

UTILIZAÇÃO DA LINGUAGEM DE SCRIPTING PARA MELHORAR A AUTOMAÇÃO DA PRESTAÇÃO DE SERVIÇOS

Andressa M. de Sá Fernandes⁽¹⁾, Geraldo C. Guimarães⁽¹⁾, João Cícero da Silva⁽²⁾
Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica⁽¹⁾ e Faculdade de Engenharia Mecânica⁽²⁾
andressa-0306@hotmail.com, gcaixeta@ufu.br, jciceros@mecanica.ufu.br

Resumo – No mercado competitivo a busca por qualidade técnica e econômica é primordial. Neste contexto, é necessário visar à eficiência da produção dos bens de consumo e serviços em geral. Quanto aos serviços, a automação também atua como peça chave, pois otimiza o tempo gasto em determinado procedimento devido a agilidade e precisão nos processos realizados. Dessa forma, este trabalho prioriza o estudo de métodos para obter melhorias neste segmento. Para ir além do *software* programado por empresas próprias da área, verificou-se a importância de estudar uma linguagem de *scripting*. Esta viabiliza a elaboração de algoritmos e códigos genéricos que permitem muitas aplicações e é de fácil entendimento, o que permitirá a própria prestadora de serviços elaborar o programa ideal de acordo com sua necessidade, melhorando a produtividade sem estar vinculada a outra empresa que licencie o *software*.

Palavras-Chave – Automação, linguagem de *scripting*, qualidade, produtividade, prestação de serviços.

USING OF SCRIPTING LANGUAGE TO IMPROVE THE SERVICE DELIVERY AUTOMATION

Abstract – In the competitive market the search for technical and economical quality is vital. In this context, the aim is efficiency in production of consumer goods and services in general. For services, the automation also acts as a key tool, because it optimizes the time spent on particular procedure due to speed and accuracy in processes carried out. Thus, this work emphasizes the study of methods of automation in this market segment. To go beyond the software programmed by companies of this area, it is very important to study a scripting language which enables the development of generic algorithms and codes that allow many applications. It is also easy to understand which can allow the service provider itself to develop the ideal program according to its needs, improving productivity without being linked to another company responsible for the software license.

Keywords - Automation, language scripting, quality, productivity, service delivery.



X CEEL - ISSN 2178-8308
24 a 28 de setembro de 2012
Universidade Federal de Uberlândia - UFU
Uberlândia - Minas Gerais - Brasil

I. INTRODUÇÃO

O setor produtivo brasileiro é altamente oligopolizado, afirmam os economistas Marco A. Vasconcellos e Manuel E. Garcia [1] e ainda, de acordo com a Teoria Microeconômica, há uma tentativa de maximização dos lucros por parte das empresas. Diante desse fato, a estratégia de produção em massa não é mais válida. Para permanecer neste mercado a empresa deve tornar a produção viável e efetiva em termos técnicos e financeiros, ou seja, é necessário flexibilizar e agilizar a produção, além de garantir qualidade no produto e reduzir custos.

Quando se fala em processo produtivo o leitor deve estar ciente de que um produto pode ser resultado da transformação de insumos ou resultado do processamento de serviços prestados, visto que os dois são produtos dos quais o cliente necessita e utiliza. Tendo em vista que a concorrência neste mercado pode impedir um aumento nos preços, para que a empresa consiga cobrir os custos ela deve reduzir gastos e aumentar a produtividade, incluindo os serviços, pois, como foi dito, este também está inserido no processo. Nesse sentido, este trabalho explora os meios de automação nesta etapa da produção, ou seja, na prestação de serviços.

Nos tópicos abaixo, serão abordados os conceitos de automação enfatizando o setor de serviços. Em seguida, a contribuição do estudo estará voltada para a análise na aplicabilidade de uma linguagem de extensão visando à redução da intervenção humana. Por fim, propõe-se uma solução que utiliza uma linguagem de extensão *freeware* nos moldes da linguagem de programação BASIC que permite o controle de GUI's do Windows e *scripting* em geral.

