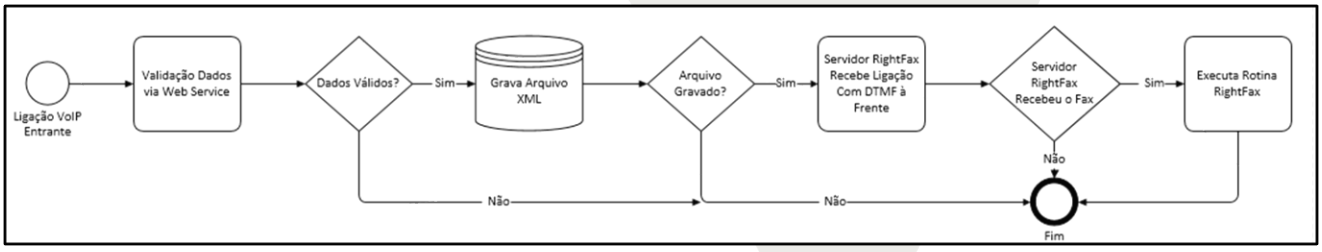
Projeto Desenvolvimento do Programa XML2RFAX - v0.03.000

1. Legendas dessa documentação:
   1.1. Sistema URA: São os servidores Microsoft Windows que integram o ambiente de telefonia VoIP, com Web Services para a consulta de dados digitados pelos clientes, com o objetivo de identificar os faxes antes de serem transmitidos, gerando arquivos XML de metadados com base na consulta Web Service;
   1.2. Sistema FAX: São os servidores Microsoft Windows que hospedam os servidores de fax, de forma a receber os faxes transmitidos pelos clientes, através do ambiente de telefonia VoIP. Faxes recebidos geram metadados hospedados em um banco de dados Microsoft SQL Server, além do arquivo de imagem eletrônica do fax;
   1.3. Sistema URA/FAX: É o conjunto de recursos compreendendo o Sistema URA mais o Sistema FAX;
   OBSERVAÇÃO: O Sistema URA/FAX é ainda dividido em ANTIGO e NOVO, onde o Sistema URA/FAX ANTIGO, se refere ao projeto atualmente em produção, que será substituído pelo Sistema URA/FAX NOVO. Esse projeto de desenvolvimento do programa XML2RFAX se refere ao Sistema URA/FAX NOVO. Contudo, essa documentação vai apresentar a programação da rotina antiga no Sistema URA/FAX ANTIGO, para facilitar o entendimento, a celeridade e o desenvolvimento do novo programa XML2RFAX.
   1.4. CONTRATADA ou DESENVOLVEDOR: É a empresa ou o analista contratado para desenvolver o programa XML2RFAX. Toda a plataforma de desenvolvimento, e recursos auxiliares para a programação, são de responsabilidade da CONTRATADA ou DESENVOLVEDOR;
   1.5. CONTRATANTE: É a empresa Telemikro, representante da OpenText™ no Brasil, cujo seu cliente necessita do referido programa. A Telemikro deve oferecer para a CONTRATADA ou DESENVOLVEDOR, as condições técnicas para a integração do programa com o Sistema FAX;
   1.6. RIGHTFAX: É o ambiente de testes do Sistema FAX, usado para o desenvolvimento e homologação do referido programa;
   1.7. FONTE: É todo o código fonte, e recursos auxiliares, utilizados para o desenvolvimento e compilação do programa, que serão de propriedade da Telemikro.
2. Objeto: O objeto desse projeto é dar condições ao desenvolvimento do programa XML2RFAX, e colocá-lo em produção no ambiente do Sistema URA/FAX NOVO do cliente da CONTRATANTE. De agora em diante, o programa XML2RFAX poderá ser chamado simplesmente de PROGRAMA.
3. Objetivo: O programa XML2RFAX é um integrador de metadados entre um arquivo XML, gerado por um Sistema URA, e os metadados gerados pelo recebimento de fax pelo Sistema FAX, com o servidor OpenText RightFax Fax Server™. O PROGRAMA também será responsável pela gerência de arquivos XML processados, e o encaminhamento dos faxes recebidos para seus usuários finais, dentro do ambiente do Sistema FAX. O processo de juntar os metadados e gravar as informações no fax recebido, chamamos de RENOMEAR o fax.
4. Diagrama de contexto do Sistema URA/FAX NOVO: De acordo com o diagrama abaixo, o PROGRAMA se encaixa no projeto do Sistema URA/FAX NOVO, na tarefa “Executa Rotina RightFax”.



4.1. Descrição do diagrama: O Sistema URA é o responsável por atender uma ligação telefônica, e fazer todo o fluxo do atendimento com prompt de voz, solicitando ao cliente que informe dados antes do recebimento do fax. Esses dados digitados são sempre números, como por exemplo, o número do CPF. O Sistema URA valida esses dados através de Web Services, e agrupa os metadados em um único arquivo XML. Como por exemplo, através da validação do número do CPF, coloca no arquivo XML o metadado com o nome do cliente, em vez do número do CPF. Após gerar e gravar em um diretório compartilhado esse arquivo XML, cujo nome é único, o Sistema URA transfere a ligação telefônica para o Sistema FAX. O Sistema FAX recebe do Sistema URA a informação de qual é o nome único do arquivo XML, que estará associado a esse fax que será recebido. Essa informação sobre o nome do arquivo XML é registrada nos metadados do Sistema FAX, que em seguida, recebe o fax. Após o fax ter sido recebido, o Sistema FAX disponibiliza a imagem do fax, junto com os metadados do processo de seu recebimento, em uma única caixa-postal de fax de um usuário do Sistema FAX. Essa caixa-postal de fax funciona de forma semelhante a uma caixa-postal de e-mail. O documento de fax recebido pode ser tratado pelo PROGRAMA, que vai juntar aos metadados do recebimento do fax, com as informações exclusivas do arquivo XML gerado pelo Sistema URA, através do nome do arquivo XML, gravado no metadados do Sistema FAX. Com os metadados completos, o PROGRAMA encaminha o fax recebido para a caixa-postal do usuário destino final no Sistema FAX. O PROGRAMA gerencia também os arquivos XML já processados, ou que tenham algum problema e precisam ser descartados, ou os que nunca foram analisados, acessando os diretórios compartilhados dos arquivos XML.

