dados do SABE são o problema, sua causa e a solução adotada.

Uma tela existente no SABE mostra o total de ocorrências de cada problema, causa e solução. Porém, esta tela também é pouco utilizada.

A última tela do aplicativo de indicadores realiza uma ligação entre os três parâmetros, o que auxiliará na prevenção de erros e tornará a manutenção mais rápida.

3.6: Considerações finais

Neste capítulo foram mostrados os problemas de configuração de validadores, de ressarcimento, de comunicação e de restauração de arquivos, além da necessidade de se poder visualizar indicadores do setor.

Estes problemas foram identificados ao longo do projeto, pela SuperVia ou pelos funcionários da Radix.

A especificação dos novos aplicativos será descrita no próximo capítulo.

Capítulo 4: Especificação dos novos aplicativos

Com a implementação dos novos sistemas, criados pelo autor deste trabalho para resolver os problemas apresentados no Capítulo 3:, diversas tarefas foram facilitadas no SBE. Neste capítulo será mostrado o funcionamento de cada um dos novos programa.

4.1: Configuração de validadores: ValparaMod.exe

O Valparamod é o aplicativo onde o arquivo de configuração de validadores, o “valparam.ini”, será configurado.

Ao abrir o programa, o usuário já pode realizar uma busca. Para isso, ele informa o nome, a sigla ou o número da estação, e o número da posição do validador. É necessário digitar apenas parte do nome da estação, sendo que o programa retorna as opções caso haja mais de uma (por exemplo, se a estação digitada for “penha”, o sistema mostra as opções “Penha” e “Penha Circular”). Quando o validador for selecionado, o aplicativo mostrará uma tela com todos os parâmetros do validador.

Caso o usuário deseje editar um parâmetro, ele deve realizar o login, informando o nome de usuário e senha.

Se o usuário não for administrador, ele poderá apenas alterar o número do validador físico (CPG), a data e a hora do validador selecionado. Estes são os parâmetros com necessidade de alteração mais frequentes, pois cada vez que um validador vai para campo no lugar de um que apresente defeitos, a CPG do novo validador físico deve estar associada ao carro equivalente no valparam.

O usuário administrador terá permissão para alterar, para cada validador, o servidor de conexão e de tempo. Além disso, ele pode alterar os parâmetros de configuração comuns a todos os validadores (configurações de APN e da rede). Por último, o administrador poderá adicionar novos registros no Valparam, quando o fluxo de passageiros em uma estação estiver muito grande e surgir a necessidade de adicionar novos validadores.

4.2: Ressarcimento

4.2.1: ArquivosFaltantes.exe

O “ArquivosFaltantes.exe” lista tanto arquivos faltantes por pulo sequencial quanto por dias sem comunicação. As informações são mostradas na forma de uma tabela, onde as linhas equivalem a registros de falta de arquivos. Cada registro mostra o sequencial inicial e final, o número de pulos, a data de abertura e fechamento, o ramal, o número e nome da estação, o número do validador e o tempo de atraso. Caso haja falta contínua de arquivos, o sequencial final, o número de pulos e a data final ficarão vazios.

É possível excluir da lista os “pulos equivocados”, que são pulos sequenciais onde não há falta de arquivo (erro decorrente de uma falha no validador), realizando um duplo-clique sobre a linha do validador. Também é possível desabilitar validadores que estejam inoperantes, cuja linha se torna amarela para fácil identificação.

Diariamente, esta tabela é atualizada automaticamente, as sete horas da manhã. Os novos pulos e faltas contínuas são adicionados na tabela, sendo possível alterá-los ao longo do dia.

No momento que a tabela é gerada, uma planilha no Excel é criada e enviada ao responsável por pedir a coleta de arquivos aos atendentes.

4.2.2: SDFinder.exe

O SDFinder realiza uma busca no sistema de pastas onde os cartões SD são descarregados, para analisar possíveis problemas relativos no descarregamento, além de auxiliar no momento de liberação para formatação.

O aplicativo permite a busca por estação/validador ou por número de SD. O usuário deve informar o período desejado e clicar no botão “Buscar”. Uma lista de SDs será retornada para o período. O usuário pode duplo-clicar no SD desejado, o que mostrará uma tabela com todos os arquivos BIN contidos na pasta onde o SD foi descarregado.

4.2.3: Novos SBE_DIFF

O novo método de comparação é feito após a cópia dos valores pagos do site para uma planilha Excel. Escolhe-se a data e clica-se em "Gerar RAT-Site". Assim como o SBE_DIFF, uma tabela será gerada, mostrando o resultado do confronto entre a digitação e o pagamento. Este valor é copiado automaticamente para a planilha do RAT-Site. A tabela dinâmica contida na planilha será atualizada.

A seguir, a planilha é carregada no programa e as células onde há diferença entre o valor pago e o digitado pela Arrecadação são pintadas automaticamente.

Quando todas as células estiverem pintadas, o sistema salva a planilha com as novas cores.

4.2.4: BU_DIFF

Este software faz a conferência automática do ressarcimento do governo à SuperVia.

Assim como na geração do RAT-Site no SBE_DIFF, é necessário copiar a tabela de valores ressarcidos pelo governo através do site. O usuário deve informar o período de análise desejado e clicar em "Carregar". O programa irá carregar do banco de dados os valores de ressarcimento importados dos arquivos BIN. O confronto é feito para os dias informados, gerando uma tabela de erros por dia.

O usuário deve duplo clicar na linha com a data desejada, levando o aplicativo à tela detalhada, onde são listados todos os erros, por validador.

É possível também, na tela de erros por dia, clicar no botão "BU_DIFF", que criará uma tela de detalhes listando os erros de todos validadores para todo o período especificado.

4.3: Ferramentas análise de comunicação

4.3.1: Resumo da comunicação

O ResCom.exe é um aplicativo que converte os arquivos textos de log dos servidores em uma planilha do Excel, de fácil visualização. Os logs continham muitas informações sem utilidade, que não são mostradas na planilha.

Para o funcionamento do programa, só é necessário informar o mês e ano desejado no console e o programa gerará a planilha.

4.3.2: Comunicação com servidores

Este aplicativo tenta se comunicar com os três servidores de coleta em intervalos de tempo pré-determinados. Os dados de êxito ou não serão gravados em um banco de dados, sendo possível visualizá-los em um gráfico.

4.3.3: Teste de servidores para Android

O aplicativo para Android possui uma interface simples, com um botão para cada servidor. Ao clicar em um botão, o aplicativo tentará alcançar o servidor desejado, informando se foi possível ou não atingi-lo.

4.3.4: Comunicação atualizada no SOUBE

A nova tela no SOUBE mostra uma lista com todas as estações sua situação de comunicação no dia atual. Se o validador comunicou, ele será exibido na cor verde, com o texto “SIM”. Se não comunicou, ele ficará vermelho, com o texto “NÃO”.

4.4: Separação de arquivos

O aplicativo de separação de arquivos foi criado para juntar todos os arquivos “BIN” dos últimos 5 anos e reenviá-los à RioCard. Além disso, era necessário separar os arquivos corrompidos dos demais, pois eles passarão por um processo de recuperação, que levará mais tempo.

O SeparadorArquivosBins.exe junta todos os arquivos bins de backup em uma pasta, removendo duplicidade. Os arquivos “BIN” contidos em arquivos zip também são tratados: são extraídos diretamente na pasta destino.

Após a junção de todos os arquivos de um período em uma pasta, sem repetição, o procedimento de separação de corrompidos é realizado. É feita uma tentativa de ler cada arquivo. Se ela for bem sucedida, nada é feito. Se ocorrer um problema, o arquivo é separado em uma pasta de corrompidos.

Este procedimento todo será feito em 4 partes, pois o volume de arquivos é muito grande. A primeira parte já foi feita em paralelo com o desenvolvimento deste projeto, sendo que as próximas três serão realizadas ao longo dos próximos meses.

4.5: Indicadores do SBE

4.5.1: Gráfico de Trocas

Esta tela busca informações sobre a troca de validadores, tanto os físicos quanto por posição na estação, para descobrir quais apresentam maiores problemas.

O usuário deve escolher um período, o modo de ordenação (por número de trocas ou pela nota que o aplicativo dá ao validador) e clicar no botão de confirmação.

São exibidas duas listas ordenadas: de validadores físicos e de posições. Escolhe-se um validador de uma lista (ou das duas, caso se queira comparar) e um gráfico é gerado, mostrando quando o validador esteve em campo e quando esteve em manutenção.

4.5.2: Atendimento de chamados

Nesta tela são listados os técnicos e o total de chamados atendidos por eles.

Primeiramente, o usuário deve informar o período de onde se deseja visualizar as informações. A seguir, basta clicar no botão “Listar” que todos os técnicos serão listados, mostrando o total de chamados por eles atendidos e suas respectivas taxas de reincidência

4.5.3: Busca de solução

Na tela de busca de solução, o usuário informa a data inicial de análise e clica em “Buscar”. Uma lista com todos os problemas possíveis, ordenada por número de ocorrências, será exibida. O usuário deve selecionar o problema que deseja analisar.

Uma nova lista mostrará todas as causas que já provocaram o problema informado, também ordenada por número de ocorrências. O usuário escolhe uma das causas e o programa mostrará a última lista, com todas as soluções para a relação problema-causa que foi informada.

4.6: Considerações finais

Neste capítulo foi apresentado o modo como os novos programas de configuração de validadores, de automatização do ressarcimento, de análise da comunicação, de restauração de arquivos e de visualização de indicadores foram idealizados.

Para isso, o funcionário que requisitou o programa esboçava, em papel, diagramas e telas de interface gráfica, para explicar como o programa deveria funcionar. A partir desses esboços, o autor deste relatório criava seus próprios diagramas de funcionamento, já pensando na melhor forma de realizar a implementação (Capítulo 5:).

Capítulo 5: Desenvolvimento dos novos aplicativos

Este capítulo aborda a implementação dos aplicativos criados pelo autor deste relatório, para a resolução dos problemas apresentados no Capítulo 3:. Serão mostradas as ferramentas utilizadas para o desenvolvimento, assim como diagramas que demonstram a organização e o funcionamento dos algoritmos.

5.1: Processo de desenvolvimento

Esta seção descreve como ocorreu o processo de desenvolvimento, desde a identificação dos problemas até a integração na versão de produção.

Inicialmente, os problemas eram identificados e as soluções eram idealizadas por funcionários do SBE. No caso do ValparamMod e do programa de indicadores, a identificação foi feita pelos técnicos e pelo coordenador do projeto da Radix. Já as ferramentas para comunicação foram solicitadas pelo funcionário responsável pela comunicação, também em conjunto com o coordenador. O BU_Diff e a separação de arquivos foram pedidos pela própria SuperVia, embora a solução do SeparadorDeArquivos tenha sido pensada pelo autor deste relatório. Por fim, a ideia de automatização do RAT-Site também foi elaborada pelo autor, pois ele era encarregado de fazer o RAT-Site e encontrou uma forma de facilitar a tarefa.

Para explicar o modo de operar desejado ao desenvolvedor (autor deste trabalho), o funcionário esboçava diagramas de funcionamento e a interface gráfica. O autor, então, iniciava a codificação, pesquisando soluções conforme a necessidade. A solução de diversos problemas foram encontradas no site <http://stackoverflow.com/> [9].

Os testes eram feitos utilizando a ferramenta de “debug” do ambiente de desenvolvimento. O único teste mais complexo foi do ValparamMod, que utilizava o servidor de homologação (nucsrv32) para realizar testes fora de produção. A utilização deste servidor era de extrema importância, pois um erro no arquivo valparam em produção poderia atrapalhar o funcionamento de todos os validadores da SuperVia.

Em qualquer programa, quando não eram detectados mais erros utilizando a ferramenta de “debug”, a instalação era feita no computador do usuário.

Caso o usuário identificasse alguma melhoria a ser feita, o autor deste relatório fazia as alterações, testava novamente e colocava em produção.

5.2: Ferramentas para o desenvolvimento

5.2.1: Plataforma “.NET”

Um “software framework”, ou plataforma de software, é um conjunto de códigos e funcionalidades comuns entre aplicações. A plataforma .NET é uma plataforma de software, criada pela Microsoft para rodar programas no Windows [10]. Possui muitas bibliotecas, sendo possível utilizar diversas linguagens de programação diferentes.

Com a plataforma, não é necessário desenvolver diversas funcionalidades, que já estão implementadas, bastando chamar as funções e passar os parâmetros necessários.

5.2.2: Plataforma ASP .NET

ASP .NET é uma plataforma criada pela Microsoft para desenvolvimento de aplicativos para a “web” [11]. Ela permite o uso de classes criadas em C# e outras linguagens para desenvolver páginas dinâmicas (cujo conteúdo varia de acordo com parâmetros fornecidos pelo usuário ou pelo próprio computador). É ligada à plataforma “.NET”.

Para hospedar um “site” criado em ASP .NET, é necessário possuir o IIS (Internet Information Services), que atuará como servidor “web”.

5.2.3: Visual Studio

O Visual Studio é o ambiente de desenvolvimento integrado (IDE) criado pela Microsoft. Pode ser utilizado tanto para criar aplicações para console quanto com

interface gráfica para o usuário [12]. Pode ser usado também para criar “sites” para a “web” utilizando a plataforma “ASP .NET”.

Possui telas para edição de código e para criação da interface gráfica, quando pertinente. O modo “debug” permite encontrar problemas facilmente e alterar o código, em tempo de execução.

Diversos “plugins” são aceitos, e a adição de bibliotecas com funcionalidades prontas é simples, o que acelera o desenvolvimento.

Suporta várias linguagens, embora ele venha apenas com C, C++, “Visual Basic”, C# e F#. Outras linguagens também podem ser instaladas. Neste trabalho foi utilizada a linguagem C#, para desenvolvimento de aplicativos locais.

5.2.4: C#

C# é uma linguagem orientada a objetos baseada em C++, fortemente tipada (há restrições na interação entre variáveis de tipos diferentes), criada pela Microsoft [13]. Ela está muito vinculada à plataforma “.NET”, de onde obtém diversas funções já programadas. Pode ser utilizada para desenvolver aplicativos para o console ou com interface gráfica.

5.2.5: MySQL e HeidiSQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) de ampla aceitação em um âmbito global, utilizado por grande empresas, como Google e Facebook [14].

Neste projeto, alguns aplicativos utilizam o MySQL para manipular informações dos bancos de dados, por meio de “queries”.

O HeidiSQL, cliente para o MySQL, foi utilizado no trabalho. Através dele, o usuário pode efetuar o login para visualizar e alterar informações diretamente dos bancos de dados, conforme desejado.