Dada a facilidade e generalidade desta linguagem, foi possível obter bons resultados ao implementar programas específicos de acordo com a necessidade de serviço prestado.

II. O CONCEITO DE AUTOMAÇÃO EM SERVIÇOS

Inicialmente cabe definir o que vem a ser automatização e automação, pois em bibliografias comumente utilizadas pode-se constatar a utilização destes termos como sinônimos. Em seguida, o enfoque será dado ao setor de serviços.

A. Automatização

Automatizar nada mais é do que planejar a automação de determinado sistema ou serviço. Implica na idéia de coletar, armazenar, processar e transmitir informações materializadas nos equipamentos de produção de bens e serviços. Como conceituou Francisco S. Borba [2], pode-se dizer que a automatização trabalha baseada na automação dos equipamentos. No caso deste estudo, a automatização está voltada a análise de requisitos do programa a fim de gerar o

algoritmo ideal para controlar a complexidade e os custos da informática dentro de uma empresa.

B. Automação

A automação operacionaliza as atividades necessárias para a melhoria do processo em estudo. Dessa forma, esta pode ser definida como sendo o controle de processos automáticos por meio do acréscimo de algum tipo de inteligência a fim de executar tarefas de modo mais eficiente, com vantagens econômicas e técnicas como: redução do gasto com matéria prima, agilidade, diferenciação e qualidade do produto, garantia de repetibilidade, flexibilidade na produção e segurança nos processos.

C. Automação em Serviços

Na análise voltada aos serviços, a estratégia é integrar os objetivos do negócio com a tecnologia da informação para uniformizar procedimentos no âmbito corporativo. Dentro da empresa, a transmissão e o processamento da informação de forma rápida e segura são essenciais para tomada de decisões e agilidade nos demais processos que envolvem um mesmo produto. Para alterar estas decisões organizacionais é necessário aplicar a automatização, registrando o que é relevante para o negócio e determinando quais os processos computacionais serão implementados ou alterados para garantir uma solução efetiva.

De acordo com James A. Fitzsimmons e Mona J. Fitzsimmons [3] serviços de infraestrutura em comunicações formam um elo essencial entre os setores da economia, incluindo o consumidor final. Deve-se ter em mente que automação de serviços contempla procedimentos realizados por pessoas e programações efetuadas em equipamentos que garantem regularidade de execução.

Após verificar que a automação de serviços pode manipular dados de transações ou operações realizadas por clientes e funcionários, quando mais rápido e seguro for esta transmissão, melhor o fluxo do produto dentro da empresa, facilitando tomadas de decisões e incorporando métodos de registros e avaliações de desempenhos.

Em se tratando de qualidade do serviço, Lourdes Hargreaves, Rose Zuanetti e Renato Lee [4] descrevem que em um ambiente competitivo, surge o cliente mais exigente em relação ao produto e ao serviço que consome. Como consequência, o padrão de qualidade faz a diferença no mercado e pode ser definido como algo bem projetado, elaborado, organizado, administrado e bem apresentado que atenda as especificações e satisfaça necessidades de produtores, prestadores de serviço e clientes.

No caso dos prestadores de serviço, John R. Schermerhorn Junior, James G. Hunt e Richard N. Osborn [5] mencionam que geralmente serviços muitos simplificados ou repetitivos não fornecem motivação ao trabalhador, gerando problemas de desempenho dentro da empresa.

III. JUSTIFICATIVA DO ESTUDO

Investir na automação de serviços é uma ótima solução para a empresa manter sua representatividade no mercado e alcançar um bom padrão de qualidade. Este estudo se justifica pelo empenho de uma concessionária de energia elétrica em aprimorar os serviços prestados. Dentro da

empresa todas as etapas, desde o projeto até a finalização e capitalização das obras, devem ser cadastradas e registradas no sistema. Diante desse fato, foi necessário elaborar uma forma rápida e segura de inserir estes dados, garantindo eficiência no processamento de informações. Assim, o maior fluxo de informações auxilia na interação entre os diferentes setores que trabalham em uma mesma obra, podendo, inclusive, influenciar no prazo estipulado com o cliente.