5. Plataformas e recursos necessários para o desenvolvimento do FONTE do PROGRAMA: São componentes desse ambiente de desenvolvimento e recursos operacionais:
   5.1. O PROGRAMA deve ser desenvolvido para ser instalado e operar em plataforma Microsoft Windows Server 2012 R2;
   5.2. O PROGRAMA será instalado em um único servidor do Sistema FAX;
   5.3. A instalação do PROGRAMA deve ser como um serviço Microsoft Windows, de execução AUTOMÁTICA, e o referido serviço será executado com uma conta de usuário de rede, inserida explicitamente no Grupo de Administradores do Sistema FAX. Com essas permissões de administrador, o PROGRAMA poderá ler, criar, executar, excluir, e editar, arquivos, e diretórios no servidor de fax, ou em locais na rede local autorizados;
      5.3.1. O PROGRAMA não terá nenhuma interface sistêmica com o operador ou usuário do Windows Server, uma vez que sua execução é como um serviço Windows;
      5.3.2. Caso a CONTRATADA ou DESENVOLVEDOR necessite de uma interface de programa, para efeitos de análise de erros, por exemplo, o acesso a essa interface deve ser manual, quando o referido serviço Windows estiver parado. Exemplo: Abrir uma linha de comando MSDOS, ir ao diretório de instalação do PROGRAMA, e executar o programa com um parâmetro. Exemplo: “XML2RFAX -debug”.
   5.4. O PROGRAMA deverá trabalhar acessando os arquivos XML hospedados na rede local, bem como terá acesso aos recursos de gerência sobre faxes recebidos dentro do ambiente do Sistema FAX;
   5.5. O programa pode ser desenvolvido com qualquer linguagem de programação, e recursos auxiliares, sabendo que:
      5.5.1. A CONTRATADA ou DESENVOLVEDOR precisa integrar os metadados gerados pelo Sistema URA/FAX, utilizando os seguintes recursos disponibilizados pela CONTRATANTE, baseados em uma biblioteca de objetos, métodos, e propriedades, chamada “RightFax COM API”, compatível com o suporte à programação em linguagem C, C++, Java, JScript, Visual Basic, VBScript, Delphi, PowerBuilder e linguagens baseadas em .NET como C# e VB.NET. Essa biblioteca é nativa no servidor de fax, baseada em componentes Windows 64/32-bit, e contida em uma única DLL (Dynamic Link Library), chamada “RFCOMAPI.DLL”;

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

5.5.2. A CONTRATANTE vai disponibilizar uma sorte de documentações sobre a RFCOMAPI.DLL, incluindo alguns exemplos de programação;

5.5.3. Um servidor OpenText RightFax Fax Server v16EP2 será disponibilizado em versão de demonstração, para testes e homologação do programa. Esse servidor de fax estará instalado no ambiente da CONTRATANTE, e a CONTRATADA ou DESENVOLVEDOR, terá acesso remoto ao ambiente computacional onde se localiza esse servidor de fax. O acesso remoto a esse ambiente seguirá normas de segurança a serem definidas pelos administradores da CONTRATANTE;

5.5.4. Quaisquer outros recursos necessários para o desenvolvimento do programa, que não fazem parte do sistema operacional Windows Server, ou dos recursos do OpenText RightFax Fax Server, devem ser fornecidos pela CONTRATADA ou DESENVOLVEDOR, sem custos para a CONTRATANTE;

5.5.4.1. A CONTRATADA ou DESENVOLVEDOR deve informar a CONTRATANTE sobre todos os recursos necessários utilizados para o desenvolvimento e compilação do PROGRAMA.

5.5.5. Após a homologação do programa pela COMTRATANTE, a CONTRATADA ou DESENVOLVEDOR, poderá participar da instalação remota do PROGRAMA no ambiente do cliente final da CONTRATANTE, para o monitoramento e diagnósticos de problemas, e possíveis manutenção no PROGRAMA, para que seja observado o melhor desempenho e a satisfação do cliente final com o PROGRAMA.

6. A CONTRATADA ou DESENVOLVEDOR deverá manter registro da FONTE do PROGRAMA e suas versões, disponibilizando uma cópia para a CONTRATANTE, de forma a manter as versões do PROGRAMA em ambiente de homologação e de produção final no cliente, igual ao FONTE entregue a CONTRATANTE;

6.1. A CONTRATADA ou DESENVOLVEDOR deverá seguir o padrão de criação de versão de software, de acordo com as seguintes regras semânticas adotadas como padrão: https://semver.org/lang/pt-BR/spec/v2.0.0.html e https://semver.org/lang/pt-BR/spec/v2.0.0-rc.1.html;

6.2. De acordo com as regras semânticas acima adotadas as seguintes legendas serão utilizadas, no formato X.XX.XXX:

6.2.1. Versão MAJOR é o primeiro conjunto a esquerda “X.”: Começa em “0.”. É incrementado para “1.” Quando a primeira versão do PROGRAMA for homologada para ser instalada no ambiente de produção do cliente final;

6.2.2. Versão MINOR é o segundo conjunto “.XX.”: Começa em “.01.”. É incrementada sempre que a CONTRATADA ou DESENVOLVEDOR, modificar o FONTE do PROGRAMA, de acordo com as regras semânticas;

6.2.3. Versão PATCH é o terceiro conjunto “.XXX”: Começa em “.000”. É incrementada sempre que a CONTRATADA ou DESENVOLVEDOR, modificar o FONTE do PROGRAMA, de acordo com as regras semânticas;

6.2.4. É permitido o uso de rótulos semânticos adicionais como build, alpha, beta, candidate e outros, de acordo com as regras semânticas acima estabelecidas;

6.2.5. Fica estabelecido um novo rótulo semântico adicional chamado DOC, que deve ser usado obrigatoriamente como último parâmetro de versão, indicando que essa versão do PROGRAMA possui comentários internos novos, ou alterados, para o entendimento das execuções das rotinas, métodos e procedimentos programados:

6.2.5.1. Essas alterações, ou inclusão de comentários no FONTE, não obrigam a instalação do PROGRAMA, nos ambientes de homologação, ou de produção no cliente final, mas a versão do PROGRAMA deve receber um incremento numérico, na parte que se refere a versão do DOC.