5.2.6: Android

Android é um sistema operacional de código aberto, baseado em Linux, criado pela Google [15]. Ele funciona em celulares smartphone e em tablets. Neste trabalho, o aplicativo para Android foi feito em Java, através do Eclipse.

O sistema Android, que é o mais utilizado no mundo, foi escolhido por ser o sistema operacional dos smartphones dos usuários do SBE que estão utilizando o programa.

5.3: Configurador do Valparam

Quando o programa de configuração de validador é iniciado, ele irá copiar o arquivo valparam para a pasta local do aplicativo. Dessa forma, se houver problema na rede, no aplicativo ou no computador de execução, o valparam que está em produção não será corrompido.

Após a cópia, o arquivo é lido, carregando as informações para uma matriz de “strings”. Cada linha da matriz representa um validador, enquanto que as colunas representam os 21 parâmetros configuráveis (Figura 3.1).

Além do arquivo valparam, o aplicativo carrega uma tabela que contém o mapeamento entre o número do “carro” (o segundo parâmetro do valparam, que representa uma posição na estação), a estação e a posição do validador (por exemplo, o carro 574 no sistema equivale à estação Gramacho, ou estação 79, validador 7).

Quando o usuário realiza uma busca (Figura 5.1), digitando parte do nome da estação, sigla ou número da estação, o sistema inicialmente faz uma varredura buscando linhas que contenham os valores informados. Caso haja apenas uma linha, o número dela é gravado em uma variável. Caso haja mais de uma, o usuário seleciona o validador desejado a partir de uma tabela na interface, e a variável também é salva. Em seguida, a tela de edição é carregada, com todos os parâmetros editáveis.

Porém, para que os parâmetros possam ser alterados, é necessário efetuar o “login”. O sistema fará a autenticação do usuário, utilizando o banco de dados do aplicativo.

O usuário pode fazer as alterações e clicar no botão salvar (Figura 5.2). O arquivo valparam local da aplicação é alterado. A seguir, o sistema tenta copiar um arquivo temporário para cada um dos 4 servidores, para testar a conexão pela rede. Caso a cópia seja realizada com sucesso, o valparam será copiado para os servidores. Um arquivo de log no servidor 31 será alterado, adicionando o nome do usuário e a alteração que ele realizou.

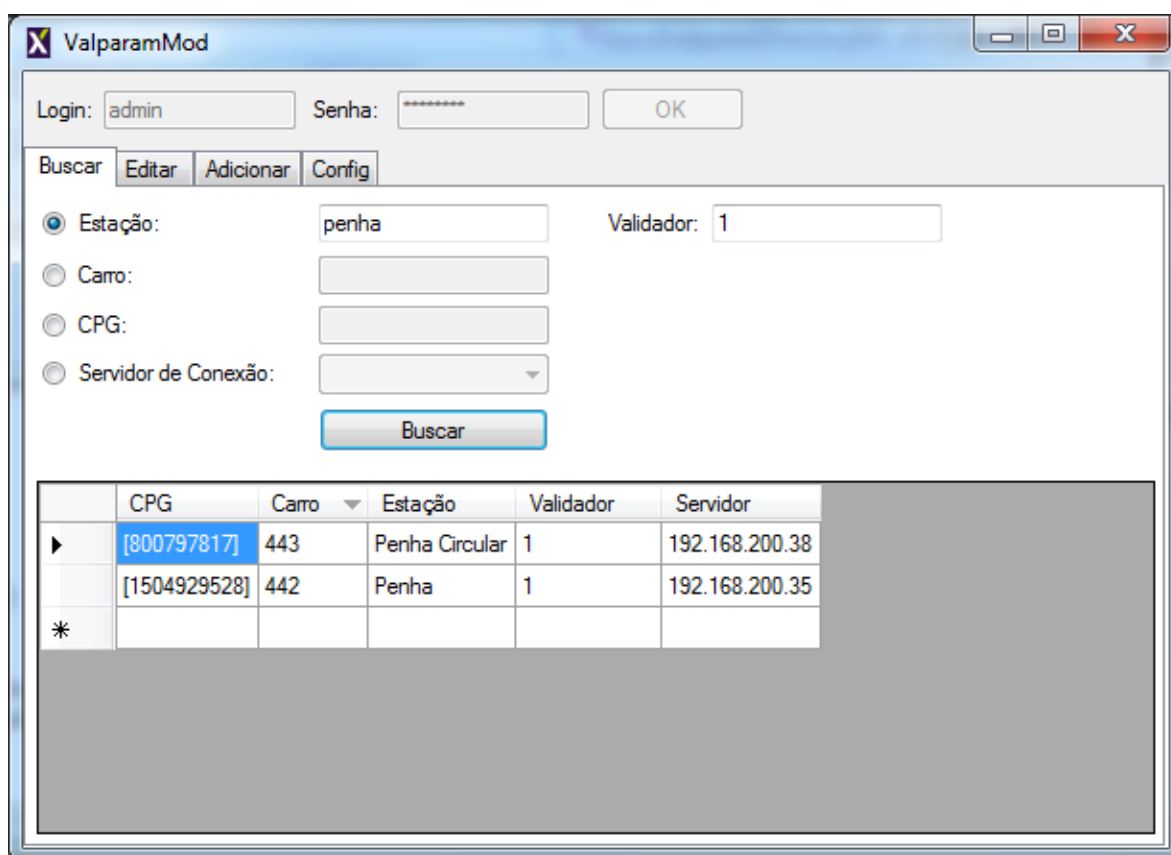


Figura 5.1: Interface de busca do ValparamMod

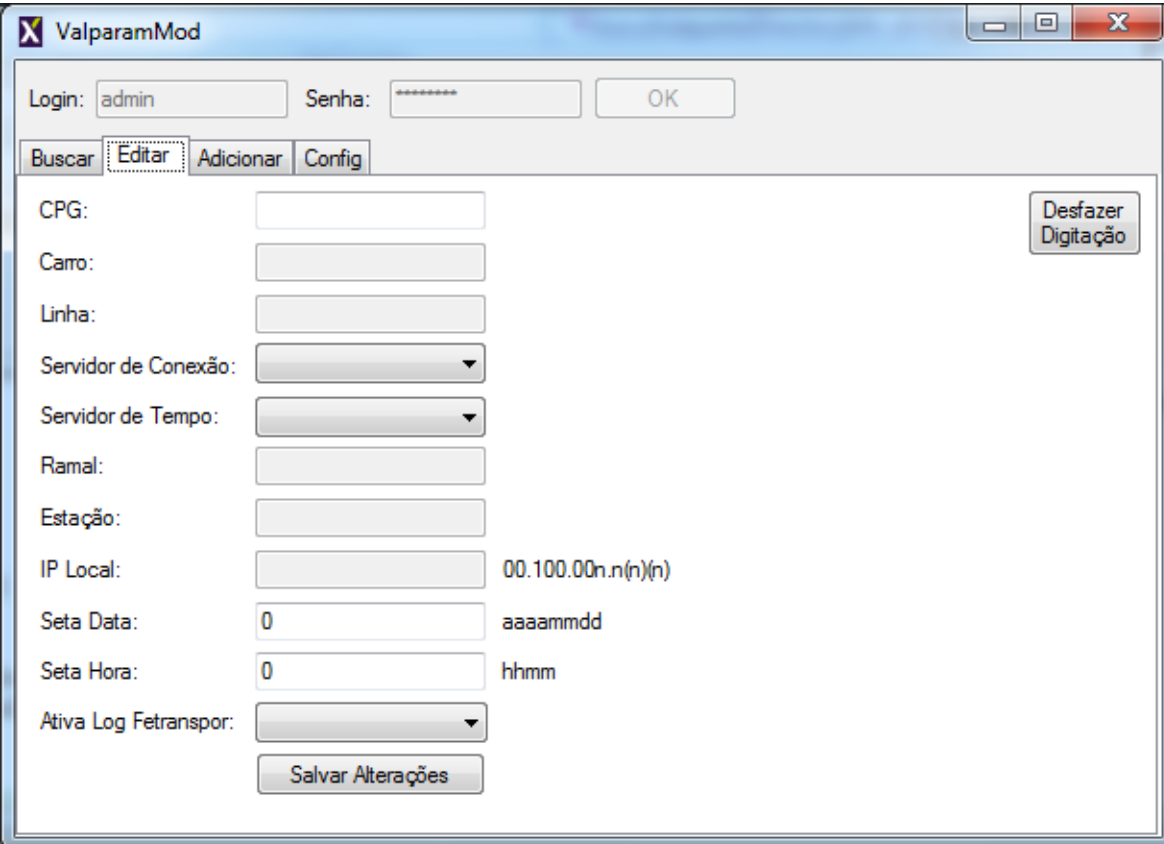
Os procedimentos de adição de um novo validador e de alteração das configurações comuns (Figura 5.3), que só podem ser realizados por usuários administradores, é similar.

Para resolver a questão de exclusão mútua, o sistema cria um arquivo temporário. Quando o usuário fechar a aplicação, este arquivo será excluído.

Enquanto o usuário estiver conectado, o sistema irá atualizar o arquivo a cada 2 minutos, inserindo o nome do usuário e o horário atual.

Se outro usuário tentar acessar o programa, aparecerá uma mensagem informando que já existe um usuário utilizando o software.

Caso o aplicativo do usuário conectado pare de funcionar ou perca a conexão com a rede, o arquivo temporário não será excluído. Porém, se outro programa tentar acessar o aplicativo após certo tempo, ele detectará que a hora do arquivo temporário está desatualizada. Assim, ele conseguirá entrar no programa, e um novo arquivo temporário sobrescreverá o antigo.



The image shows a screenshot of the ValparamMod application window. The window title is "ValparamMod". At the top, there is a login section with "Login: admin" and "Senha: *****" followed by an "OK" button. Below the login section, there are four tabs: "Buscar", "Editar", "Adicionar", and "Config". The "Editar" tab is currently selected. The main area contains several configuration fields: "CPG:" (text input), "Carro:" (text input), "Linha:" (text input), "Servidor de Conexão:" (dropdown menu), "Servidor de Tempo:" (dropdown menu), "Ramal:" (text input), "Estação:" (text input), "IP Local:" (text input) with a placeholder "00.100.00n.n(n)(n)", "Seta Data:" (text input) with "0" and a format "aaaammdd", "Seta Hora:" (text input) with "0" and a format "hhmm", and "Ativa Log Fetranpor:" (dropdown menu). There is a "Desfazer Digitação" button on the right side and a "Salvar Alterações" button at the bottom.

Figura 5.2: Interface de edição do ValparamMod

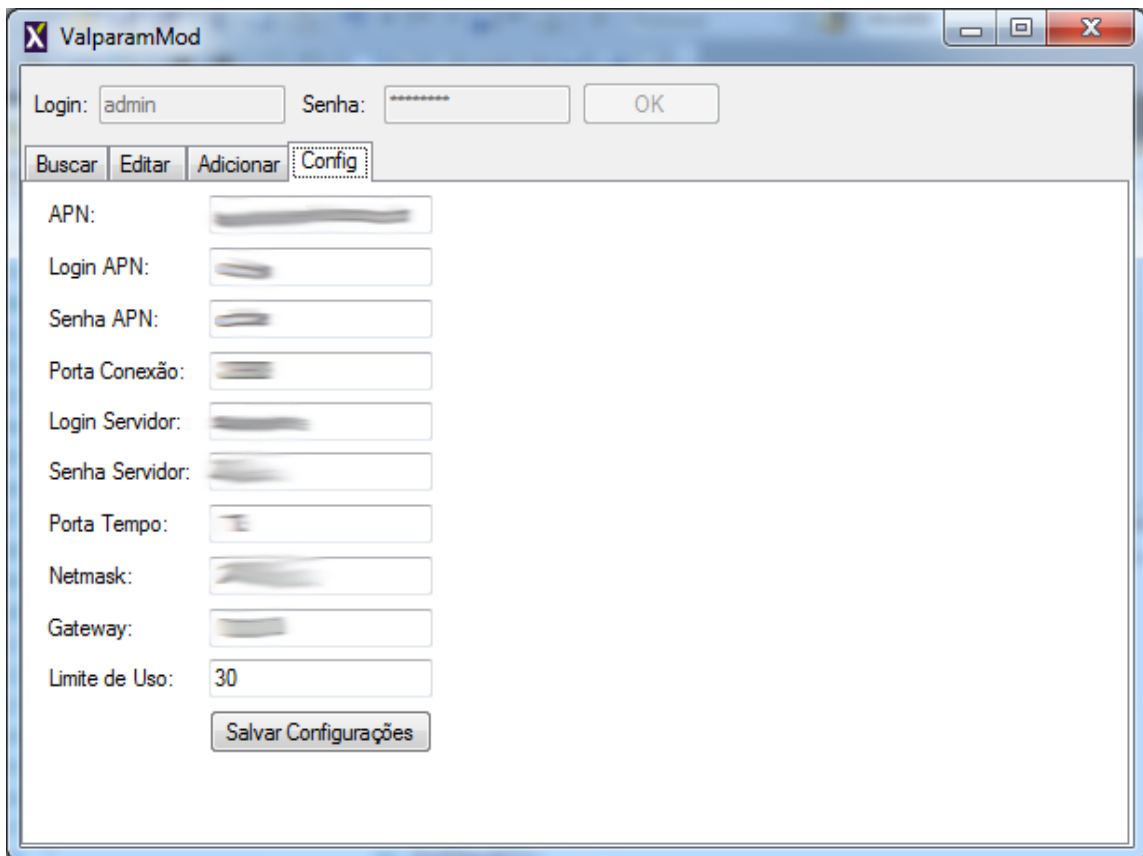


Figura 5.3: Interface de edição de parâmetros comuns do ValparamMod

A organização do ValparamMod pode ser vista no diagrama de classes da Figura 5.4:

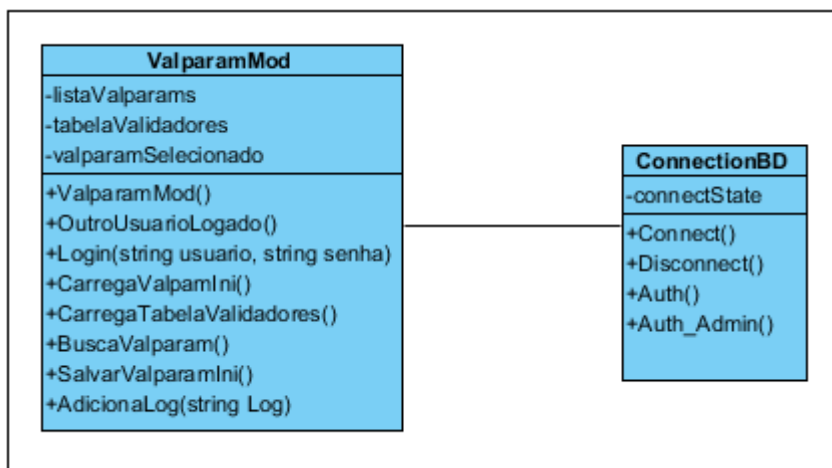


Figura 5.4: Diagrama de classes do ValparamMod

O funcionamento do programa é representado no diagrama de atividades da Figura 5.5.