O motivo maior para o desenvolvimento do *software* se deve as alterações em códigos ou preços nos materiais que já haviam sido registrados e, por algum motivo, houve a necessidade de reclassificar estes dados. Nestes casos, o volume de serviço é maior e, por se tratar de um processo repetitivo que lida com várias tabelas e valores numéricos, existe a probabilidade de ocorrer erros graxos. Dessa forma a automação tem como vantagem a garantia de atualização rápida e segura das informações, ou seja, sem conter erros.

Portanto, em termos de efetividade técnica, reduzir a intervenção humana nesta tarefa possibilitará a economia de tempo e a segurança de que não haverá erros nos cadastros efetuados, o que de fato é vantajoso para a empresa. Após a implementação do programa, caberá ao funcionário somente a inserção dos dados em planilhas para habilitar a manipulação de valores conforme a ordem de colocação. Logo, com o uso do programa, o trabalhador influencia em uma pequena parcela do processo, algo relativamente menos cansativo do que efetuar o cadastro completo ou corrigir dados repetidas vezes.

Além disso, para os casos de novas contratações, por exemplo, um simples programa como este pouparia horas dedicadas a treinamentos de funcionários, pois, se o serviço estiver automatizado, não se faz necessário que o trabalhador saiba manipular o programa interno de cadastros.

IV. APLICAÇÃO DA LINGUAGEM DE SCRIPTING

Uma linguagem de *script* é um arquivo de texto simples que contém palavras chaves para orientar a compilação. Como definido por Robert W. Sebesta [6] tais linguagens possuem uma lista de comandos como base e a esta são adicionadas variáveis, instruções de fluxo, funções e outras capacidades que tornam a programação completa.

A importância deste tipo de linguagem é o fato de não ficar restrita ao ambiente na qual está sendo aplicada, pelo contrário, esta é capaz de estender a funcionalidade e controle de um determinado programa sendo, inclusive, utilizada em sistemas operacionais. Logo, existe uma infinidade de aplicações para este tipo de programação em *script* e muitas das linguagens são comumente conhecidas, como: BASIC, Linguagem C, JavaScript, Python e VBScript.

A linguagem utilizada neste trabalho é o AutoIt v3 que, assim como as outras, também é de extensão. Os motivos para a escolha desta linguagem estão relacionados com as seguintes características dadas pelos contribuintes desta codificação [7]:

A. Facilidade de programar

Código derivado do BASIC é fácil de aprender, sendo considerado “didático” segundo A. Min Tjoa [8]. Além disso, as mensagens de erro são amigáveis, tornando fácil a identificação e correção dos problemas.

B. Possui boa extensão

A linguagem sai do escopo de linhas de código e consegue manipular os controles do Windows. Tendo boa interação com o sistema operacional, é capaz de simular teclas, movimentos do mouse, manipular janelas e processos. Estas rotinas de manipulação são configuráveis em termos de velocidade e funcionalidade.

C. É possível compilar o código em executáveis "standalone"

A elaboração e edição podem ser feitas no bloco de notas ou em outros arquivos próprios para *scripts*, proporcionando flexibilidade para a manipulação do código.

D. É autônomo

A execução não depende de *RunTimes*, pois utiliza o executável principal do AutoIt (*AutoIt3.exe*). Este arquivo pode ser instalado com facilidade ou então poderá ser executado de forma portátil.

E. Help do AutoIt

O programa possui um extenso e detalhado arquivo de ajuda que serve de suporte técnico para qualquer iniciante na programação. Durante a programação existe um executável de informações (*Au3Info*) na qual constam todas as atribuições para referenciar determinado objeto.

F. É isento de licenças

O AutoIt é um *software* gratuito, logo, basta fazer o *download* e começar a programar. Não precisa de licenças nem cadastros, facilitando o acesso.

V. ANÁLISE DO MÉTODO PROPOSTO

Dada a motivação para este trabalho, é possível determinar quais serão as funções utilizadas de acordo com a lógica de funcionamento do programa. O programa deverá manipular os controles de mouse e teclados para clicar e escrever o que for necessário no programa de cadastros. Também deverá estar habilitado para gerenciar as janelas abertas no momento de execução e ainda deverá ser capaz de criar ou abrir arquivos.