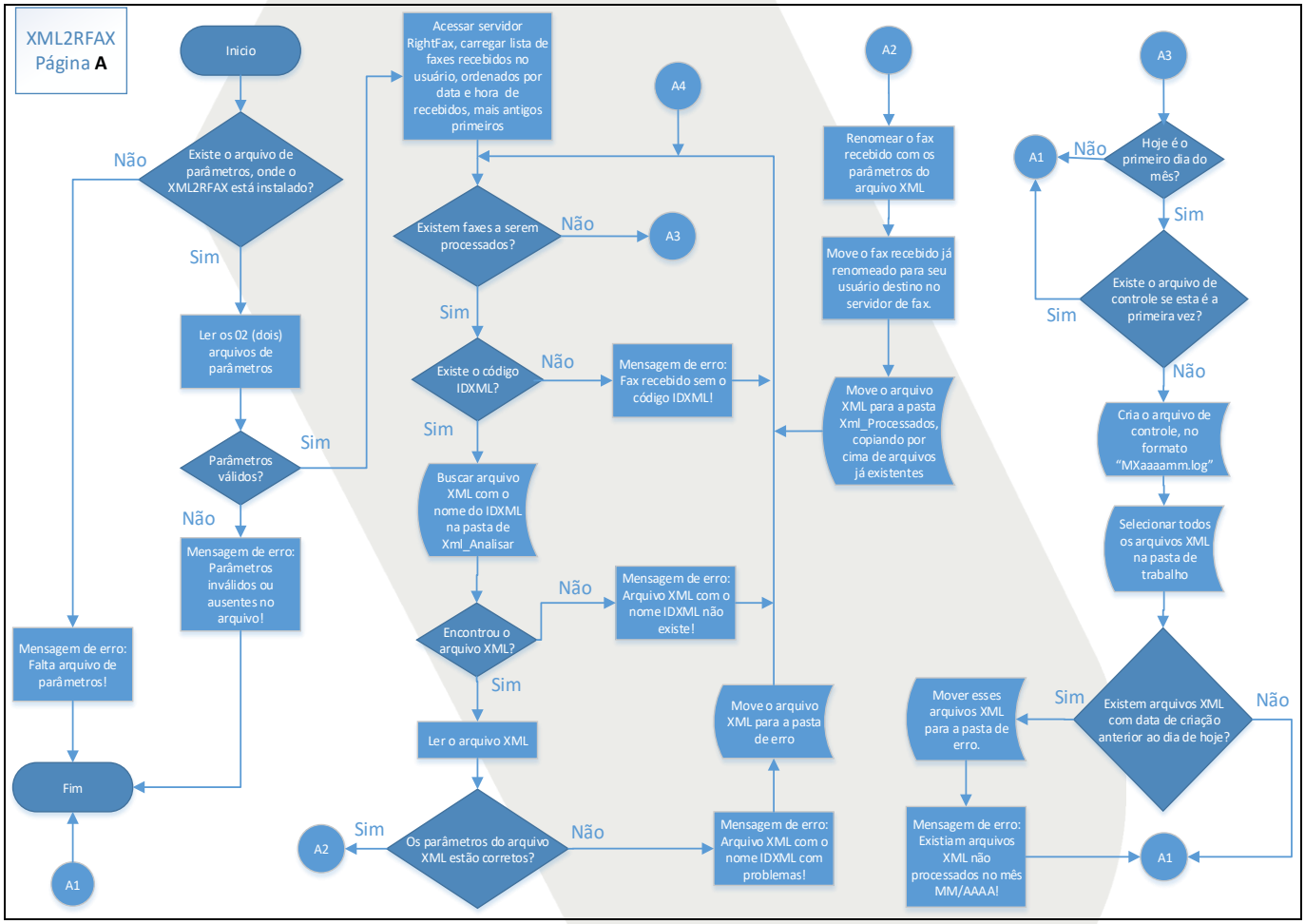
6.2.5.2. O parâmetro DOC pode ser excluído da versão do PROGRAMA, assim que uma versão MAJOR, MINOR ou PATCH é lançada, uma vez que as alterações de comentários que gerarão a versão DOC, deverão ser incluídas na nova versão do FONTE.

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

6.3. O FONTE deve ter comentários sobre a sua programação, para o entendimento das execuções das rotinas, métodos e procedimentos programados, de forma que a CONTRATANTE possa entender o fluxo da programação usado;

6.4. A CONTRATANTE pode solicitar a CONTRATADA ou DESENVOLVEDOR, que comente alguma parte do FONTE do PROGRAMA, a qualquer tempo, para que possa, no seu entendimento, esclarecer sobre as execuções das rotinas, métodos e procedimentos programados. Essa solicitação obriga a modificação da versão do FONTE do PROGRAMA, conforme estabelecido no item 6.2.5.