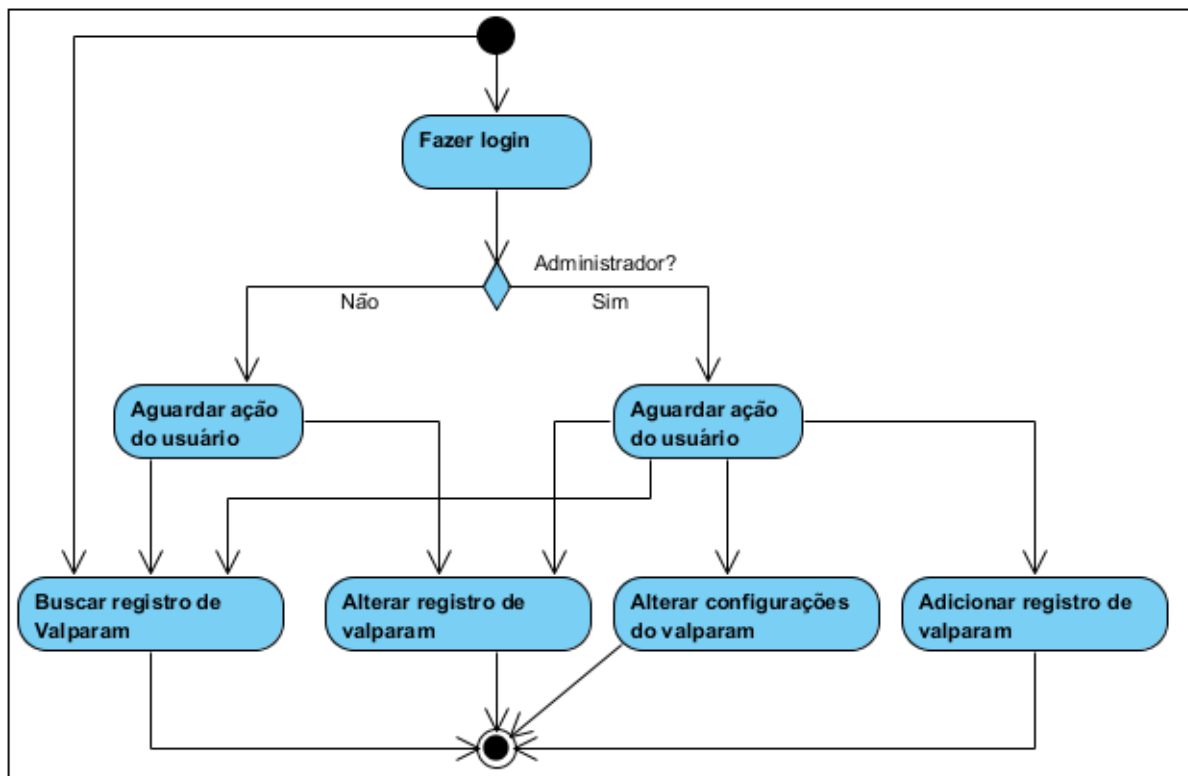


Figura 5.5: Diagrama de atividades do ValparamMod

5.4: Ressarcimento

5.4.1: Arquivos Faltantes

Para entender o funcionamento deste aplicativo, é necessário compreender a nomenclatura dos arquivos BIN:

L0205_0012_0003_07907_12345678901234567890_01532_20120828_0413.BIN

A primeira parte do arquivo (L0205_0012_0003_) é padrão, sendo igual para todos os BINs. Na sequência, temos o número da estação e do validador (07907 equivale à estação Gramacho, ou 079, validador 07). O próximo número, de 20 dígitos, é o número associado ao “chip” de celular no validador. O número seguinte, de cinco dígitos, representa o sequencial (no caso, temos o 1532º arquivo de Gramacho 7). A seguir temos a data de criação do arquivo, no formato ano-mês-dia (“20120828”, o que equivale a 28 de Agosto de 2012), e por último o horário de criação, no formato hora-minuto (04:13).

Embora todas as transações no arquivo sejam criptografadas, há uma assinatura digital no final do arquivo que não é. Um exemplo de assinatura digital é o seguinte:

0001160000029406000000000000409600012200000000409636700846

Nesta assinatura, encontram-se os seguintes dados: (1) o total de transações; (2) o total de dinheiro (em reais), que será pago pela RioCard à SuperVia; (3) a data de fechamento do arquivo; e (4) a hora de fechamento do arquivo.

Os seis primeiros dígitos equivalem ao número de transações, que no caso é 116. Os próximos 10 dígitos representam o valor em reais, porém é necessário realizar a seguinte conta:

$$\text{Valor Real} = \frac{\text{Valor Arquivo} - 4096}{100}$$

No exemplo:

$$\text{Valor Real} = \frac{0000029406 - 4096}{100} = 253,10 \text{ reais}$$

Dos últimos 8 dígitos, os quatro primeiros representam o número de dias, a partir de “01/01/2003”, para o fechamento do arquivo. É necessário somar o número de dias da assinatura a esta data para obter a data de fechamento:

$$\text{Data de Fechamento} = 01/01/2003 + 3670 \text{ dias} = 28/08/2012$$

Por fim, os último quatro dígitos representam o horário de fechamento, que no caso é “08:46”.

Neste programa de arquivos faltantes, apenas a data de fechamento é utilizada. O número de transações é utilizado para a pintura automática do RAT-Site, explicada mais adiante.

Com o entendimento do nome e da assinatura dos arquivos, podemos partir para o funcionamento do aplicativo.

Todos os arquivos que chegam ao SBE após passarem pelo processamento do SAGA, caem numa pasta chamada “Quarentena”.

Quando o aplicativo é iniciado, ele cria uma lista com o nome de todos os arquivos do mês atual e do anterior presentes na quarentena. Uma varredura é feita buscando pulos nos sequenciais e buscando a data do último arquivo de todas as estações.

Para encontrar os pulos, durante a varredura, o programa verifica se, para um mesmo validador, o sequencial do arquivo atual é igual ao sequencial do arquivo anterior acrescido de 1. Se não for, há um pulo.

Havendo um pulo, o programa adiciona um registro em uma tabela (Figura 5.6), informando a estação, o validador, a data e a hora de fechamento do arquivo anterior ao pulo (o que equivale à data e hora de abertura do arquivo faltante), e a data e a hora de abertura do arquivo posterior ao pulo (o que equivale à data e hora de fechamento do arquivo faltante).

Além dos pulos, se a data do último arquivo de uma estação não for igual à data atual, é adicionado um registro na mesma tabela, com as mesmas informações do pulo (com exceção do sequencial, data e hora de abertura do próximo arquivo, pois este não está presente).

Figura 5.6: Tela do ArquivosFaltantes.exe

A partir da data de fechamento do último arquivo, obtida da assinatura, o sistema retorna, para cada registro de pulo ou falta contínua, um valor de prioridade, que é calculado da seguinte forma:

$$\text{Prioridade} = \text{Data Atual} - \text{Data de Fechamento do Último Arquivo}$$

Os registros da tabela são adicionados em um banco de dados, onde uma variável de “status” é configurada como “buscar”. Os registros são mostrados em uma tabela na interface gráfica.

Caso uma estação esteja desativada, haverá um registro informando falta de arquivos contínua. É possível desabilitar este validador da tabela, o que altera a situação dele no banco de dados para “desativado”.

Se houver a confirmação de que um pulo sequencial foi falha do equipamento (ou seja, não há falta de arquivos), o usuário deve remover a linha do pulo. A situação no banco de dados será alterada para “pulo equivocado”, e a linha será removida da tabela na interface gráfica.

O sistema, então, irá criar uma planilha do Excel, sem os validadores inativos e sem os pulos equivocados, e enviará ao responsável pela coleta.

O usuário tem a possibilidade de visualizar um histórico, para um período de 30 dias, que mostra quantos validadores estão atrasados em um, dois, três ou mais dias. O programa utilizará uma “query” para obter esses dados direto do banco de dados, imprimindo-os na tela.

A organização do ArquivosFaltantes pode ser vista no diagrama de classes da Figura 5.7:

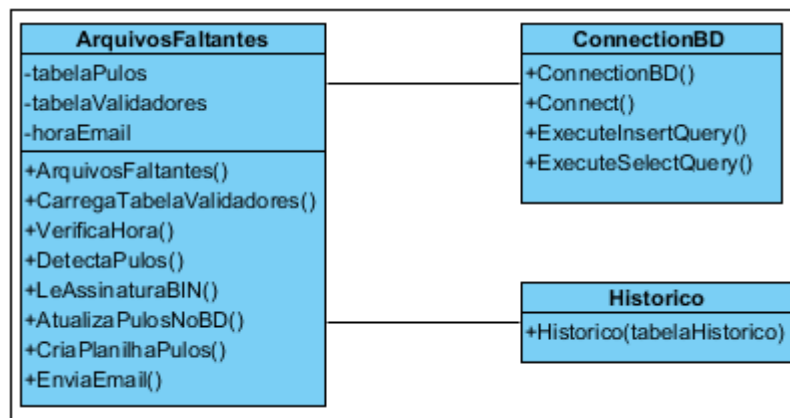


Figura 5.7: Diagrama de classes do programa ArquivosFaltantes

O funcionamento do programa é representado no diagrama de atividades da Figura 5.8:

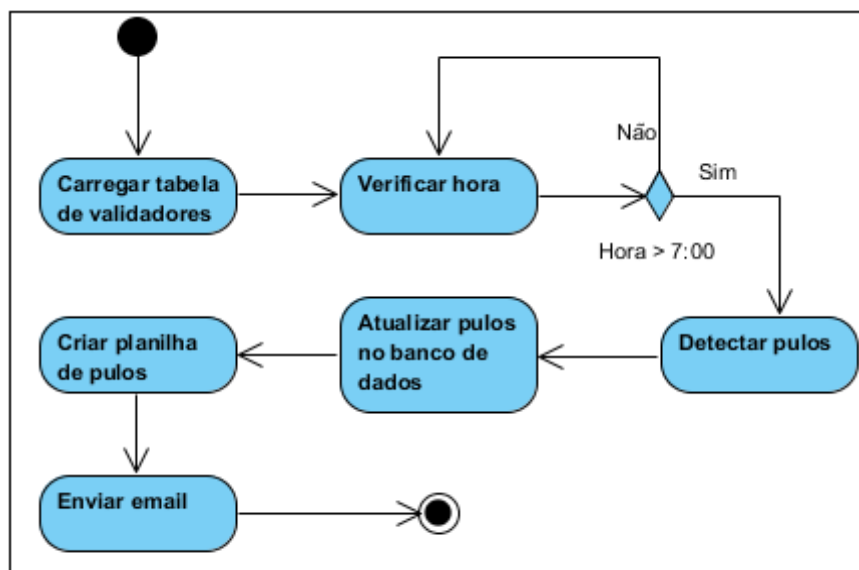


Figura 5.8: Diagrama de atividades do ArquivosFaltantes

5.4.2: Busca de SD

O usuário do SDFinder deve passar as seguintes informações ao programa:

- data de início e fim;
- se a busca será pelo conjunto estação/validador ou por número do cartão SD;
- string de busca, que pode ser tanto por estação/validador quanto por número do SD, dependendo da escolha feita.

Na tela da Figura 5.9, o SD 1298 foi procurado. O programa busca subpastas na pasta “SDCard” que possuam o mesmo nome que o número do cartão SD. Caso exista mais de uma ocorrência do SD no período informado, o usuário poderá escolher um deles. Todos os arquivos nele contidos serão listados.

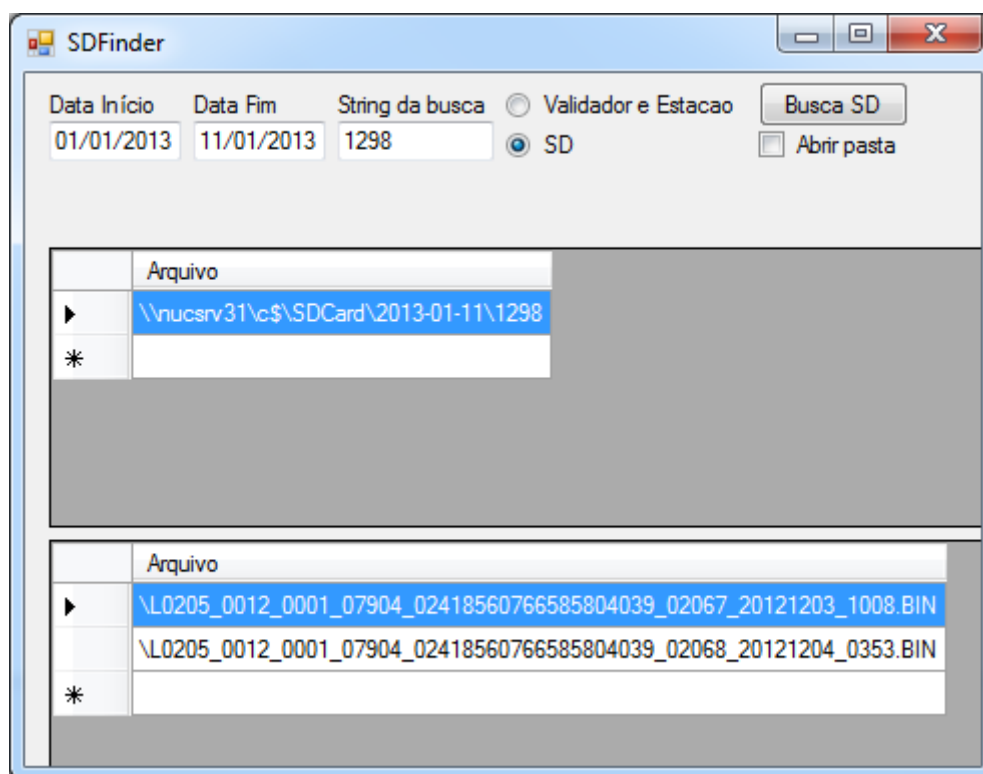


Figura 5.9: Interface do SDFinder

Quando a busca é feita por estação/validador, o usuário deve digitar a string no formato do nome do arquivo (para Gramacho 4, ele digitaria 07904). O programa buscará todos os arquivos que contenham a string especificada nas subpastas dos cartões SDs salvos no servidor 31.