Após o estudo da automatização, transferir do algoritmo para o código em si é uma questão de conhecimento a respeito da linguagem e do que esta é capaz de fazer. Portanto, como proposto por Jonathan Benennett [7] é necessário dominar a sintaxe das principais bases do AutoIt que são: referências de linguagem e funções.

A. Referências de Linguagem

1) Variáveis e operadores

O espaço de memória para armazenar um dado qualquer foi utilizado para todas as informações inseridas no programa empresarial. Dado que o código foi elaborado por etapas todas as variáveis foram declaradas como globais, o que permite a utilização do valor em mais de uma função. Ao utilizar este artifício foi possível copiar todos os valores da planilha de uma só vez alocados em espaços diferentes da memória.

$Global \$código = "valor"$ (1)

Já os operadores foram utilizados em condicionais e loops por serem essenciais na comparação de valores e tomadas de decisões do programa.

2) Declarações de condicionais e loop

No caso de mudança do tipo de cadastro, mudança de janelas durante o processo ou alterações na interface do programa manipulado, para que o programa tenha poder de decisão, é necessário fazer uso de condicionais. Os termos utilizados nesta declaração são autoexplicativos: *If*, *Then* e *Else*.

Dado que o estudo visa à automatização de serviços, a declaração de loop é justamente o ponto que gera a otimização do processo. Para tal, utiliza-se o *While/WEnd* (*Enquanto*). Logo, enquanto houver dados para armazenar o programa fica rodando e inserindo todas as informações que constam na planilha.

B. Funções

Utilizando as bibliotecas do programa, o AutoIt possui várias funções que podem ser exploradas de acordo com a aplicação de cada código criado. Entretanto será abordado apenas as funções utilizadas no programa proposto.

1) Controle do mouse

A maneira mais confiável de se controlar o mouse é por meio de uma função *ControlClick()*. Diferentemente de comandos que apenas movem o cursor por coordenadas *x* e *y* esta função possibilita identificar o controle na qual o programa interage. Pela sintaxe exibida em (2) é possível notar as diversas formas de controle que permitem a obtenção de resultados específicos e dando boa precisão na automação.

$ControlClick("título", "texto", controlID, [botão[, cliques[, x[, y]]]])$ (2)

A *ControlClick()* foi utilizada com a finalidade de colocar o cursor nas caixas de texto, deixando o programa pronto para inserir os valores e ainda, para clicar em botões de avanço durante o processo.

2) Controle do teclado

Para simular o pressionar das teclas utilizou-se a *Send()* que é a mais usada neste tipo de controle. Esta função pode inserir dados, variáveis ou strings do código, e executar qualquer comando especial como: *CTRL+C*, *TAB*, *ENTER* etc. Para uma maior confiabilidade. Para tal, possui uma flag utilizada na distinção de valores brutos (4) com especiais (5). Observe as diferentes formas:

$Send(\$variavel)$ (3)

$Send("valor-bruto", 1)$ (4)

$Send("{comando-especial}", 0)$ (5)

O processamento do programa comandado tem delay na resposta, consequentemente a função teve de ser configurada para operar em uma velocidade menor. Para um funcionamento eficaz foi utilizado um delay de 10 segundos no pressionar entre as teclas (6) e no tempo que cada uma permanece pressionada (7).

`AutoItSetOption("SendKeyDelay",10)` (6)
`AutoItSetOption("SendKeyDownDelay", 10)` (7)

3) Gerenciamento de Ambiente

Para auxiliar a `Send()` existem mais duas funções: `ClipGet()` e `ClipPut()`. Respetivamente são capazes de captar e inserir um valor na área de transferência. Dessa forma é possível alocar várias variáveis com diferentes dados para posteriormente inserir no local desejado, conforme a aplicação do código. Estas funções são utilizadas principalmente nos casos que se deseja somente inserir um valor, pois a `Send()`, ao enviar um dado, também pressiona simultaneamente a tecla ENTER, porém nem sempre é interessante pressionar esta tecla. Usando `ClipPut()` é possível apenas colar o valor da área de transferência com a auxílio da `Send()` ao invés de enviar o valor, dessa forma a função não simula o pressionar de ENTER.