7. Para o desenvolvimento do PROGRAMA, a CONTRATADA ou DESENVOLVEDOR, deverá seguir o fluxo idealizado abaixo. Esse fluxo do programa não compreende detalhes de programação, que vão ser explorados nessa documentação:



7.1. Principais etapas do fluxo de programação do PROGRAMA:

7.1.1. Entrar em execução como serviço no Windows;

7.1.2. Ler os arquivos de parâmetro para a sua execução;

7.1.3. Enquanto existirem faxes recebidos na caixa-postal de fax do usuário de trabalho;

7.1.3.1. Encontrar no diretório compartilhado o arquivo XML associado ao fax recebido, e ler os metadados;

7.1.3.2. Juntar os metadados do arquivo XML com os do fax recebido, e o RENOMEAR;

7.1.3.3. Enviar e mover o fax recebido já RENOMEADO, para o usuário de destino;

7.1.3.4. Gerenciar o arquivo XML processado.

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

7.1.4. Todo primeiro dia do mês corrente, fazer a manutenção sobre arquivos XML que não foram processados no mês anterior;

7.1.5. Aguardar um INTERVALO de tempo, para executar novamente a rotina de fluxo do PROGRAMA.

8. A CONTRATADA ou DESENVOLVEDOR pode com bom senso, e com o aceite da CONTRATANTE, propor modificações nesse fluxo, desde que o objetivo deste projeto com o PROGRAMA, seja alcançado;
9. Sobre as mensagens de erros e testes de falhas:
   9.1. No fluxo desse PROGRAMA estão sendo previstos algumas condições de testes de falhas básicas, que vão gerar mensagens indicando o erro ou falha detectada. Outras condições de falhas avançadas podem ser incluídas pela CONTRATADA ou DESENVOLVEDOR, garantindo a execução do fluxo do programa;
   9.2. Todas as condições de teste de erro do PROGRAMA, obrigatoriamente, devem gerar mensagens no Microsoft Windows Event Viewer, não sendo necessário janelas de pop-up, ou a geração de mensagens em arquivos de LOG;
   9.3. Algumas mensagens de erro podem gerar a interrupção ou não do fluxo do PROGRAMA, mas nenhuma geração de mensagem de erro deve interromper a execução do serviço Windows do PROGRAMA:
      9.3.1. Quando o fluxo do PROGRAMA terminar, por causa de alguma mensagem de erro severa gerada, o PROGRAMA deve aguardar o próximo INTERVALO de tempo determinado, para recomeçar o fluxo do PROGRAMA novamente. Exemplo: Falta do arquivo de parâmetros de iniciação do PROGRAMA, ou falta algum parâmetro nesse arquivo. Nessa condição de erro é impossível continuar o fluxo do PROGRAMA, mas o PROGRAMA deve continuar a enviar mensagens para o Windows Event Viewer, até que um operador do Sistema URA/FAX, manualmente resolva o problema;
      9.3.2. Quando o PROGRAMA gera uma mensagem de erro de alerta, o fluxo do PROGRAMA é redirecionado, e continua a executar o fluxo. Exemplo: Um fax recebido não tem o código IDXML, que o relaciona com um arquivo XML. Nessa condição, após a geração da mensagem de erro, o fluxo do PROGRAMA pode continuar a trabalhar no próximo fax recebido a ser analisado.
10. Detalhamento do fluxo do PROGRAMA:
   10.1. O PROGRAMA quando é executado, vai sempre ler um arquivo do tipo texto puro ou XML, em modo apenas de leitura, contendo os primeiros parâmetros, para a sua execução:
   OBSERVAÇÃO: Essa documentação será baseada em arquivo de parametrização em texto puro, mas a CONTRATADA ou DESENVOLVEDOR, pode usar a seu critério, um arquivo tipo XML, para esse arquivo de parametrização inicial.
      10.1.1. A localização desse arquivo de parametrização inicial é sempre no mesmo diretório de instalação do PROGRAMA.
      10.1.2. O PROGRAMA deve validar a existência do primeiro arquivo de parâmetros e o seu conteúdo: Caso esse primeiro arquivo de parâmetro não exista, falte algum parâmetro ou campo, ou o valor do conteúdo do campo esteja inválido, uma mensagem de erro é gerada no Windows Event Viewer, e após 5 (cinco) minutos – valor de tempo fixo, o PROGRAMA tentará verificar o conteúdo do arquivo de parâmetros novamente. É possível que nessa condição de erro, a geração de mensagens de erro continue, até a intervenção manual de um operador do Sistema URA/FAX seja necessária para corrigir o problema;
      10.1.3. O nome desse primeiro arquivo de parametrização é fixo: “Config_XML2RFAX.ini” se for texto puro, ou “Config_XML2RFAX.xml” se for XML;
      10.1.4. São campos desse arquivo de parametrização “Config_XML2RFAX.ini”, separados por um caractere “;” (ponto e vírgula):
         10.1.4.1. Campo (1) – INTERVALO: Tipo valor numérico. É a quantidade de tempo em minutos para a próxima execução do PROGRAMA, com valor padrão inicial de 1 (minuto): Sabendo que o PROGRAMA é um serviço Windows, sua execução será baseada em um tempo de intervalo em

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

minutos inteiros, após a sua última execução. Exemplo: O serviço Windows é iniciado uma primeira vez, o fluxo do PROGRAMA é acionado, e após o término do fluxo, um novo intervalo de tempo de 01 (um) minuto é aguardado para a execução do PROGRAMA novamente. Esse fluxo de execução do PROGRAMA é interrompido apenas quando o serviço Windows é manualmente parado;

10.1.4.2. Campo (2) - FILECONFIG: Tipo string. É a URL que aponta a localização na rede local, do nome do arquivo de parâmetro utilizado em comum, por todos os recursos do Sistema URA/FAX. Esse arquivo é por definição um arquivo texto puro, e sua estrutura será analisada nesta documentação. Exemplo: ““\\MZ-CW-AP-067\BSURA\parametros.ini”;

10.1.4.3. Exemplo de um conteúdo do primeiro arquivo de parâmetros “Config_XML2RFAX.ini” (sem as aspas duplas): “1; \\MZ-CW-AP-067\BSURA\parametros.ini”

IMPORTANTE: Caso a CONTRATADA ou DESENVOLVEDOR necessite de mais informações para o correto processamento do PROGRAMA, esse arquivo de configuração poderá ser modificado, para agregar mais campos, desde que em comum acordo com a CONTRATANTE.