A busca por estação/validador, por acessar muitas subpastas e nomes de arquivos, é mais demorada que a busca por número de cartão SD.

A organização do SDFinder pode ser vista no diagrama de classes da Figura 5.10:

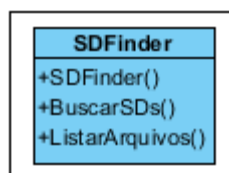


Figura 5.10: Diagrama de classe do SDFinder

O funcionamento do programa é representado no diagrama de atividades da Figura 5.11:

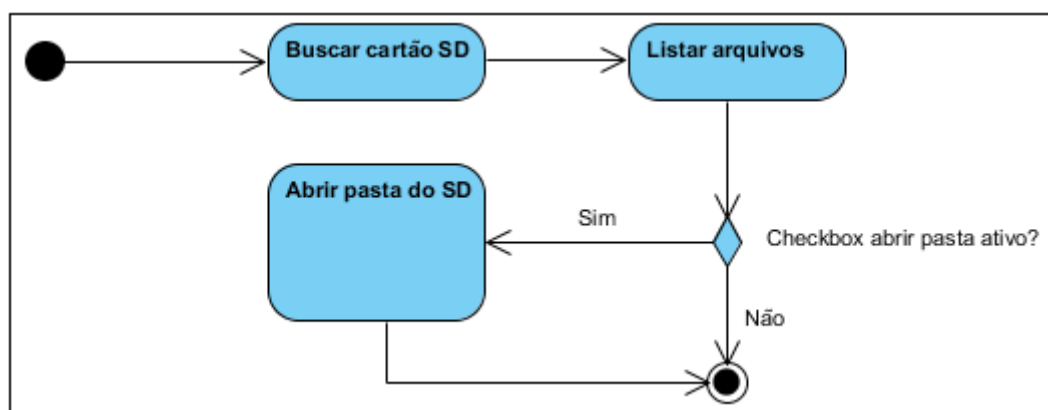


Figura 5.11: Diagrama de atividades do SDFinder

5.4.3: Novo SBE_DIFF

O antigo SBE_DIFF carregava as informações de pagamento (de uma planilha) e de digitação (do banco de dados da SuperVia), e os inseria em “arrays” na memória. Para obter a tabela do RAT-Site, o programa realizava um “for” dentro de um “for” para comparar os “arrays”.

O novo SBE_DIFF insere os dados de pagamento da planilha em um banco de dados e realiza uma consulta, por meio de uma “query”, para obter a tabela do RAT-Site completa. O novo método é muito mais eficiente.

A seguir, a tabela é inserida na planilha do RAT-Site e a tabela de comparação é atualizada automaticamente.

A pintura automática deve executar a mesma lógica utilizada pelo ser humano, para colorir as células da planilha do RAT-Site na cor correta. Cada célula representa a diferença para um validador em uma data específica.

Inicialmente, a planilha é lida, tendo seus valores carregados numa matriz no programa. A seguir, os arquivos da quarentena são carregados, da mesma forma que no programa de “Arquivos Faltantes”.

A matriz do RAT-Site será varrida, célula a célula, buscando valores maiores que zero. Quando um valor positivo for encontrado, o sistema vai primeiramente buscar um pulo de sequencial na matriz de arquivos da quarentena. Havendo um pulo, a célula na planilha é pintada de azul. Caso o programa não encontre pulo, mas verifique que o último arquivo de um validador não seja do dia atual, a célula positiva também é pintada de azul.

Se não houver falta, os arquivos do validador na data da célula são lidos. Assim como é feito nos “Arquivos Faltantes”, a assinatura digital do arquivo é lida. Se a assinatura não existir, o arquivo está corrompido, e a célula é pintada de cinza. Porém, se for possível obter o número de transações das assinaturas, seus valores serão somados.

Caso o valor desta soma seja igual ao que foi pago, verifica-se se algum arquivo do validador está na pasta FTP (onde o SAGA move os arquivos a serem enviados à RioCard). Se houver algum arquivo, a célula é pintada de vermelho. Se não houver, o valor pago está de fato igual ao valor dos arquivos, o que indica um erro de digitação da arrecadação. A célula será pintada de amarelo.

O último caso é quando o valor da soma dos BINs não é igual ao que foi pago. Isso ocorre quando a RioCard pagou um arquivo como zerado (ele possui transações, mas no sistema caiu como se ele não possuísse). O programa pinta a célula de verde. Neste caso, é necessário entrar em contato por telefone com a RioCard, que irá remover o registro de pagamento de seu banco de dados e reprocessará o arquivo.

A organização do novo SBE_Diff pode ser vista no diagrama de classes da Figura 5.12:

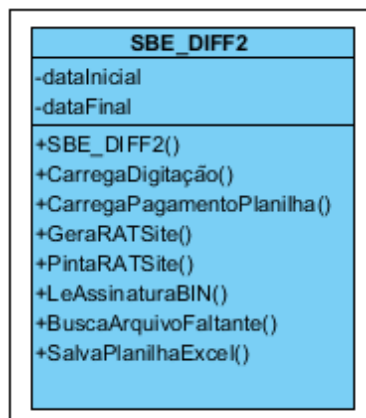


Figura 5.12: Diagrama de classes do novo SBE_DIFF

O funcionamento do programa é representado no diagrama de atividades da Figura 5.13:

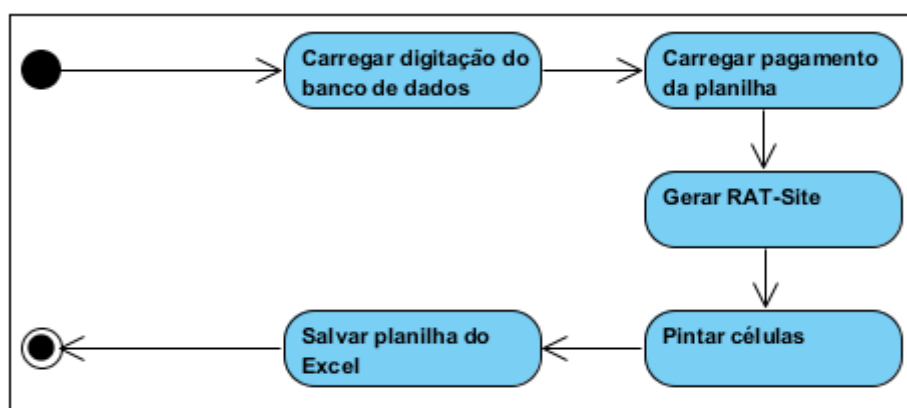


Figura 5.13: Diagrama de atividades do novo SBE_DIFF

5.4.4: BU_DIFF

O BU_DIFF deve mostrar os erros de ressarcimento do governo. Para o BU, não há o processo de digitação pela Arrecadação. Então, a comparação é feita entre o que foi pago e o valor da importação dos arquivos.

Após a definição do período de análise pelo usuário, o programa irá buscar os dados de pagamento. Diferente do ressarcimento da RioCard, onde cada linha equivalia a um arquivo, a planilha de ressarcimento do governo possui uma linha

para cada transação onde houve integração. Assim, há quase um milhão de linhas, para o período de um mês.

Como o volume de dados é excessivo, o BU_DIFF irá buscar as linhas que estejam inseridas no período informado. Como a planilha está ordenada por data, é possível realizar uma pesquisa binária, que é um algoritmo de busca que funciona em tabelas ordenadas, testando o elemento central de um intervalo, eliminando a metade da tabela onde o valor desejado não se encontra, e repetindo este procedimento de forma iterativa [16]. Seu funcionamento é explicado no fluxograma da Figura 5.14:

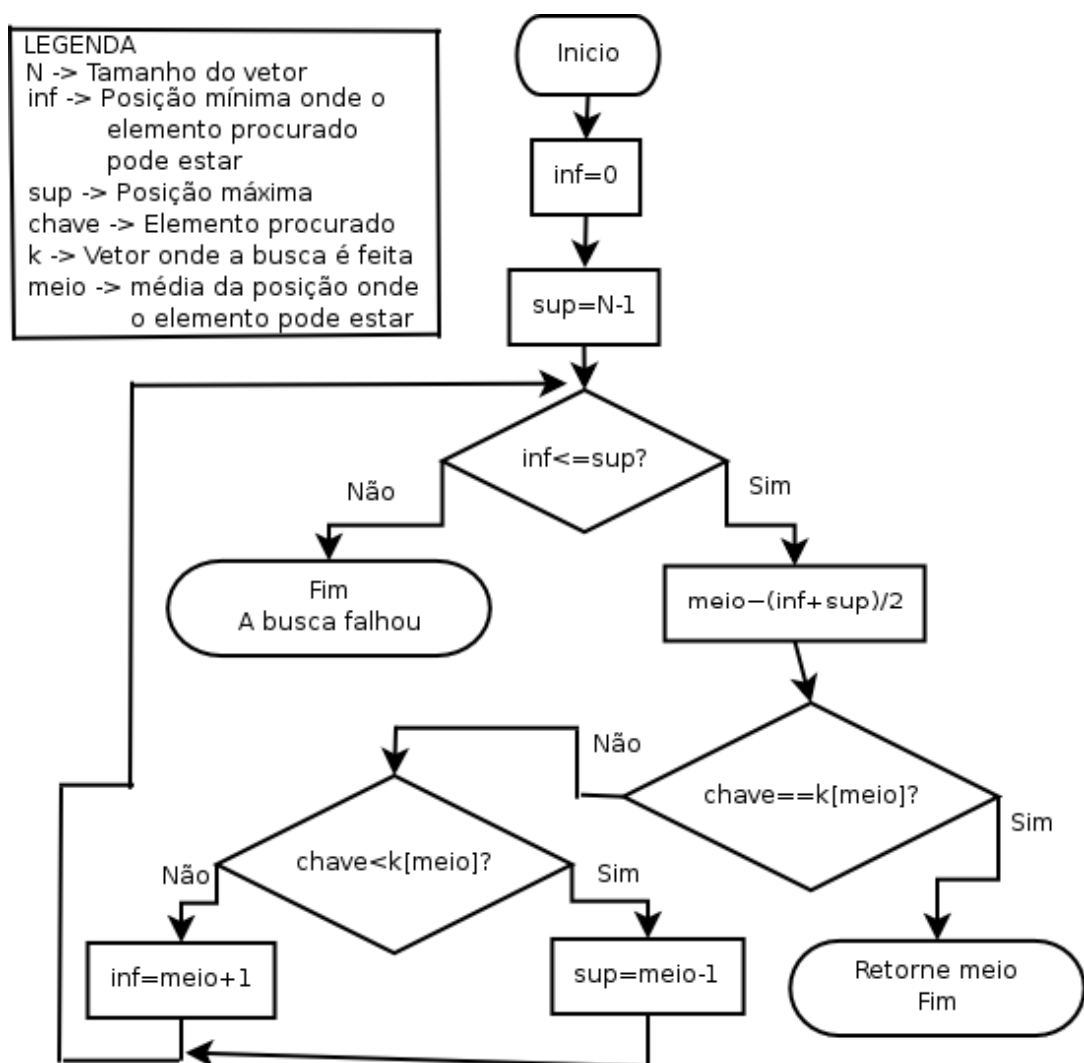


Figura 5.14: Fluxograma da pesquisa binária

Quando é encontrada uma linha “L” cuja data está no período desejado, o programa fará uma busca linear pela primeira linha com a data de início do período e outra pela última linha com a data final do período. Ambas as buscas lineares partirão da linha “L”. As buscas estão representadas nos fluxogramas das Figura 5.15 e Figura 5.16.

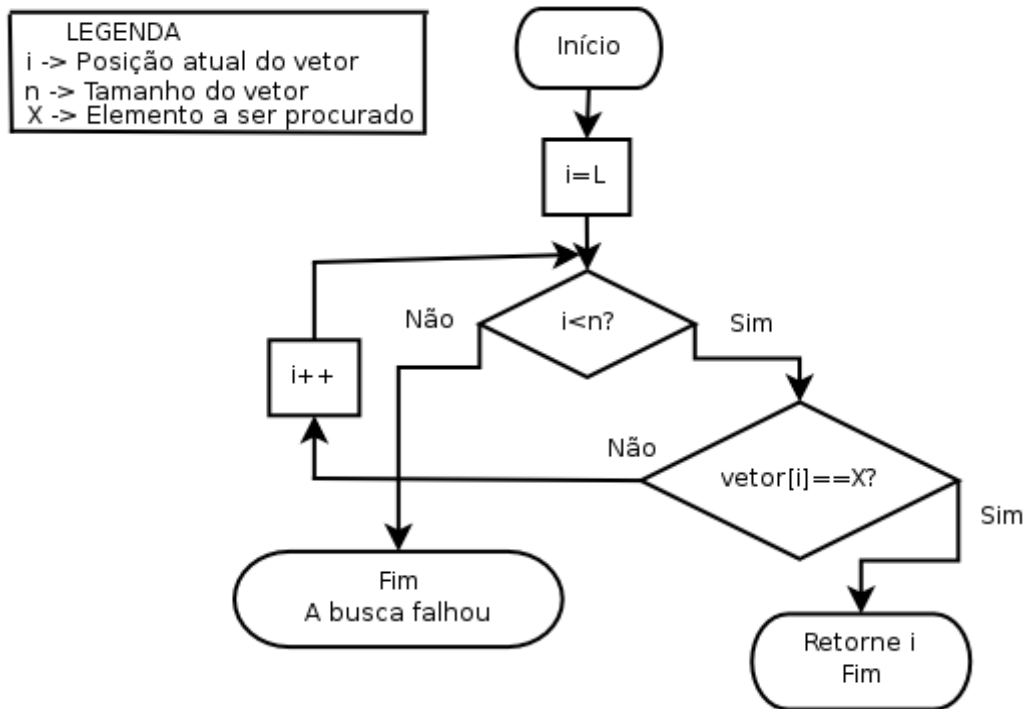


Figura 5.15: Fluxograma da pesquisa linear superior

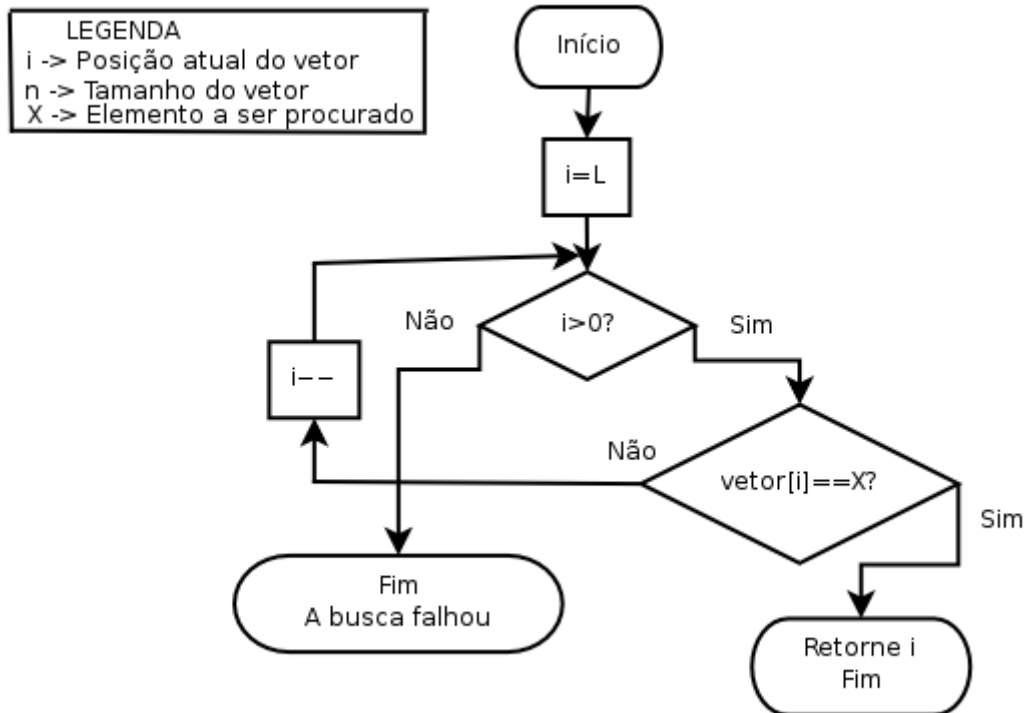


Figura 5.16: Fluxograma da pesquisa linear inferior

Após encontrar as linhas do período informado, elas são carregadas no programa. A seguir, é feita uma consulta no banco de dados, para carregar os dados de importação.