`$montante = ClipGet()` (8)
`ClipPut($montante)` (9)

Primeiramente, como mostrado acima, a variável é declarada e definida com o valor de transferência do Windows (9). Em seguida exemplifica como inserir um valor desejado nesta área (10).

4) Gerenciamento de janelas

Existem 4 funções principais e 2 complementares para este tipo de comando. Ao lidar com a janela de um programa que fora recentemente iniciado é ideal aguardar um tempo até a este esteja aberto e em condições de operar com um bom funcionamento.

`WinWait("título da janela", "texto")` (10)
`WinWaitActivate("título da janela", "texto")` (11)
`WinActive("título da janela", "texto")` (12)
`WinActivate("título da janela", "texto")` (13)
`WinSetState("título da janela", "texto", @SW_MAXIMIZE)` (14)

As funções mostradas em (10) e (11) respetivamente aguardam a existência e a ativação da janela requerida pausando o *scripting*.

Já em (12) e (13) ocorre à verificação da janela solicitada, se está ativa ou não e, caso não esteja, a habilita. Como identificação da janela utiliza-se o título e o texto visível, sendo este opcional.

No caso do programa apenas os títulos já serviram de referência pois em cada etapa do cadastro o sistema altera o nome. Porém, para melhorar o reconhecimento das janelas estas funções também foram configuradas de forma que apenas um trecho do título já consiga ser identificado. Assim, o nome não precisa ser exatamente fiel a sua chamada, deixando o programa mais flexível. A função que realiza esta configuração é a `AutoItSetOption('WinTitleMatchMode', 2)` onde a flag 2 significa que o reconhecimento se dá por uma *substring*.

Ativar uma janela não significa maximizar, logo, caso a janela esteja minimizada, utiliza-se a função de auxílio com a flag `@SW_MAXIMIZE` (14).

5) Gerenciamento de Processos

Entende-se por gerenciamento de processos a capacidade de se controlar programas externos. Para abrir uma planilha do Excel que já existe utilizou-se uma função que permite iniciar qualquer aplicativo instalado no computador do usuário (15).

Caso o sistema reconheça determinado programa como um objeto é possível utilizar funções como: `ObjCreate()` ou `ObjGet()`.

O programa possui as duas opções, o arquivo no Excel pode ser aberto, permitindo a inserção de dados na planilha ou, caso já tenha uma planilha preenchida, o programa apenas abre e executa este arquivo.

`ShellExecute("Nome", "parâmetros", "diretório", "ação", [flag])` (15)

Como mostrado acima à chamada inclui o nome, os parâmetros e o diretório do arquivo. Em seguida declara a ação que pode ser: abrir, editar, imprimir ou visualizar propriedades. Já a *flag* é utilizado para maximizar, minimizar ou ocultar a janela. Os comandos entre chaves são opcionais, porém, no código todos os parâmetros foram declarados para possibilitar a abertura do arquivo correto.

6) Interação com o usuário

Por fim, como na maioria dos casos de automação do serviço, é necessário interagir com o profissional que estará usufruindo o programa. É fundamental que se tenha caixas de texto para permitir a inserção de dados ou que sirvam de alerta durante algum procedimento.

`InputBox("título", "texto", "padrão", "password", [dim, timeout, hwnd])` (16)
`MsgBox(flag, "título", "texto", [timeout, hwnd])` (17)

A função (16) é declarada com o título da janela exibida, o texto presente na janela, o padrão que já virá preenchido, a confirmação do que deve ser digitado, a dimensão da janela, o tempo que esta ficará visível e o handle usado como principal do diálogo. No caso da `MgsBox()`, a diferença consiste na flag inicial que indica os tipos de caixa com as possíveis combinações de botões.

O *software* elaborado utiliza (16) para saber os tipos de cadastro, confirmar os textos que em alguns casos se alteram e perguntar a quantidade de linhas preenchidas na tabela do Excel. Já a (17) foi utilizada para dar alertas de correção caso o programa identifique alguma uniformidade de dados.