10.2. O segundo arquivo de parâmetros se encontra no local definido pelo campo FILECONFIG, e deve ser aberto em modo apenas leitura:

10.2.1. O PROGRAMA deve validar a existência deste segundo arquivo de parâmetro e o seu conteúdo: Caso esse segundo arquivo de parâmetro não exista, falte algum parâmetro ou campo, ou o valor do conteúdo do campo esteja inválido, uma mensagem de erro é gerada no Windows Event Viewer, e após 5 (cinco) minutos – valor de tempo fixo, o PROGRAMA tentará verificar o conteúdo do arquivo de parâmetros novamente. É possível que nessa condição de erro, a geração de mensagens de erro continue, até a intervenção manual de um operador do Sistema URA/FAX seja necessária para corrigir o problema;

10.2.2. No ANEXO 1 nesse documento, temos a descrição detalhada desse segundo arquivo de parâmetro, e um exemplo válido de seu conteúdo. A documentação e estrutura desse segundo arquivo de parâmetros é compartilhada por todos os envolvidos no Sistema URA/FAX, e não pode ser modificada;

10.2.3. Nessa parte da documentação vamos explorar somente os campos que interessam ao PROGRAMA:

10.2.3.1. Campo (2) UserRF: Para o acesso ao servidor de fax, um usuário precisa ser autenticado. Esse usuário tem acesso a caixa-postal de fax, que contêm os faxes recebidos para análise dos metadados;

10.2.3.2. Campo (3) ServerRF: É o nome do servidor de fax, onde se encontra o usuário;

10.2.3.3. Campo (4) UserPassRF: É a senha para autenticação e acesso ao usuário que se encontra nesse servidor de fax;

10.2.3.4. Campo (8) DirXmlAnalisar: Caminho de rede da pasta “Xml_Analisar”. É o diretório onde o Sistema URA grava os arquivos XML, com os metadados a serem utilizados na integração com os metadados dos faxes recebidos, na conta do usuário que se encontra no servidor de fax;

10.2.3.5. Campo (9) DirXmlDescartados: Caminho de rede da pasta “Xml_Descartados”. É o diretório onde o PROGRAMA deverá mover os arquivos XML com problemas, e que não foram por algum motivo utilizados;

10.2.3.6. Campo (10) DirXmlProcessados: Caminho de rede da pasta “Xml_Processados”. É o diretório onde o PROGRAMA deverá mover os arquivos XML que foram analisados, e os metadados foram utilizados na integração com os metadados dos faxes recebidos, na conta do usuário que se encontra no servidor de fax.

10.3. Utilizando a RFCOMAPI, com os parâmetros capturados no arquivo FILECONFIG, o PROGRAMA acessa o servidor de fax, entra na caixa-postal de fax do usuário autenticado e configurado no arquivo FILECONFIG, e

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

lê todos os metadados que necessita, para saber de cada fax recebido, quais são os arquivos XML relacionado com eles;

10.4. Existindo faxes recebidos a serem processados, o PROGRAMA acessa a pasta DirXmlAnalisar, e procura pelo arquivo XML que tem o mesmo nome do código IDXML do fax recebido. Exemplo: Se o código IDXML encontrado no fax recebido é “99012”, procurar no DirXmlAnalisar o arquivo “99012.xml”;

10.5. Não existindo mais faxes recebidos com IDXML, o fluxo termina, para recomeçar no próximo INTERVALO de tempo;

10.6. Não existindo um arquivo XML com o nome IDXML, uma mensagem de erro é emitida no Windows Event Viewer, e o fluxo recomeça no passo 10.4, seguindo para o próximo fax recebido;

10.7. Leia-se em modo apenas leitura o arquivo XML, captura o conteúdo das TAGs, e valida se estão com problemas, como por exemplo, se a TAG está vazia. Uma mensagem de erro e emitida no Windows Event Viewer, e o fluxo recomeça no passo 10.4, seguindo para o próximo fax recebido;

10.8. Os metadados do arquivo XML são unidos aos metadados do fax recebido, e o fax é movido para a caixa-postal do usuário destino do fax. O fax que será movido, após juntar os metadados, é chamado de fax RENOMEADO;

10.9. O arquivo XML é movido para a pasta DirXmlProcessados em “Xml_Processados”, sobrescrevendo qualquer arquivo XML que exista com o mesmo nome;

10.10. Ao terminar de processar todos os faxes recebidos encontrados, é necessário saber se está na hora de fazer a manutenção dos arquivos XML que estão órfãos, isso é, o arquivo XML foi gerado, mas o fax não foi recebido:

10.10.1. A manutenção só acontece uma vez no primeiro dia do mês. Se este não é o primeiro dia do mês, o fluxo do PROGRAMA termina, para recomeçar no próximo INTERVALO de tempo;

10.10.2. Precisamos saber se a manutenção dos arquivos XML já não foi feita, evitando a sua repetição. Para isso, é criado um arquivo de controle com o nome “MXaaaamm.log”, onde “MX” é fixo e significa que a **M**anutenção dos arquivos **X**ML já foi executada nesse dia, em que, “aaaa” é o ano da manutenção, e “mm” é o mês corrente da manutenção;

10.10.2.1. A CONTRATADA ou DESENVOLVEDOR, pode a seu critério, gravar nesse arquivo quaisquer informações que julgue relevante e importante para o registro do processamento dessa rotina, como por exemplo, a lista de arquivos XML processados na manutenção, a data final de encerramento da operação, e outras informações, como referência para uma futura análise de problemas. Não existe obrigação de fazer isso, pois a simples existência desse arquivo de controle é o suficiente para o fluxo do PROGRAMA.