Com todos os dados disponíveis, o programa realiza a soma do ressarcimento e da digitação por dia, mostrando ao usuário (Figura 5.17).

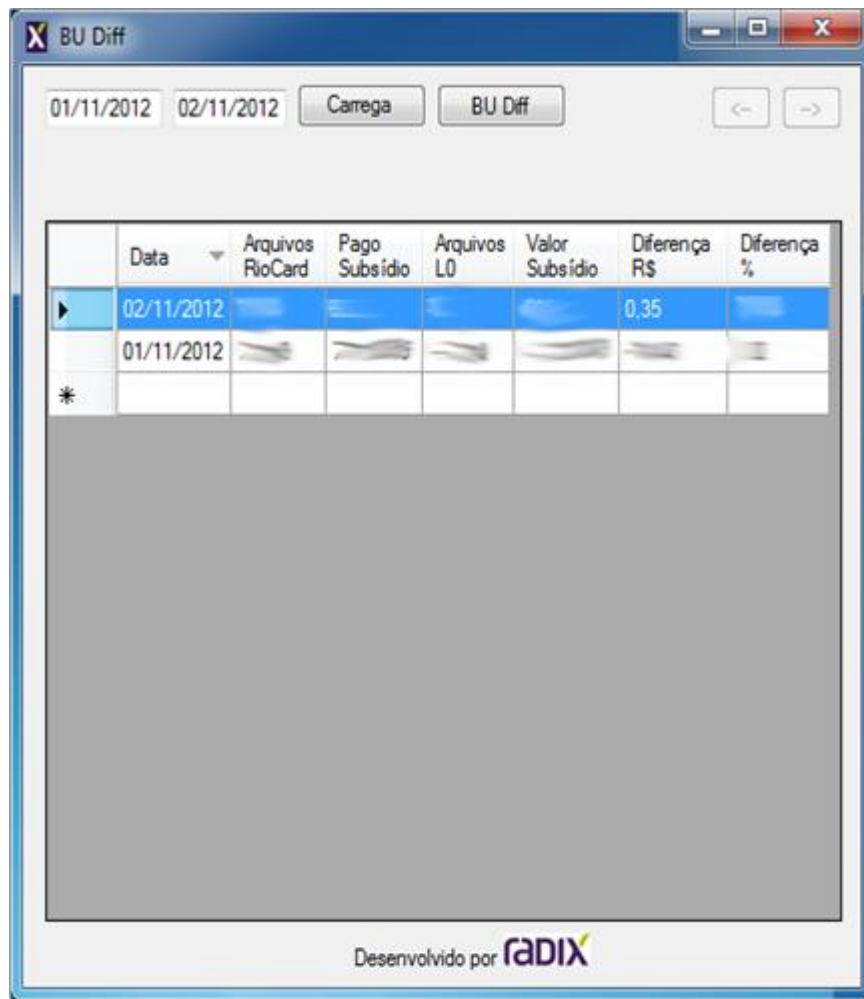


Figura 5.17: Interface do BU_DIFF

O usuário pode escolher uma data fazendo um duplo-clique na linha. O programa faz, então, a comparação dos dados de importação e de pagamento para cada validador na data escolhida, mostrando apenas os que contém erro (Figura 5.18).

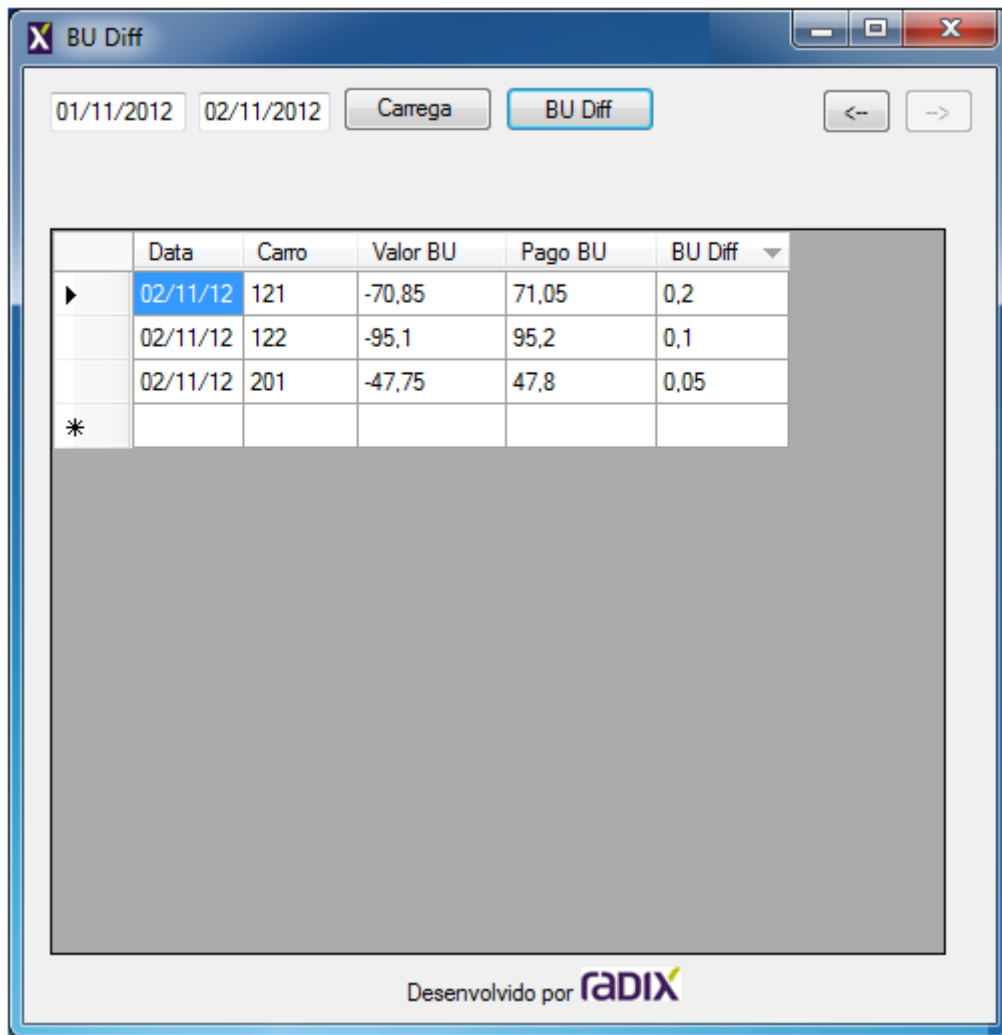


Figura 5.18: Tela detalhada do BU_DIFF

É possível, também, clicar no botão “BU DIFF” que realiza a comparação para o período completo, não apenas para uma data. A organização do BU_DIFF pode ser vista no diagrama de classes da Figura 5.19:

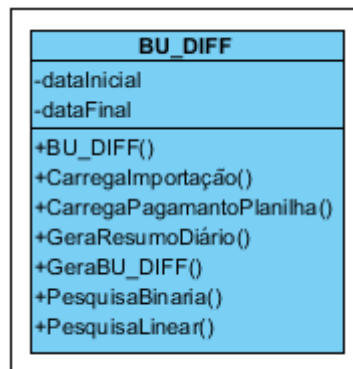


Figura 5.19: Diagrama de classe do BU_DIFF

O funcionamento do programa é representado no diagrama de atividades da Figura 5.20:

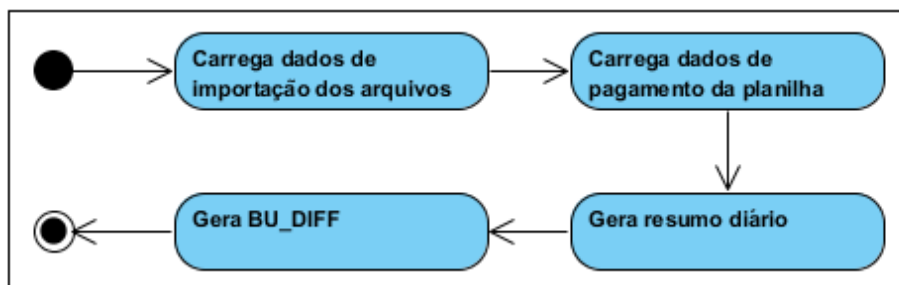


Figura 5.20: Diagrama de atividades do BU_DIFF

5.5: Ferramentas análise de comunicação

5.5.1: Resumo da Comunicação

O aplicativo realiza um conjunto de buscas nos arquivos texto dos logs dos servidores, para montar a planilha. Para obter as informações desejadas, o sistema faz uma leitura do texto, linha-a-linha, para encontrar os seguintes termos:

1. Número da Thread, no formato “<número>”. Ao encontrar, é dado início à leitura das linhas que contenham o número da Thread. A cada linha lida, é salva a data contida nela.
2. “Status: 6”, que indica o início de uma comunicação. Na linha deste termo há a data e a hora de início da comunicação.
3. “VAL”, que estará em conjunto com a sigla da estação e com o número do validador (por exemplo, CBLVAL10, que é o validador 10 da Central). O programa utiliza uma tabela de mapeamento para obter o nome da estação a partir da sigla.
4. “Desconectando... thread”, que indica o final da comunicação. A data da última linha lida será a data de fim da comunicação. Se o termo de desconexão não for encontrado no arquivo, a coluna “Finalizado” recebe o valor “NÃO”.

A Figura 5.21 mostra um exemplo de log:

```
Arquivo  Editar  Formatar  Exibir  Ajuda
<1868> - Iniciada a thread número: 1868
<1868> - [16/04/2012 21:35:56] Status: 6
<1868> - [16/04/2012 21:35:59] idPacote: 01_00016_0011 recebido com sucesso.
<1868> - [16/04/2012 21:36:18] Status: 8
<1868> - -----> Procurando pacote tipo 1 para enviar para o validador: d:
\empresa1\pacotes\enviar\01_00016_0011*.zip
<1868> - [16/04/2012 21:36:28] Status: 19
<2172> - Iniciada a thread número: 2172
<2172> - Desconectando... thread número: 2172
<2172> - Finalizada a thread número: 2172
<1868> - [16/04/2012 21:36:31] IDFETRANSPOR recebido:
26062006110400280820091430001204201210100012042012170700050420121946001304201214500
00504201219410029112010154400
<1868> - [16/04/2012 21:36:40] Status: 17
<1868> - -----> Procurando pacote tipo 7 para enviar para o validador: d:
\empresa1\pacotes\enviar\digicon_*.zip
<1868> - [16/04/2012 21:36:40] 01_00016_0011 versão não encontrada
<1868> - Desconectando... thread número: 1868
<1868> - Finalizada a thread número: 1868
<768> - Iniciada a thread número: 768
<768> - [16/04/2012 21:39:27] Status: 6
<3692> - Iniciada a thread número: 3692
<3692> - Desconectando... thread número: 3692
<3692> - Finalizada a thread número: 3692
<768> - [16/04/2012 21:39:33] idPacote: 01_00016_0011 recebido com sucesso.
<768> - [16/04/2012 21:39:53] Status: 8
<768> - -----> Procurando pacote tipo 1 para enviar para o validador: d:
\empresa1\pacotes\enviar\01_00016_0011*.zip
<768> - [16/04/2012 21:40:09] Status: 19
<768> - [16/04/2012 21:40:15] IDFETRANSPOR recebido:
26062006110400280820091430001204201210100012042012170700050420121946001304201214500
00504201219410029112010154400
<768> - [16/04/2012 21:40:28] Status: 17
<768> - -----> Procurando pacote tipo 7 para enviar para o validador: d:
\empresa1\pacotes\enviar\digicon_*.zip
<768> - [16/04/2012 21:40:28] 01_00016_0011 versão não encontrada
<768> - Desconectando... thread número: 768
<768> - Finalizada a thread número: 768
<1488> - Iniciada a thread número: 1488
```

Figura 5.21: Log da comunicação

Após passar pelo programa, uma planilha de comunicação é gerada (Figura 5.22). Podemos facilmente visualizar a comunicação por validador, vendo se ela chegou ao fim ou não.