C. Executando o programa

Após a explicação da linguagem e das principais funções utilizadas resta esclarecer como o arquivo foi compilado e qual o procedimento para gerar um executável.

No total, para chegar à versão final do programa, foram executados cinquenta e três testes. Os editores utilizados foram o bloco de notas e o Notepad++. No caso do primeiro, os arquivos foram salvos no formato .au3. Já o segundo contém a linguagem AutoIt no menu de opções.

Como o AutoIt é uma aplicação pequena e autônoma, pode-se executar *scripts* com o executável principal (`AutoIt3.exe`). O código também pode ser convertido em arquivo .exe com o compilador `Aut2Exe`. Assim o usuário não precisa ter o AutoIt instalado.

VI. RESULTADOS OBTIDOS

Nesta seção o método proposto será avaliado quando a sua eficiência no ambiente de trabalho ao qual está inserido. A automatização do serviço surgiu da necessidade da empresa efetuar, com agilidade e segurança de informações, um processo repetitivo de cadastro dos dados de serviços prestados.

Assim em termos práticos, considerando que o usuário já tenha entrado com o *login*, o programa criado foi capaz de executar todos os passos necessários para automação do processo. Para efeitos de ilustração observe as figuras.

A. Análise dos passos durante o processo

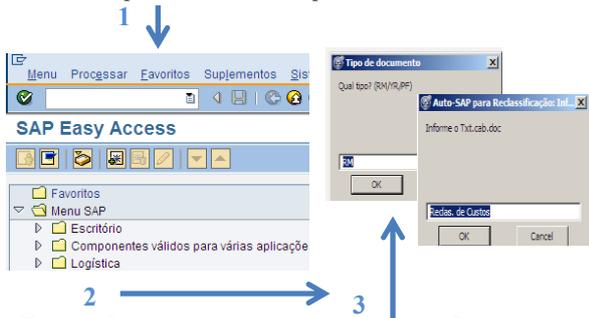


Fig. 01. Representação do funcionamento – Passos 01 a 03.

Onde:

1) As funções de gerenciamento de janelas aguardam até que a primeira janela do SAP Easy Access exista para depois maximizar e ativar;

2) O controle de teclado insere a chave de registro que identifica o processo de reclassificação de serviços e pressiona a tecla ENTER, alterando a tela de interface;

3) Em seguida, as interações com o usuário perguntam o tipo de cadastro e o texto que será inserido na reclassificação. O cadastro é feito para mais de um tipo de serviço e para cada tipo o programa a ser manipulado tem uma interface diferente. Este fato justifica a necessidade de tomadas de decisão do código, pois para cada registro os controles se alteram.

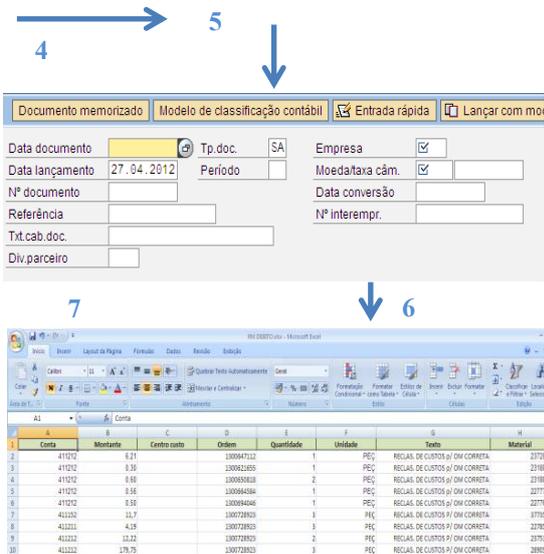


Fig. 02. Representação do funcionamento – Passos 04 a 07.

Onde:

4) Para dar continuidade o programa novamente aguarda até que a próxima tela exista, pausando o código, após a tela será maximizada e a ativada para manipulação.

5) Nesta tela os dados serão cadastrados, sendo que os únicos valores que se alteram já foram perguntados ao funcionário no passo 3. Dessa forma, os demais, ou seja, valores que não dependem do usuário, o modo automatizado preencherá as colunas necessárias (exemplo: datas, código da empresa, período e taxa de câmbio).