10.10.3. Se o arquivo “MXaaaamm.log” já existe, então a manutenção já foi feita no primeiro dia do mês, e fluxo do PROGRAMA termina, para recomeçar no próximo INTERVALO de tempo;

10.10.4. Se o arquivo de controle não existe, cria-se o arquivo de controle “MXaaaamm.log”;

10.10.4.1. Executaremos a manutenção dos arquivos XML órfãos: Mover todos os arquivos XML encontrados no subdiretório “Xml_Analisar”, com data de criação anterior à data de hoje, para a pasta DirXmlDescartados em “Xml_Descartados”, sobre escrevendo possíveis arquivos XML existentes com o mesmo nome, gerando uma mensagem de erro emitida no Windows Event Viewer, informando que durante a manutenção mensal, foram encontrados arquivos XML gerados sem faxes correspondentes, e o fluxo do PROGRAMA termina, para recomeçar no próximo INTERVALO de tempo;

10.11. É possível existir um fax recebido com um código IDXML válido, mas não existir um arquivo XML para relacionar com esse fax, e a causa desse problema pode ser de telefonia, no momento que o Sistema URA tentou gravar o IDXML, com o nome do arquivo XML que precisa estar associado ao fax recebido. Nesse caso, uma mensagem de erro é gerada no Windows Event Viewer informando esse erro, e o PROGRAMA

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

continua no próximo fax a ser analisado. É possível que nessa condição de erro, a geração de mensagens de erro continue, até a intervenção manual de um operador do Sistema URA/FAX seja necessária para corrigir o problema;

10.12. É possível existir um fax recebido sem um código IDXML, uma vez que algum problema de telefonia, pode ter causado o problema. Nesse caso, uma mensagem de erro é gerada no Windows Event Viewer informando esse erro, e o PROGRAMA continua no próximo fax a ser analisado. É possível que nessa condição de erro, a geração de mensagens de erro continue, até a intervenção manual de um operador do Sistema URA/FAX seja necessária para corrigir o problema;

10.13. É possível duas condições para existirem arquivos XML que ainda não foram analisados:

10.13.1. Uma condição de erro é que arquivos XML foram criados, mas o transmissor do fax não transmitiu algum documento, ou aconteceu um problema durante a transmissão do fax, e este fax não chegou a ser recebido. Nesse caso, este arquivo XML é considerado órfão de fax;

10.13.2. Outra condição de erro para existirem arquivos XML anda não analisados, é porque os faxes ainda não terminaram de serem recebidos, e o transmissor continua a enviar documentos por fax. Esses não são arquivos XML órfãos, pois na próxima vez que o PROGRAMA for executado, o fax já pode ter sido totalmente recebido, e então será processado. Observe que um fax pode ter várias páginas de documentos, e receber todas as páginas pode demorar vários minutos, mesmo que o arquivo XML tenha sido criado a muito tempo atrás. Por isso é importante o parâmetro de INTERVALO de tempo entre as execuções do fluxo do PROGRAMA, para respeitar o tempo de recebimento do fax.;

10.14. O Sistema FAX será programado para excluir automaticamente os faxes analisados ou não, após 24 horas, em um horário determinado pelo administrador do Sistema URA/FAX. Dessa maneira, em um horário pré-estabelecido fixo, e seguro, faxes que estejam gerando mensagens de erro por problemas no IDXML, serão automaticamente excluídos da caixa-postal do usuário de trabalho.

11. O arquivo XML gravado pelo Sistema URA tem sempre o seguinte formato:

11.1. Nome do arquivo XML sempre com 6 (seis) dígitos, no formato “99XXXX.xml”, onde “99” é fixo, e os 4 (quatro) próximos dígitos são numéricos, indo de “0001” até “9999”.

11.2. Conteúdo dos metadados dentro do arquivo XML:

11.3. Primeira linha é a identificação do tipo do arquivo XML, e será sempre: <?xml version="1.0"?>;

11.4. A próxima linha contém várias TAG, como no exemplo:

<fax><codigoFax>990123</codigoFax><Billinfo1>82549</Billinfo1><RoutingCode>206468</RoutingCode><Billinfo2>724189001074031</Billinfo2><FromName>SERVICOS MEDICOS</FromName><Comments>APARECIDA SERAFINA BATALHA</Comments></fax>

11.4.1. TAG <fax>: Abre e fecha todos os campos de metadados a serem utilizados, e não podem ser vazios;

11.4.2. TAG <codigoFax>: É o nome do arquivo XML gerado pelo Sistema URA. Esse dado tem que ser igual ao IDXML, encontrado no metadado gravado no Sistema FAX. No exemplo, o arquivo é “990123.xml”;

11.4.3. TAG <Billinfo1>: É o código de identificação do remetente do fax. No exemplo, o código numérico “82549” é o identificador do transmissor do fax;

11.4.4. TAG <RoutingCode>: É o código da caixa-postal do usuário de destino do fax. Após juntar os metadados do arquivo XML com os metadados do fax recebido, todo o documento de fax já RENOMEADO, é encaminhado para esse usuário final. No exemplo, o Sistema URA atendeu a chamada no número de ramal do telefone “206468”, e é esse o usuário destino final do fax RENOMEADO;

11.4.5. TAG <Billinfo2>: É o código do beneficiário pelo fax transmitido. No exemplo, “724189001074031” é o código da carteira de identificação do beneficiado pelo documento de fax transmitido;

11.4.6. TAG <FromName>: É o nome do remetente ou referenciado, quem transmitiu o fax. No exemplo, “SERVICOS MEDICOS” é o nome do referenciado;

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

11.4.7. TAG < Comments >: É o nome do beneficiado pelo fax transmitido pelo referenciado. No exemplo, “APARECIDA SERAFINA BATALHA” é o nome sobre o qual o assunto do documento no fax recebido se trata.

12. Relacionamento de objetos nos metadados encontrados nos arquivos XML, e metadados de fax, que serão utilizados para o processo de RENOMEAR os documentos de faxes recebidos, com base na documentação da RFCOMAPI:

	Objetos no Servidor de Fax	Objetos no Arquivo XML
	DIDNumber	RoutingCode
	BillingCode1	Billinfo1
	BillingCode2	Billinfo2
	FromName	Comments
	UserComments	FromName
	ToName	RoutingCode
	ToFaxNumber	990000

OBSERVAÇÃO: O valor 990000 é fixo.