ResumoComunicacao28.xlsx - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K
1	Thread	Sigla	Ramal	Estacao	Val	Data_Abertur	Hora_Abe	Data_Fecham	Hora_Fech	Duração	Finalizado
2	10872	ATNVAL04	Japeri	AUSTIN (46)	4	10/09/2012	18:23:21	10/09/2012	18:39:10	00:15:49	
3	7096	QMOVAL07	Japeri	QUEIMADOS (47)	7	10/09/2012	18:42:20	10/09/2012	19:12:25	00:30:05	
4	10888	CSSVAL03	Japeri	COMENDADOR SOA	3	10/09/2012	18:46:14	10/09/2012	19:04:08	00:17:54	
5	11396	NIUVAL08	Japeri	NOVA IGUAÇU (44)	8	10/09/2012	18:49:49	10/09/2012	19:07:27	00:17:38	
6	12164	NLSVAL02	Japeri	NILÓPOLIS (40)	2	10/09/2012	18:52:53	10/09/2012	19:36:50	00:43:57	
7	10280	SARVAL01	Saracuruna	Saracuruna (82)	1	10/09/2012	20:00:30	10/09/2012	20:41:13	00:40:43	
8	10744	SCOVAL10	Deodoro	São Cristóvão (4)	10	10/09/2012	20:28:17	10/09/2012	20:36:23	00:08:06	
9	3668	SCOVAL10	Deodoro	São Cristóvão (4)	10	10/09/2012	21:28:13	10/09/2012	22:14:35	00:46:22	
10	11076	PCIVAL02	Japeri	PARACAMBI (51)	2	10/09/2012	21:32:56	10/09/2012	22:17:05	00:44:09	
11	5696	BRXVAL01	Belford Roxo	Belford Roxo (66)	1	10/09/2012	21:42:35	10/09/2012	21:43:33	00:00:58	
12	10256	BRXVAL04	Belford Roxo	Belford Roxo (66)	4	10/09/2012	21:42:46	10/09/2012	22:03:48	00:21:02	
13	10704	MRAVAL06	Deodoro	Madureira (16)	6	10/09/2012	22:08:23	10/09/2012	22:17:04	00:08:41	NÃO
14	11836	MRAVAL09	Deodoro	Madureira (16)	9	10/09/2012	22:08:46	10/09/2012	22:22:04	00:13:18	NÃO
15	12004	APOVAL02	Belford Roxo	Agostinho Porto (6	2	10/09/2012	22:13:08	10/09/2012	22:18:03	00:04:55	
16	10608	PVAVAL02	Belford Roxo	Pavuna (62)	2	10/09/2012	22:17:56	10/09/2012	22:21:25	00:03:29	
17	10440	SCZVAL04	Santa Cruz	Santa Cruz (36)	4	10/09/2012	22:23:20	10/09/2012	22:31:26	00:08:06	NÃO
18	11704	SCZVAL01	Santa Cruz	Santa Cruz (36)	1	10/09/2012	22:25:49	10/09/2012	22:31:58	00:06:09	NÃO
19	10620	BGUVAL05	Santa Cruz	BANGU (26)	5	10/09/2012	22:53:53	10/09/2012	22:54:10	00:00:17	
20	11520	CDAVAL01	Deodoro	Cascadura (15)	1	10/09/2012	22:53:54	10/09/2012	23:03:05	00:09:11	
21	6332	EDOVAL02	Deodoro	Engenho de Dentro	2	10/09/2012	22:53:55	10/09/2012	23:32:56	00:39:01	
22	8080	EDOVAL05	Deodoro	Engenho de Dentro	5	10/09/2012	22:53:56	10/09/2012	23:36:02	00:42:06	
23	7968	EDOVAL08	Deodoro	Engenho de Dentro	8	10/09/2012	22:54:00	10/09/2012	23:36:26	00:42:26	
24	3624	JRIVAL03	Japeri	Japeri (49)	3	10/09/2012	22:54:11	10/09/2012	23:12:38	00:18:27	

Figura 5.22: Planilha de resumo de comunicação

O funcionamento do programa é representado no diagrama de atividades da Figura 5.23

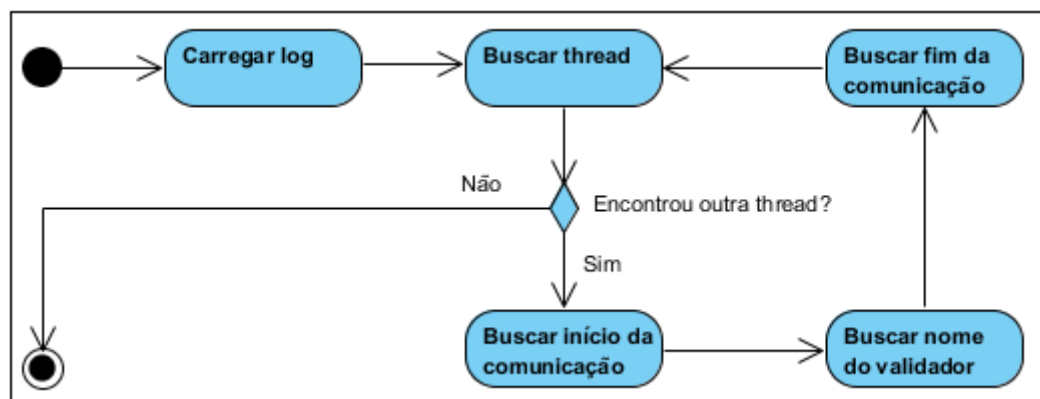


Figura 5.23: Diagrama de atividades do ResCom.exe

5.5.2: Comunicação com servidores

Este aplicativo tentará abrir um conexão via socket com os servidores de coleta, em intervalos fixos. Os dados de falha ou sucesso na conexão serão salvos e um gráfico exibirá as informações.

5.5.3: Testador de servidores para Android

Este aplicativo simples para o celular possui apenas três botões, um para cada servidor. Ao clicar em um botão, o programa tentará abrir um socket numa porta no servidor (a mesma porta configurada no valparam). Se for bem sucedido, aparecerá uma mensagem de “OK”. Senão, será exibido o erro que ocorreu (por exemplo, erro de “timeout”, ou falha de “IO”, que ocorre quando o celular está sem sinal).

A Figura 5.24 mostra a interface gráfica do aplicativo:

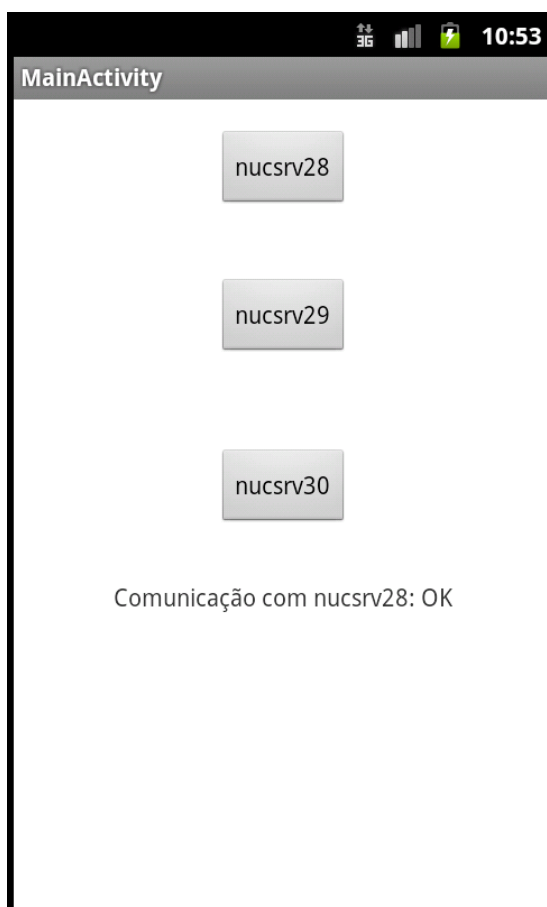


Figura 5.24: Tela do aplicativo de teste de comunicação para Android

5.5.4: Comunicação atualizada no SOUBE

Quando o usuário deseja verificar quais validadores já comunicaram na data atual, a qualquer hora do dia, ele deve acessar esta tela. O SOUBE realizará uma consulta no banco de dados do SAGA, verificando quais validadores já enviaram ao menos um arquivo “BIN”.

A Figura 5.25 mostra a tela criada, cujo desenvolvimento utilizou a plataforma “ASP.NET”:

Estação	21/01/2013
Praça da Bandeira 1	NÃO
Praça da Bandeira 2	NÃO
Praça da Bandeira 3	NÃO
Praça da Bandeira 4	NÃO
São Cristóvão 1	NÃO
São Cristóvão 2	NÃO
São Cristóvão 3	NÃO
São Cristóvão 4	NÃO
Mangueira 1	OK
Mangueira 2	OK
São Francisco Xavier 1	NÃO

Figura 5.25: Tela de comunicação do SOUBE

5.6: Separador de arquivos bins

A primeira tentativa de implementação deste programa carregava todos os arquivos “BIN” e “ZIP” de todas as subpastas do diretório “C:\SBE” para um “array”. Em seguida, todos os “BIN” seriam copiados para a pasta “C:\SBE\Bins”, e os arquivos “BIN” contidos nos “ZIP” seriam extraídos para a mesma pasta. Porém,

dado o grande volume de dados, a listagem de arquivos ocupava a memória inteira do computador, e o programa travava.

Para resolver este problema, foi necessário utilizar o conceito de recursividade. Funções recursivas são aquelas que chamam a si mesmas, até que uma condição de parada seja atingida. É uma forma de se criar algoritmos para atingir objetivos complexos, embora o consumo de memória torne o programa ineficiente [17]. Porém, um comando no C# que faz uma limpeza na memória (“GC.Collect()”), contorna este problema.

A função “CopiaArquivos” recebe como parâmetro apenas o caminho de uma pasta. Ela irá carregar o nome de todos os “BIN” da pasta em um “array” e copiar os arquivos para a pasta “C:\SBE\Bins”. Os arquivos “ZIP” também são listados mas neste caso, os “BIN” são extraídos de dentro dos “ZIP”.

Os “arrays” de “BIN” e “ZIP” são igualados a “null” após a cópia dos arquivos, o que libera os recursos associados aos “arrays”. A função “Garbage Collector” é chamada, removendo da memória todos os recursos não mais utilizados. Este procedimento foi o grande diferencial para a primeira implementação, permitindo que a memória do programa não ultrapassasse 20MB (diferente do outro que atingia 2GB e o programa travava).

Em seguida, as subpastas diretas do diretório são carregadas em outro “array”. Para cada uma das subpastas, o aplicativo chama, de forma recursiva, o próprio método “CopiaArquivos”.

A condição de parada é que o “array” de subpastas tenha tamanho zero, indicando que o procedimento não deve continuar.

Quando todos os BINs estiverem em uma única pasta, sem repetições, o procedimento de separação de corrompidos inicia. É feita uma tentativa de ler a assinatura digital do arquivo. Se a tentativa não for concluída com êxito, o arquivo é separado na pasta “C:\SBE\Corrompidos”.

A organização do SeparadorDeBins pode ser vista no diagrama de classes da Figura 5.26:

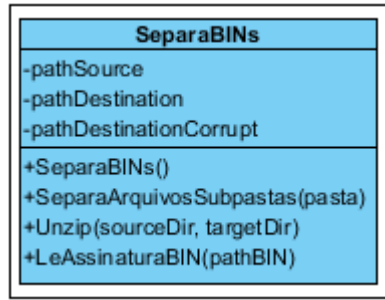


Figura 5.26: Diagrama de classes do SeparadorDeBINs

O funcionamento do programa é representado no diagrama de atividades da Figura 5.27:

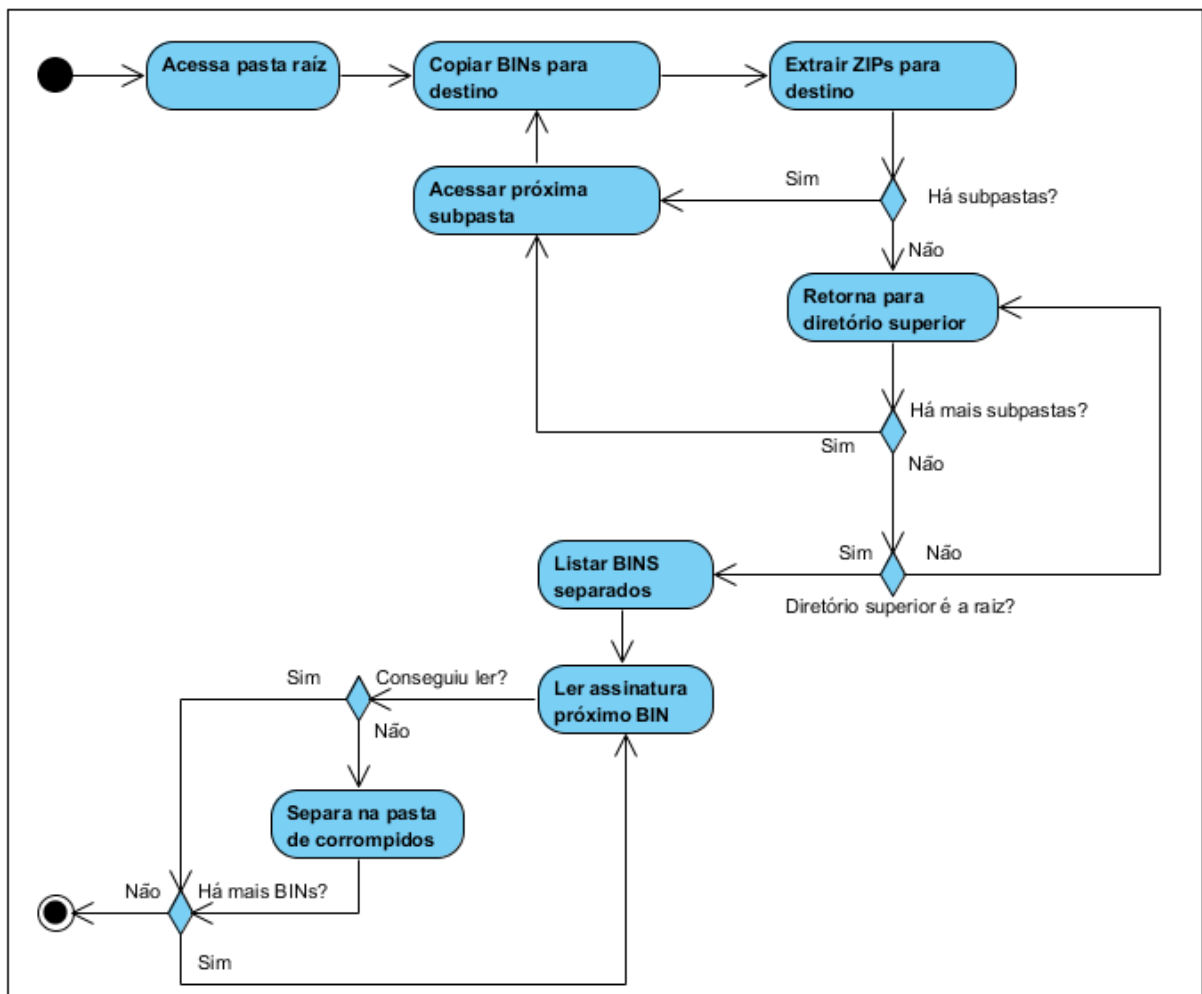


Figura 5.27: Diagrama de atividades do SeparadorDeBINs

5.7: Indicadores do SBE

O programa de indicadores possui três módulos, descritos a seguir. A organização do aplicativo completo é mostrada no diagrama de classes da Figura 5.28:

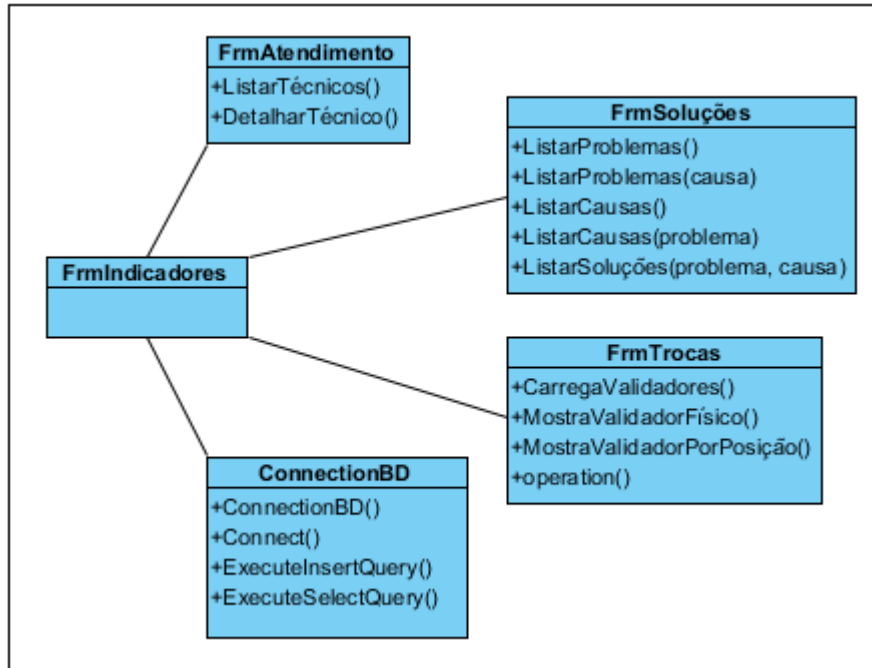


Figura 5.28: Diagrama de classes do IndicadoresSBE

5.7.1: Gráfico de Trocas

Esta tela (Figura 5.29) busca na base de dados do SABE todas as trocas de validador que ocorreram. O sistema efetuará uma consulta, considerando apenas o período escolhido. Uma “combobox” será populada com todos os validadores, ordenando por número de trocas no período ou por nota.

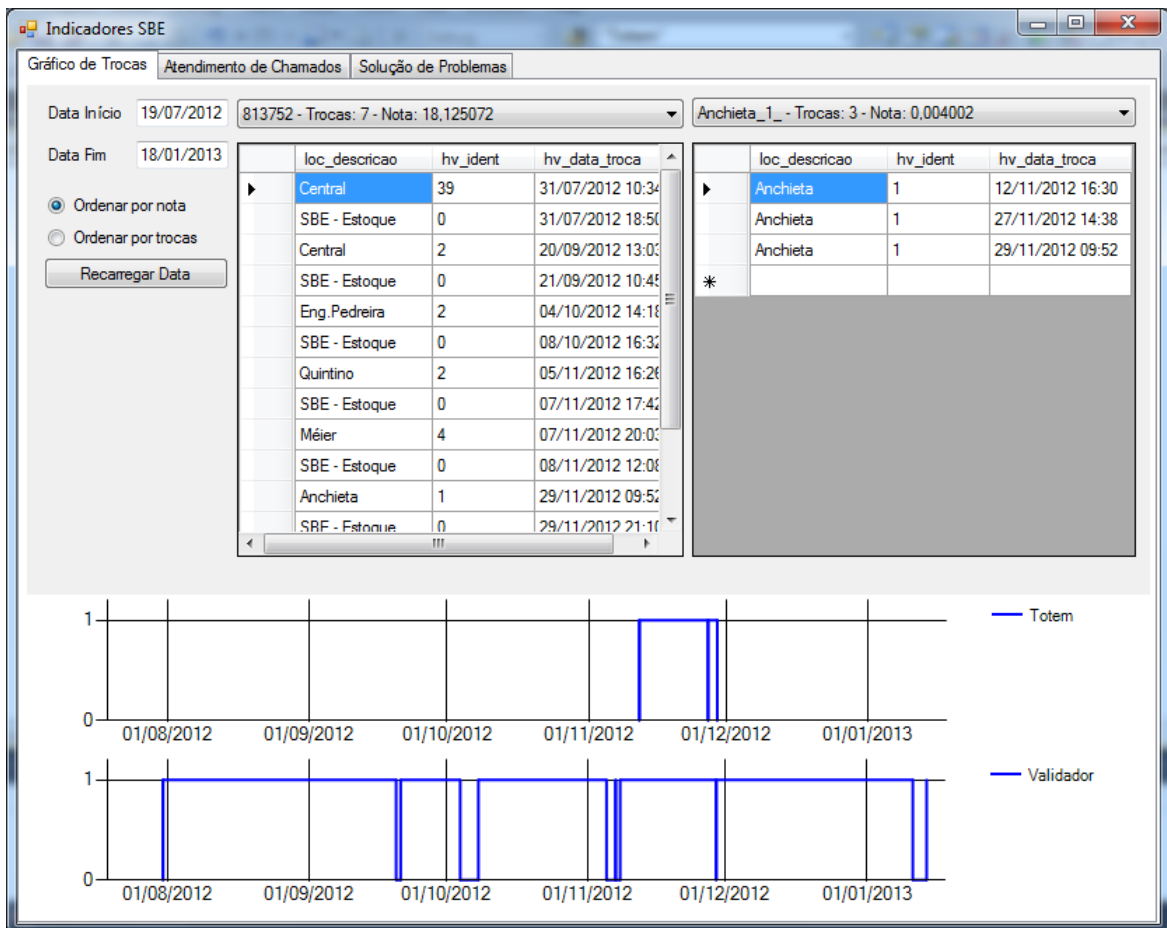


Figura 5.29: Tela de troca de validadores

A nota é um parâmetro de qualidade de um validador. Uma nota alta representa um validador de baixa qualidade. Ela é calculada utilizando a seguinte fórmula:

$$Nota = \text{NúmeroTrocas} \cdot \sum \frac{DataTroca - DataInicial}{DataFinal - DataInicial}$$

Da fórmula tiramos que quanto maior o número de trocas, pior é o validador. Porém, o termo do somatório insere um peso nas trocas. Se ela foi realizada em uma data próxima do início do período, seu peso será baixo. Se a troca for realizada mais para o final, seu peso será maior. A seguir segue um exemplo.

Consideraremos dois validadores:

- Validador 1: trocado três vezes, no primeiro dia de Março, Abril e Maio;

- Validador 2: trocado duas vezes, no primeiro dia de Novembro e Dezembro.

Calculando a nota para o validador 1:

$$Nota = 3. \left(\frac{2}{12} + \frac{3}{12} + \frac{4}{12} \right) = 2,25$$

Calculando a nota para o validador 2:

$$Nota = 2. \left(\frac{10}{12} + \frac{11}{12} \right) = 3,5$$

Podemos observar que o validador 1, embora tenha mais trocas, teve uma nota menor que o validador 2. Então, o validador 2 é considerado pior, por ter apresentado falhas em uma data mais próxima da atual.

Tendo a lista ordenada por nota ou número de trocas, o usuário pode selecionar um dos validadores (físico ou por posição). Na Figura 5.29, escolheu-se o validador físico 813752 e o validador da estação Anchieta, posição 1.

O gráfico mostra quando o validador esteve em campo ($y=1$) e quando esteve no SBE ($y=0$). Cada transição de 1 para 0 no gráfico indica uma falha onde foi necessário levar o validador ao SBE. Uma transição de 0 para 1 mostra quando o validador retornou para campo.

5.7.2: Atendimento de chamados

Esta tela executará uma “query” no banco de dados do SABE, buscando todos os chamados, por técnico, e a quantidade de vezes que houve uma reincidência. Para isso, a “query” consulta o total de chamados de um técnico em cada estação (se for maior que 1, houve reincidência). É mostrada a taxa de reincidência, em porcentagem do total de chamados atendidos. Ela deve ser menor ou igual a 20 para todos os técnicos. Ao selecionar um dos técnicos, o programa mostrará uma tabela detalhando seus chamados.

Na tela da Figura 5.30, podemos observar que o técnico “Leandro” atendeu 8 chamados, sendo que todos eles são para validadores diferentes. Então, não houve reincidência.

O programa permite ainda a filtragem dos chamados por tipo de equipamento (validador, PDV, entre outros).

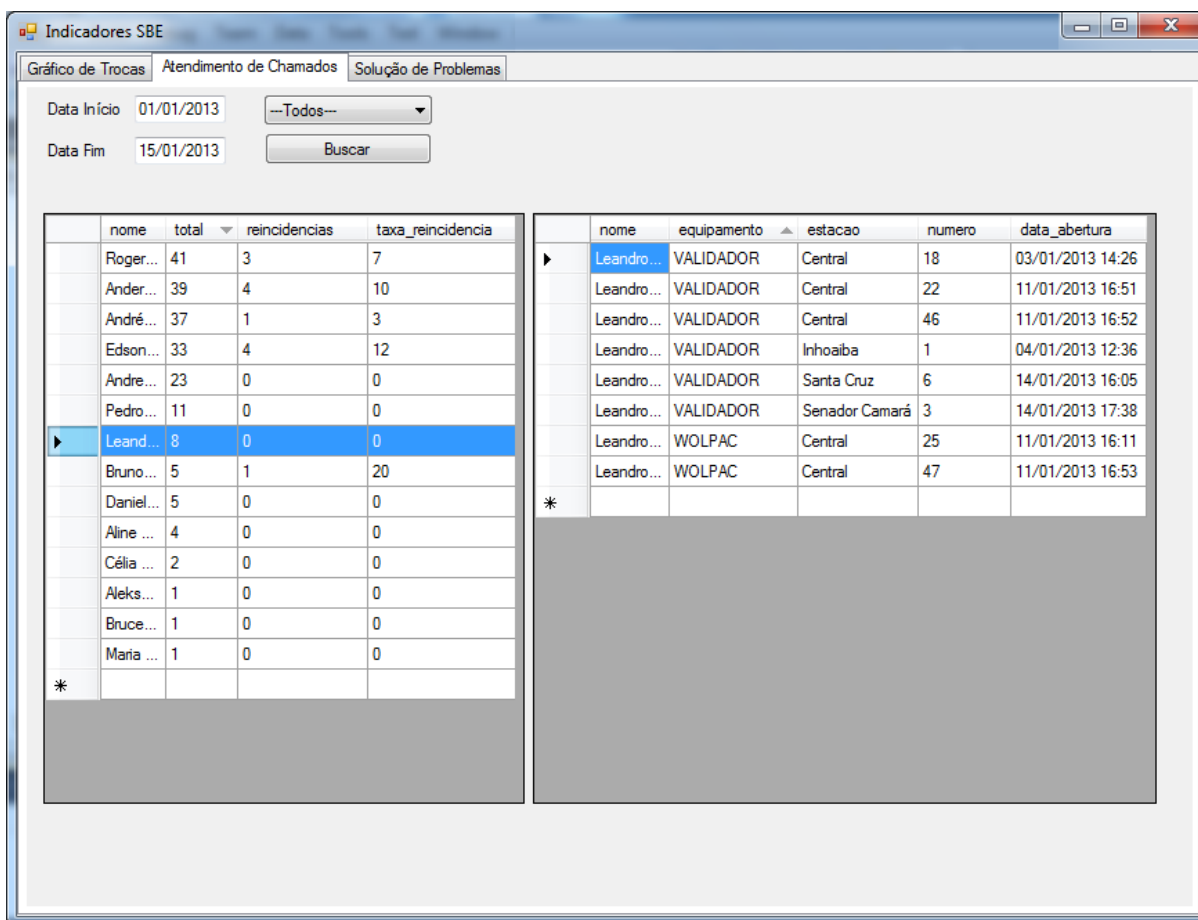


Figura 5.30: Tela de atendimento de chamados

5.7.3: Busca de solução

Na tela de busca de soluções (Figura 5.31), o usuário indica a data inicial da análise. O programa busca no banco de dados do SABE todos os problemas ocorridos desde a data informada até a data atual. A tabela é ordenada pelo número de ocorrências do problema no período.

Em seguida, o usuário deve selecionar um dos problemas. O programa informará quais as causas que provocaram a falha no período.

Por fim, o usuário seleciona uma das causas e observa as soluções possíveis para o problema.