6) Caso não exista uma planilha, é criado um objeto com formato .xlsx (arquivo do Excel). Na aplicação atual, o documento está sendo gerado e referenciado com o título Pasta1, dado que foi uma nova planilha aberta pelo programa.

7) O executável gera um arquivo no Excel devido a facilidade de manipulação para o funcionário. Dessa forma, o código fica aguardando o preenchimento de todas as informações a serem cadastradas.

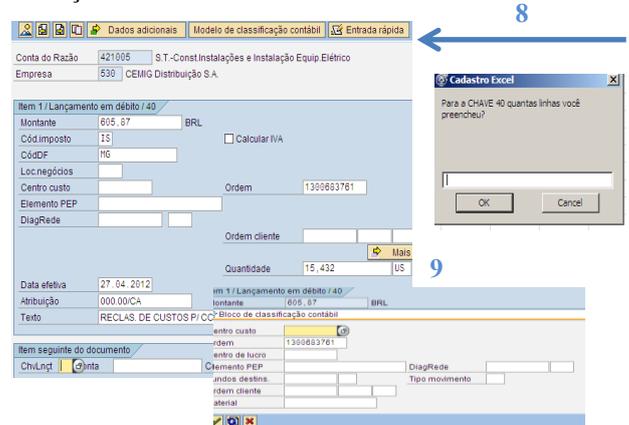


Fig. 03. Representação do funcionamento – Passos 08 e 09.

Onde:

8) A pausa do código é dada pela *InputBox* na qual o usuário deverá inserir a quantidade de linhas preenchidas na planilha. Em seguida todos os dados da primeira linha são copiados, armazenados em variáveis e a janela do Excel é minimizada;

9) Mais uma vez, por questões de confiabilidade na automação oferecida, o código aguarda até que a nova tela do SAP exista e esteja ativada. Assim, valores como montante, ordem, centro de custo, texto de preenchimento, quantidade e código do material, após copiados, com o auxílio das funções de gerenciamento de ambiente, são inseridos. Os valores da planilha que estão armazenados entram na área de transferência e são colados no programa de cadastro por meio da *Send()*, que controle o pressionar das teclas com o comando especial "CTRL+V". Para identificar botões de avanço a função *ControlClick()* foi utilizada, possibilitando precisão de local devido aos vários atributos utilizados. Também, para evitar erros, ao pular para as janelas de texto que devem ser preenchidas utilizou-se apenas controle de teclado e não de mouse.

Este passo, juntamente com a cópia dos dados foi realizado ciclicamente de acordo com a quantidade de informações contidas na planilha. Fato que justifica a

utilização de loops. Além disso, os mesmos dados devem ser cadastrados duas vezes para duas chaves de registro diferente. Logo, quando acabar o primeiro loop, o programa deve alterar a chave de registro e tornar a entrar no loop.

10

CL	Div.	Conta	BRL	Montante	Montante de
001	40	0000421005 Cons.Inst. Inst. Eq. E		605,87	IS
002	40	0000421005 Cons.Inst. Inst. Eq. E		605,50	IS

Fig. 03. Representação do funcionamento – Passo 10.

10) Por fim, após o processo repetitivo com os todos os registros e para as duas chaves. O programa deverá exibir a síntese e o funcionário deverá apenas verificar se os valores somados estão de acordo com o esperado. Se sim, é sinal de que o modo automatizado executou o código de acordo com o esperado. Assim, o usuário poderá salvar as alterações do SAP e encerrar o programa.

B. Análise quantitativa e qualitativa

Em termos de eficiência, pode-se dizer que o programa criado de acordo com esta necessidade em específico é capaz de responder perfeitamente ao que foi solicitado, ou seja, o código executa os cadastros de todos os serviços de forma segura, confiável e rápida.

Numericamente, normalmente, em um dia de serviço, são cadastrados no mínimo 45 serviços, contabilizando todos os tipos de cadastro. Pode-se dizer que um funcionário ficava aproximadamente uma hora fazendo um serviço que atualmente pode ser feito em 20 minutos ou menos. Dessa forma, poupa-se o tempo de ocupação do contratado deixando-o livre para exercer outra atividade.