13. Principais métodos que serão utilizados para o processo de RENOMEAR os documentos de faxes recebidos, com base na documentação da RFCOMAPI:

13.1. Conectar ao servidor de fax:

```
Sub ConectarNoRightFax ()
    gblServer =  RFAXSERVER
    gblUserName =  RFAXUSER
    glbPassword = RFAXPASS

    Set gbl_FaxServer = CreateObject ("RFComAPI.FaxServer.1")
    gbl_FaxServer.ServerName = gblServer
    gbl_FaxServer.AuthorizationUserID = gblUserName
    gbl_FaxServer.AuthorizationUserPassword = glbPassword
    gbl_FaxServer.Protocol = 4
    gbl_FaxServer.UseNTAuthentication = false
    gbl_FaxServer.OpenServer()
End Sub
```

13.2. No ANEXO 2 temos um trecho de programação utilizado no Sistema URA ANTIGO, que será substituído pelo PROGRAMA novo. Observo que as instruções e métodos são comentados para a documentação do programa antigo.

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

ANEXO 1

1. Arquivo de parâmetros do Sistema URA/FAX no formato texto puro nome “Config_Bradesco.ini”:

1.1. Esse arquivo deve ser de comum acordo com a Telemikro, e ele terá todos os parâmetros necessários para a execução do Sistema URA/FAX, e suas integrações;

1.2. Cada linha do arquivo contém uma ou mais informações separadas pelo caractere “;” (ponto e vírgula), mas no último parâmetro da linha, esse caractere não existe;

1.3. Linhas no arquivo de parâmetro começando com um caractere “#” (chamado de jogo da velha, ou tralha), significa que a linha é um comentário, e que deve ser ignorada;

1.4. Todos os valores de campo contidos nesse arquivo são tratados como string.

2. Conforme definido no projeto, seguem os parâmetros de conteúdo desse arquivo:

2.1. Linha 1 do arquivo de parâmetro:

2.1.1. Ramal interno DDR, ou VDN, utilizado pelo Sistema URA/FAX para transferir a ligação do Sistema URA para com o Sistema FAX. Campo terá 06 dígitos numéricos;

2.1.2. Nome do usuário RightFax, que também é da caixa-postal de fax, onde a API RightFax deve encontrar os faxes recebidos pelo sistema URA/FAX. Campo chamado “RFAXUSER”, é alfanumérico e pode ter entre 03 até 15 dígitos;

2.1.3. Nome do servidor RightFax utilizado pela API do RightFax. Campo chamado “RFAXSERVER” com 12 dígitos alfanuméricos;

2.1.4. Senha de acesso ao servidor RightFax, utilizada pelo usuário RightFax acima. Campo chamado “RFAXPASS”, com nenhum até 15 dígitos alfanuméricos;

2.1.5. Ativação de LOG para depuração do sistema. Campo com 01 dígito numérico com valor 0 (zero) para não ativado e 1 para ativado;

2.1.6. Primeira posição de canal telefônico de atendimento da URA. Campo com 1 até 3 dígitos numéricos;

2.1.7. Última posição de canal telefônico de atendimento da URA. Campo com 1 até 3 dígitos numéricos;

2.1.8. Diretório de trabalho compartilhado na rede local, onde os arquivos XML devem ser gravados pelo Sistema URA, campo chamado “XMLANALISAR”. Exemplo: “\\MZ-CW-AP-067\BSURA\Xml_Analisar”;

2.1.9. Diretório compartilhado na rede local, onde os arquivos XML órfãos ou com erros, são movidos. Existem as seguintes possibilidades para arquivos XML serem movidos para esse diretório: Um fax não foi recebido, mas o arquivo XML foi criado ficando órfão, e um arquivo XML foi criado com erro em seu conteúdo. Esse campo será chamado “XMLDESCARTADOS”. Exemplo: “\\MZ-CW-AP-067\BSURA\Xml_Descartados”;

2.1.10. Diretório compartilhado na rede local, onde os arquivos XML já analisados, são movidos, uma vez que eles foram processados corretamente pelo Sistema FAX, e as informações contidas no arquivo XML foram associadas a um fax recebido. Esse campo será chamado “XMLPROCESSADOS”. Exemplo: “\\MZ-CW-AP-067\BSURA\Xml_Processados”;

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

2.1.11. Valor de horas em que um arquivo XML pode permanecer no diretório de trabalho “Xml_Analisar” do Sistema URA. Campo com 1 até 2 dígitos numéricos.

2.2. Linha 2 do arquivo de parâmetro:

2.3. Ramais DDR atendidos pelo WebService da Bradesco Seguros (separados por ponto-e-vírgula). Exemplo: 6479;1681;5874;1013;1723. Isso facilita a entrada e saída de números DDR para o serviço de URA da Bradesco Seguros;
IMPORTANTE: Esses ramais acima são os de produção na Bradesco Seguros. Mas ao serem migrados para o Banco Bradesco eles serão modificados, pois no Banco Bradesco vamos trabalhar com 06 (seis) dígitos, em vez de 04 (quatro) dígitos.

2.4. Linha 3 do arquivo de parâmetro:

2.4.1. Endereço do WebService para a consulta do código do referenciado na Bradesco Seguros;
IMPORTANTE: Esse endereço de URL para a consulta de WebService não deve ser fixado no programa do Sistema URA, pois eles já sofreram modificações no passado. Por tanto, deve ser um parâmetro a ser lido pelo Sistema URA.

2.4.2. Endereço do WebService para a consulta do código do segurado (beneficiado) na Bradesco Seguros;
IMPORTANTE: Esse endereço de URL para a consulta de WebService não deve ser fixado no programa do Sistema URA, pois eles já sofreram modificações no passado. Por tanto, deve ser um parâmetro a ser lido pelo Sistema URA.

2.5. Linha 4 do arquivo de parâmetro:

2.5.1. Ramais DDR atendidos pela WebService da Mediservice (separados por ponto-e-vírgula). Exemplo: 6480;6100. Isso facilita a entrada e saída de números DDR para o serviço da Mediservice;
IMPORTANTE: Esses ramais acima são os de produção na Bradesco Seguros. Mas ao serem migrados para o Banco Bradesco eles serão modificados, pois no Banco Bradesco vamos trabalhar com 06 (seis) dígitos, em vez de 04 (quatro) dígitos.