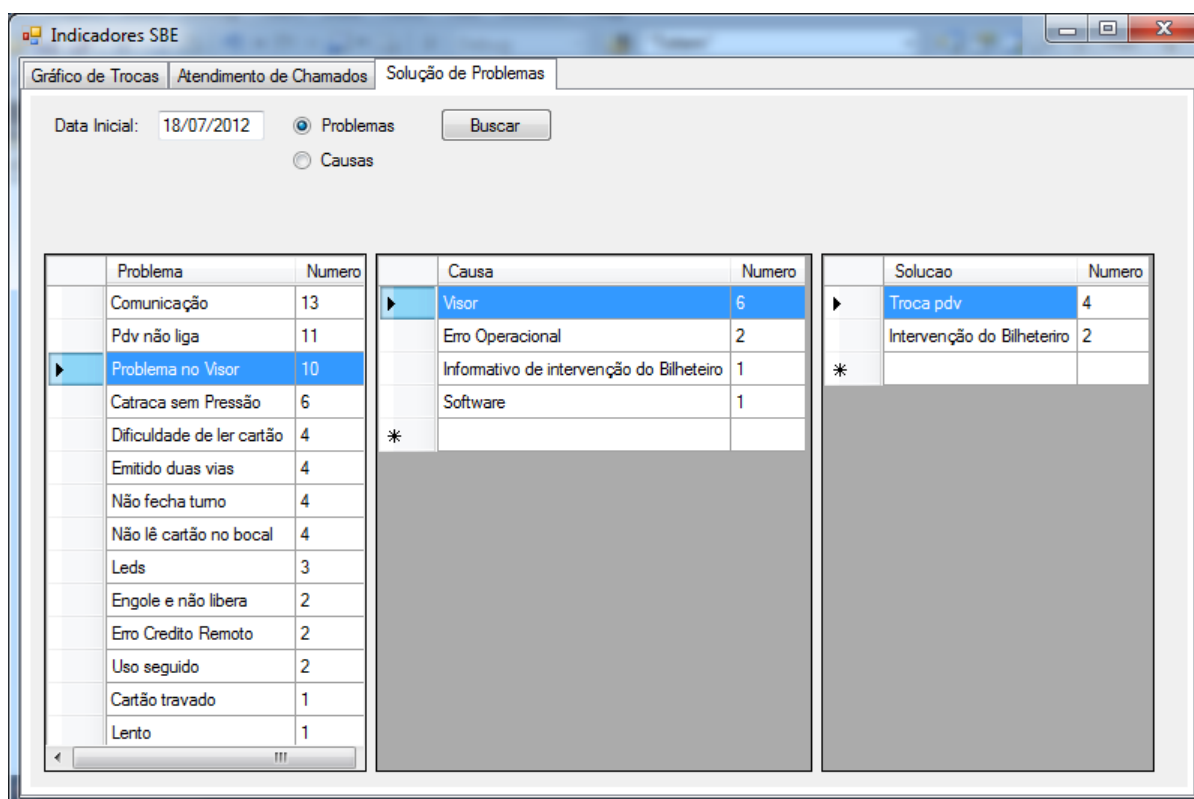


Figura 5.31: Tela de solução de problemas

No exemplo da figura, para um “Problema no Visor”, há quatro possíveis causas. Para a causa “Visor”, há duas soluções, que são “Troca de PDV” ou “Intervenção do Bilheteiro”.

O programa mostra o número de vezes que cada solução foi adotada. Assim, é possível verificar qual solução tem a maior probabilidade de ser adotada, que no exemplo é a troca do PDV.

5.8: Considerações finais

A utilização do Microsoft Visual Studio facilitou a implementação das soluções para os problemas identificados. As funções prontas da plataforma “.NET” aceleram muito o desenvolvimento.

Os programas de configuração de validadores, de automatização do ressarcimento, de análise da comunicação, de restauração de arquivos e de visualização de indicadores foram implementados, apresentando resultados que serão expostos no próximo capítulo.

Capítulo 6: Resultados

Com a implantação dos novos sistemas criados pelo autor deste relatório, para solucionar os problemas apresentados no Capítulo 3:, o SBE e a SuperVia obtiveram diversos ganhos. Neste capítulo serão mostrados os resultados obtidos pelo uso dos novos programas, que incluem, principalmente, redução do tempo da execução de tarefas por funcionários e entrada de dinheiro para a SuperVia.

6.1: Configuração de validadores

Os benefícios trazidos pelo novo programa de alteração do Valparam foram evidentes. Em um diálogo com os usuários, eles afirmam que o novo software não apresenta erros como o outro, além de ser muito mais simples de manusear. Eles estimam que a redução do tempo de configuração de um validador foi reduzida em aproximadamente 50%.

6.2: Ressarcimento

6.2.1: Arquivos Faltantes

Este novo sistema aperfeiçoou a atividade diária de coleta de arquivos, mostrando não só os validadores que não comunicaram como também a data de recebimento do último arquivo, sendo possível priorizar os mais atrasados.

Além disso, ele mostra onde houve pulos sequenciais. Assim, é possível coletar os arquivos sem depender da digitação pela arrecadação, que pode estar atrasada em 2 a 7 dias. Consequentemente, os arquivos serão enviados e pagos em um prazo menor.

6.2.2: Busca de SD

Com o novo aplicativo de busca de SDs, o tempo de verificação sobre o descarregamento dos cartões foi reduzido (embora um valor concreto não tenha sido obtido).

6.2.3: Novo SBE_DIFF

A geração da planilha do RAT-Site pelo novo modo de operar é muito mais rápida que pelo método antigo. O tempo que o novo programa leva para gerar a tabela, pintar as células e criar a planilha do RAT-Site é menor que o tempo que o SBE_DIFF antigo levava para gerar a tabela.

O procedimento de pintura da planilha do RAT-Site, quando realizado manualmente, levava até uma hora. Com o novo módulo de pintura automática, o procedimento foi substituído por um clique em um botão. O resultado foi a disponibilidade de uma hora a mais por dia para o funcionário desempenhar outras tarefas importantes, o que representa mais de 10% do seu tempo de trabalho.

6.2.4: BU_DIFF

O BU_Diff permitiu que a conferência de todos os erros de ressarcimento do governo fossem realizados rapidamente. A redução do tempo gasto pelo usuário foi de 97,5% (de 2 horas para apenas 3 minutos) para um mesmo período.

6.3: Ferramentas de análise de comunicação

6.3.1: Aplicativos de teste de comunicação

Com a planilha de comunicação e os dois aplicativos de teste dos servidores foi possível verificar que, durante o período de comunicação dos validadores (das 22h até às 2h), o maior número de falhas ocorrem entre 23h e meia-noite, que é o horário onde a maioria dos validadores tenta enviar os arquivos.

6.3.2: Comunicação atualizada no SOUBE

Com a tela de comunicação atualizada, é possível verificar o momento em que um validador comunicou. Dessa forma, quando um atendente força a comunicação, um funcionário do SBE verifica se a comunicação foi concretizada.

Caso o validador informe equivocadamente que a comunicação foi concluída com êxito (o que ocorre com certa frequência), o atendente pode forçar a comunicação novamente, ou coletar os arquivos por pen-drive ou SD.

O resultado é que a viagem do atendente até a estação não foi perdida, evitando que ele precise retornar no dia seguinte. O ganho de tempo para o atendente depende da distância da estação e da frequência dos trens nela. Para estações mais distantes, o ganho pode ser de até 2 horas.

6.4: Separador de arquivos bins

Antes do desenvolvimento deste aplicativo, o único método para descobrir o corrompimento de arquivos era abrindo-os, um a um, no “ListaArqLog”. Como a quantidade de arquivos é de mais de 500 mil, a separação manual era totalmente inviável. Com o novo software, os arquivos podem ser separados automaticamente.

A primeira parte dos arquivos já foi processada. Após a filtragem de arquivos BIN e a remoção dos duplicados, o espaço ocupado pelos arquivos passou de 250GB para 3GB. Eles foram repassados à RioCard por um pen-drive e estão sendo processados. Aproximadamente 0,1% dos arquivos dessa primeira parte (que equivale a 2 anos) estavam corrompidos, o que representa uma quantidade de dinheiro considerável para a SuperVia.

6.5: Indicadores do SBE

Este aplicativo ainda não foi muito utilizado, pois foi concluído há pouco tempo. Porém, em seu curto tempo de uso, já obteve alguns resultados interessantes.

6.5.1: Gráfico de trocas

Com este módulo, foi possível identificar três validadores que tem apresentado falhas com muita frequência nos últimos meses. Estes validadores foram trazidos ao SBE para que uma avaliação mais detalhada seja feita.

Os resultados esperados para o futuro são uma rápida identificação de validadores defeituosos e sua substituição por outros em bom estado.

Com isso, ocorrerão menos problemas em campo, e os técnicos ficarão menos sobrecarregados de tarefas.

6.5.2: Atendimento de chamados

O resultado deste módulo é a rápida identificação dos técnicos com melhor e pior desempenho no SBE. Este procedimento era feito mensalmente, de forma manual, por um funcionário do SBE. O tempo gasto era de aproximadamente um dia por mês (5% do tempo de trabalho do funcionário). Com o novo programa, a análise é feita em alguns segundos.

6.5.3: Busca de soluções

Espera-se que, utilizando a tela de busca de soluções, os técnicos consigam otimizar o tempo para a resolução de problemas.

Já foi possível identificar algumas relações entre problemas e soluções, embora nenhuma ação tenha sido tomada ainda.

Capítulo 7: Conclusões e Perspectivas

O resultados obtidos neste trabalho foram além dos esperados inicialmente. Quando o projeto foi idealizado, esperava-se desenvolver o SBE_DIFF otimizado, o BU_DIFF, o programa de arquivos faltantes, o de separação de arquivos e um dos aplicativos de análise de comunicação.

Além desses, foram desenvolvidos vários outros programas: de configuração de validadores, de localização de cartões SD, de teste de comunicação para Android, de indicadores de desempenho e de busca de soluções.

O novo método de configuração de validadores eliminou os erros do aplicativo anterior e reduziu o tempo exigido dos técnicos para esta tarefa em 50%. Por causa da sua interface amigável, o programa foi aceito rapidamente pelos técnicos, que se mostraram muito satisfeitos pela facilidade de utilizar o programa.

Os aplicativos de análise de comunicação, que testam a comunicação com os servidores de coleta, permitiram a identificação de horários de congestionamento da rede, o que impacta diretamente a comunicação.

As ferramentas de ressarcimento estão executando, automaticamente, a conferência entre o valor pago e o que deveria ter sido pago pela RioCard. Este procedimento não está mais sujeito a erros de seres humanos, além de poupar mais de 10% do tempo de trabalho de um funcionário.

A separação dos arquivos corrompidos trará um retorno financeiro considerável para a SuperVia. Há arquivos de até 5 anos atrás que não foram pagos por estarem corrompidos. Agora, eles serão reprocessados e pagos.

O programa de análise de indicadores auxiliou o SBE a identificar equipamentos defeituosos, técnicos mais eficientes e a encontrar soluções para problemas de forma mais rápida. Algumas dessas tarefas também exigiam tempo de um funcionário, mas agora são executadas automaticamente.

Resumindo, houve uma redução do tempo de execução de vários procedimentos no SBE, afetando grande parte de seus funcionários. Embora este

projeto tenha automatizado algumas tarefas, nenhum impacto social negativo foi gerado. Os funcionários continuaram trabalhando normalmente, mas sem desperdiçar tempo executando tarefas tediosas que um computador pode realizar.

Uma sugestão para trabalhos futuros é que a lista de arquivos faltantes seja enviada automaticamente, por mensagem, para os atendentes. Assim, o funcionário que hoje é responsável pela análise de comunicação e pela coleta, poderá focar mais na análise.

Outra proposta é a criação de um programa que informe automaticamente se um SD já está disponível para formatação, verificando a falta de arquivos.

Por fim, seria interessante unificar a maioria destas ferramentas em uma única aplicação. A ideia é adicioná-las ao SOUBE, permitindo a visualização de qualquer informação do SBE através da intranet da SuperVia.

Para o desenvolvimento deste projeto, o conhecimento adquirido no decorrer do curso de Engenharia de Controle e Automação foi de grande utilidade. As disciplinas de “Introdução à Informática para Automação”, “Fundamentos da Estrutura da Informação”, “Microprocessadores”, “Informática Industrial I” e “Informática Industrial II” foram a base de desenvolvimento de software para o autor deste relatório. A disciplina de “Metodologia para Desenvolvimento de Sistemas” foi importante para projetar os novos aplicativos que foram desenvolvidos. A matéria “Fundamentos de Sistemas de Banco de Dados” auxiliou na criação de aplicativos com acesso a bancos de dados. Por fim, a disciplina de “Redes de Computadores para Automação Industrial” permitiu ao autor criar programas que se comunicam com servidores por meio de redes.

A oportunidade de trabalho na Radix, uma das melhores empresas para se trabalhar no Brasil, proporcionou um grande crescimento profissional ao autor deste relatório. Ter sob sua responsabilidade todo o desenvolvimento de software do projeto e o ressarcimento foi uma experiência desafiadora e, ao mesmo tempo, gratificante, dados os vários resultados positivos.

Bibliografia:

[1] Sobre a Radix. Disponível em: <<http://www.radixeng.com.br/sobre/>>. Acesso em: 05 de Dezembro de 2012.

[2] Melhores Empresas para se Trabalhar. Disponível em: <<http://www.greatplacetowork.com.br/melhores-empresas/>>. Acesso em: 05 de Dezembro de 2012.

[3] Clientes da Radix. Disponível em: <<http://www.radixeng.com.br/clientes/>>. Acesso em 05/12/2012.

[4] Sobre a SuperVia. Disponível em: <<http://www.supervia.com.br/quemsomos.php>>. Acesso em 05/12/2012.

[5] F. Ribeiro, "Sobre o OcoMon", Disponível em: <<http://sourceforge.net/apps/mediawiki/ocomonphp/index.php?title=OcoMon>>. Acesso em 13/12/2012.

[6] R. J. Bates, "GPRS: General Packet Radio Service", Nova Iorque, Estados Unidos; McGraw-Hill, 2001

[7] L. Grossman, "Computer Literacy Tests: Are You Human?", Time Magazine, 2008. Disponível em: <<http://www.time.com/time/magazine/article/0,9171,1812084,00.html>> . Acesso em 19/01/2013.

[8] "Legislação do Estado do Rio de Janeiro". Disponível em: <<http://alerjln1.alerj.rj.gov.br/CONTLEI.NSF/c8aa0900025feef6032564ec0060dfff/76de911502259106832576a10060ff9f?OpenDocument>>. Acesso em: 19 de Dezembro de 2012.

[9] <<http://stackoverflow.com/>>

[10] Microsoft, ".Net Framework". Disponível em: <<http://msdn.microsoft.com/en-us/vstudio/aa496123.aspx>> . Acesso em 12/01/2013.

[11] Microsoft, "Introduction to ASP .NET". Disponível em: <[http://msdn.microsoft.com/en-us/library/4w3ex9c2\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/4w3ex9c2(v=vs.71).aspx)>. Acesso em 12/01/2013.

[12] Microsoft, "Visual Studio". Disponível em: <<http://msdn.microsoft.com/en-us/library/dd831853.aspx>>. Acesso em 12/01/2013.

[13] Microsoft, "C# Language Specification Version 4.0". Disponível em: <<http://go.microsoft.com/fwlink/?LinkId=199552>>. Acesso em 12/01/2013.

[14] "MySQL 5.1 Reference Manual". Disponível em: <<http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>>. Acesso em 12/01/2013.

[15] "About Android". Disponível em: <<http://www.android.com/about/>>. Acesso em 12/01/2013.

[16] MathWorld, "Binary Search". Disponível em: <<http://mathworld.wolfram.com/BinarySearch.html>>. Acesso em 12/01/2013.

[17] S. Schuller, "How to Rewrite Standard Recursion through a State Stack & Iteration", Setembro de 2006. Disponível em: <<http://www.saasblogs.com/software-development/how-to-rewrite-standard-recursion-through-a-state-stack-amp-iteration/>>. Acesso em 19/01/2013.