Ao se pensar em um mês de serviço, 40 minutos ao dia resultam em aproximadamente quatorze horas de trabalho. Além disso, mais de um trabalhador faz esta mesma função, logo a automação afeta uma escala ainda maior.

Este fato confirma o que foi dito anteriormente, a automatização de processos trouxe benefícios à empresa e também aos próprios funcionários que não se desgastam mais ao exercem a mesma atividade diversas vezes ao dia. Como afirma L. W. Turner [9], esta é a vantagem mais óbvia desta melhoria para a prestação de serviços.

VII. CONCLUSÕES

Neste trabalho foi apresentado um método eficiente para a automação em serviços o qual envolve a codificação de *software* em linguagem de *scripting*, especificamente a AutoIt, dado a facilidade de se programar.

Esta forma de automatização surgiu com o processo de análise de requisitos no qual consta o desejo da empresa em aumentar a produtividade dos funcionários, garantir a qualidade do produto final, segurança e agilidade nas

informações transmitidas no decorrer do processo.

Os detalhes principais a respeito da implementação do método foram descritos, incluindo conceitos de automação e qualidade de serviços. Além disso, visando ressaltar a importância desse tipo de inovação fez-se uma análise da situação atual de mercado e avaliou-se a forma organizacional da empresa.

Em seguida, foram apresentadas as principais funções, incluindo a sintaxe, para posteriormente demonstrar passo a passo os resultados obtidos com este tipo de programação.

A partir do estudo da linguagem e da análise de requisitos, feita a codificação, pode-se observar que a aplicação do *software* trouxe efetivos benefícios durante o processamento de informações. Houve melhoria na qualidade do serviço para com todos os relacionados. Para o funcionário a atividade deixou de ser repetitiva e cansativa. Para a empresa o programa reduziu o tempo gasto com este serviço e permite que a mão de obra possa trabalhar em outras atividades além da garantia de segurança no processo. Já para os clientes, ao expandir o pensamento para além do cadastro, pensando na rapidez da troca de informações, o cliente também tem vantagens com a automação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] VASCONCELLOS, Marco Antonio S.; GARCIA, Manuel E.. Fundamentos de Economia. São Paulo: Editora Saraiva, 1999. 57-82 p.
- [2] BORBA, Francisco S. (Org.). Dicionário UNESP: Do Português Contemporâneo. São Paulo: Unesp Editora, p. 141, 2004.
- [3] A FITZSIMMONS, James; FITZSIMMONS, Mona J. Administração de serviços operações, estratégia e tecnologia da informação. 4. ed. Porto Alegre: Porto Alegre Bookman, pp. 25-85, 2005. CD-ROM.
- [4] HARGREAVES, Lourdes; ZUANETTI, Rose; LEE, Renato (Ed.). Qualidade em Prestação de Serviços: Edição atualizada e ampliada. 2. ed. Rio de Janeiro: Senac Nacional, pp. 07-34, 2005.
- [5] SCHERMERHORN JUNIOR, John R.; HUNT, James G.; OSBORN, Richard N.. Fundamentos do Comportamento Organizacional. 2. ed. Porto Alegre: Artmed Editora S.a., pp 115-130, 1998.
- [6] SEBESTA, Robert W. . Conceitos de Linguagens de Programação. 5. ed. Porto Alegre: Porto Alegre Bookman, pp. 12-47, 2002.
- [7] BENNETT, Jonathan; LTDA, Autoit Consulting. AutoIt v3.3.8.1. Acessado em 24/abril/2012 e disponível em: <http://www.autoitscript.com/autoit3/docs/>
- [8] TJOA, A. Min et al. (Ed.). Availability, Reliability and Security for Business, Enterprise and Health Information Systems. Vienna: Springer, pp. 361-372, 2011.
- [9] TURNER, L. W.. Electronics Engineer's Reference Books. 4. ed. London: Butterwork & Co. (publishers) Ltd., pp. 241-247, 2004.