2.6. Linha 5 do arquivo de parâmetro:

2.6.1. Endereço do WebService para a consulta de código de senha, CPF e CNPJ da Mediservice.
IMPORTANTE: Esse endereço de URL para a consulta de WebService não deve ser fixado no programa do Sistema URA, pois eles já sofreram modificações no passado. Por tanto, deve ser um parâmetro a ser lido pelo Sistema URA.

3. O exemplo abaixo é de um arquivo texto puro, com o nome “Config_Bradesco.ini”, e é um exemplo do segundo arquivo de parâmetros, onde o conteúdo dos dados é bem próximo das informações reais a serem utilizadas no ambiente de produção. Cada linha contém um dos parâmetros acima listados, na mesma ordem em que foram descritos, os parâmetros que estão em destaque, são os que precisão ser utilizados, de acordo com essa documentação:

LEGENDA: O caractere # se refere a uma linha de comentário, e deve ser ignorada.

SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

203632;URASAUDE;BS_SRV_FAX;urasaude;1;16;30;\\MZ-CW-AP-067\BSURA\Xml_Analisar;\\MZ-CW-AP-067\BSURA\Xml_Descartados;\\MZ-CW-AP-067\BSURA\Xml_Processados;4
201723;205874;206468;201013
http://wwws.bradescosaude.com.br/PCBS-WebServicesSaude/service/ReferenciadoService?wsdl;http://wwws.bradescosaude.com.br/PCBS-WebServicesSaude/service/SeguradoService?wsdl
206100;206480
https://www.mediservice.com.br/webservices/wscadastro/Service.asmx?WSDL
#********Linha 1: parâmetros gerais (separados por ponto-e-vírgula.) **********
#Linha 1:
#numeroRF;
#UserRF;
#ServerRF;
#UserPassRF;
#Ativação de Log (0 = desativado ou 1 = ativado);
#Primeiro Canal do sistema;
#Último Canal do sistema;
#Caminho de rede da pasta 'xml a analisar';
#Caminho de rede da pasta 'xml analisado'
#Caminho de rede da pasta 'xml descartados'
#Permanência em Horas dos arquivos XML nas pastas
#**********Linha 2 e 3: parâmetros de configuração do 1º e 2º Web Service de consulta do Sistema URA **********
#linha 2:
#Ramais da telefonia DDR atendidos pela URA (separados por ponto-e-vírgula)
#Exemplo: 201723;205874;206468;206479
#linha 3:
#Endereço do Web Service para a 1º consulta do Sistema URA;
#Endereço do Web Service para a 2ª consulta do Sistema URA;
#**********Linha 4 e 5: parâmetros de configuração do 3º Web Service de consulta do Sistema URA **********
#Linha 4:
#Ramais da telefonia DDR atendidos pela URA (separados por ponto-e-vírgula)
#Exemplo: 206100;206480
#Linha 5
#Endereço do Web Service para a 3ª consulta do Sistema URA **********

Anexo 2

Trecho de código que faz inserção em faxes do rightfax dos dados que estão no XML

```
REM pegar a coleção de faxes do usuário na caiaxa-postal de fax.
REM interagir fax a fax para verificar se o 'fromfaxnumber' do fax coincide com o IDXML do nome do arquivo XML.
REM se coincidir os metadados são gravados no fax recebido RENOMEANDO o fax, que em seguida é movido para o usuário final.

Function InserirNoRightFax (codigo,matricula,matricula2,routingCode,FromName,Comments)
	dim retorno
	retorno = 0
	dim faxes
	dim users
	dim usersRouting
	set users = gbl_FaxServer.Users                          'pegando coleção de usuários no servidor RF

	for each userRF in users                                 'interagindo usuário á usuário para resgatar o usuário ao qual o fax será roteado
		if (Cstr(userRF.DIDNumber) = Cstr(routingCode)) then 'verificando SE o DID do usuário é o mesmo do routing code pego do XML
			set usersRouting = userRF
			exit for
		end if
	next
```


SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988

```
        set faxes = gbl_FaxServer.User(Cstr(gblUserName)).Faxes                  'faxes do usuário da caixa centralizadora

        for each fax in faxes                                                   'interagindo fax a fax da coleção resgatada no servidor RF par
                if ((Cstr(fax.FromFaxNumber)) = (Cstr(codigo))) and (fax.IsReceivedEx = 1) then   'verificando se no item fax o 'From Fax Nnumber' =  código extraído do XML atual

                        If Cstr(gblRoutingCode) = Cstr("0468")  Then             'DDR ESPECIAL REGRA BRADESCO
                                gblRoutingCode = Cstr("6468")
                        end if
                        fax.BillingCode1 = matricula                             'campo billingCode1 do Fax no RF ficará com o campo matrícula no XML
                        fax.BillingCode2 = matricula2                            'campo billingCode2 do Fax no RF ficará com o campo matricula2 no XML
                        fax.FromName = Comments                                  'campo FromName do Fax no RF ficará com o campo Comments no XML
                        fax.UserComments = FromName                              'campo UserComments do Fax no RF ficará com o campo FromName no XML
                        fax.Save(true)                                           'Salva o fax no servidor RF

                        dim faxnovo                                              'Cria novo Fax
                        set faxnovo = fax.ForwardToNewFaxNumber                  'preparação para rotemaneto de Novo fax
                        faxnovo.ToName = routingCode                             'campo ToName do Novo Fax no RF ficará com o campo routingCode no XML
                        faxnovo.ToFaxNumber = "990000"                           'caixa-postal 990000 para encaminhar novo  fax
                        faxnovo.send                                             'encaminha Novo fax ao usuário 1001

                        fax.RouteTouser usersRouting,""                          'encaminha fax renomeado ao usuário final
                        retorno = 1
                        exit for
                end if
        next

        InserirNoRightFax = retorno

End Function
```


SCS · Quadra 01 · Bloco K · Nº 30 · Edifício Denasa
Avenida Paulista · Nº 777 · São Paulo · SP
(61) 2196-8000 · (11) 3323-1988