



# **Como Melhorar a Qualidade de Softwares de Micro e Pequenas Empresas**

Guilherme Marcon Mello

**São Paulo / SP**

## Resumo

Mesmo com o atual cenário econômico não muito favorável, o número de micro e pequenas empresas vêm crescendo muito no Brasil nos últimos anos, sendo responsáveis por mais de um terço do PIB nacional. Para que essas empresas possam se manter competitivas no mercado e também ter uma organização melhor das entradas, saídas e processos que ocorrem dentro dela, é necessário um sistema integrado de gestão (ERP). Algumas dessas micro e pequenas empresas optam em desenvolver seu próprio ERP ao invés de contratar este serviço de terceiros, o que proporciona um sistema criado especificamente para essa empresa e mais velocidade no suporte, entretanto, ao mesmo tempo, pode ser que apresente alguns erros por ter uma equipe de desenvolvimento pequena e também por não ter um processo de desenvolvimento tão profissional. O presente trabalho aborda os erros que acontecem no processo de teste do desenvolvimento e sugere que, através do uso do teste de caixa branca, que é um teste mais completo e verifica todo o código, muitos bugs e erros no código seriam evitados, fazendo com que mesmo em micro e pequenas empresas (não focadas em TI) o desenvolvimento de seu software tenha uma qualidade melhor.

**Palavras Chave:** Testes; Caixa Branca; MPE; ERP.

## Abstract

Even with the current unfavorable economic scenario, the number of micro and small companies has been increasing in Brazil in the last few years, accounting for more than a third of the national GDP. In order for these companies to keep competitive in the market and also to have a better organization of the inputs, outputs and processes that take place within it, an integrated management system (ERP) is necessary. Some of these micro and small businesses choose to develop their own ERP instead of hiring this service, which provides a system created specifically for that company and a faster support service, however, at the same time, it may present some errors because these companies usually have a small development team and also for don't have a very professional development process. This paper addresses the errors that occur in the development testing process and suggests that, through the use of the white box test, which is a more complete test and verifies the entire code, many bugs and errors in the code would be avoided, that even in micro and small companies (not focused on IT) the development of its software has a better quality.

**Key words:** Tests; white box; micro and small companies; ERP.

## 1. Introdução

Segundo o SEBRAE (2014), existem mais de nove milhões de micro e pequenas empresas (MPE) ativas no Brasil, representando pouco mais de 93% do total de empresas no país, essas MPE são responsáveis por mais de 27% do PIB nacional. Muitas dessas empresas utilizam softwares de terceiros para organização e gerenciamento de processos, entretanto existe uma parte dessas MPE que preferem ter um setor de Tecnologia da Informação (TI) interno para desenvolver o próprio sistema integrado de gestão empresarial (ERP) e assim não precisar usar softwares alheios e ter um sistema feito exclusivamente para a própria empresa.

Quando uma empresa, que não tem como foco principal o desenvolvimento de software, opta por desenvolver o próprio ERP ao invés de utilizar softwares de terceiros, geralmente, ocorrem algumas falhas no processo, seja por falta de conhecimento de todos os passos para se desenvolver um software ou por falta de pessoas na equipe de TI. Um dos erros mais comuns acontece na hora do teste do software, onde, por falta de experiência, é feito apenas um teste funcional que não abrange todas as possibilidades de erro, o que resulta em muitos erros depois que o software já foi implantado (SISPRO, 2017).

É com base em situações como a citada acima que será desenvolvido o artigo, sugerindo um método para que muitos erros sejam descobertos na fase de testes mesmo nessas empresas que não são focadas no desenvolvimento de softwares.

## **2. Desenvolvimento**

### **2.1 Micro e pequenas empresas**

Um dos principais critérios para definir o “tamanho” de uma empresa, ou seja, se ela é micro, pequena, média ou grande é o faturamento ou receita anual bruta. Segundo o LIMA (2001), “existem duas esferas para definição do porte: a federal e a estadual.” No âmbito federal, é considerada microempresa aquela que possui receita anual bruta igual ou inferior a R\$ 360 mil. Já as empresas de pequeno porte são as que têm faturamento superior a R\$ 360 mil e igual ou inferior a R\$ 3 milhões e 600 mil. Cada estado pode flexibilizar esses valores como forma de ajudar as empresas para fins de recolhimento de tributos estaduais. Essas empresas, dependendo do segmento em que atuam, podem aderir ao Imposto Simples (sistema integrado de pagamento de impostos e contribuições das microempresas e empresas de pequeno porte), possuindo legislação própria.

Existe também um critério baseado na quantidade de funcionários da empresa, que varia segundo diferentes autores. Na indústria, as micro empresas possuem menos de 20 funcionários e as pequenas até 99 (BANTERLI, MANOLESCU, 2007; MAPARI, 2016). No comércio e nos serviços esses limites são um pouco menores, de até 9 funcionários nas micro e até 49 nas pequenas.

Segundo dados divulgados pelo Sebrae em 2014, a participação das micro e pequenas empresas brasileiras somadas representa 27% de todo o Produto Interno Bruto (PIB) do Brasil.

Ao compararmos com números das décadas passadas, veremos que a participação das micro e pequenas empresas na economia brasileira vêm crescendo acentuadamente. Em 1985, representavam 21% do PIB. Já em 2001, 23,2%. Em termos de valores absolutos, de 2001 a 2011, o faturamento das micro e pequenas empresas saltou de R\$ 144 bilhões para R\$ 599 bilhões, em valores da época (SEBRAE, 2014).

Por setor, a importância das micro e pequenas empresas é maior quando consideramos o comércio, uma vez que representam 53,4% do PIB deste segmento. No setor de Serviços, os pequenos negócios representam 36,3% da produção. Já na Indústria, representam 22,5%.

Quando consideramos a participação no mercado de trabalho, percebemos que as micro e pequenas empresas tem uma relevância ainda maior na economia brasileira. Segundo BÓ (2010), os pequenos negócios empregam 52% da mão de obra formal do Brasil. Além disso, os salários pagos por elas respondem por 40% da massa salarial brasileira.

São empresas que vão desde um escritório em casa (office home), passando por associações, franquias, cooperativas, empresas familiares e administração profissional. Todas são responsáveis por grande parte do mercado produtivo nacional.

As sociedades dessas empresas, pelo Código Civil, podem ser: Simples, formada por pessoas que exercem atividade intelectual, científica, literária ou artística, salvo se o exercício da profissão constituir elemento da empresa (a montagem de um consultório por dois dentistas, por exemplo); Empresária, que é aquela que exerce atividade econômica organizada para a produção ou prestação de serviço, constituindo elemento da empresa. São por exemplo as sociedades comerciais em geral. Esse tipo de sociedade deve ser constituída por um dos seguintes tipos: Sociedade em Nome Coletivo, em Comandita Simples, Limitada, Anônima e em Comandita por Ações. (SEBRAE, 2006)

## **2.2 Utilização de ERP em pequenas empresas**

Todas MPE que desejam crescer e operar em um mercado cada dia mais globalizado necessitam ter seu ERP. Hoje em dia, estes sistemas não são apenas indicados para médias e grandes empresas, mas sim para qualquer tipo de empresa que deseja um trabalho mais organizado e com chances de maior lucro.

Atualmente, segundo LEITE (2014), a competitividade não permite mais que os negócios funcionem apenas com base em planilhas e aplicações que não tem suas informações compartilhadas.

Por isso, MPEs precisam operar com o suporte de um ERP para facilitar e integrar os processos de seu negócio.

Um dos principais motivos para as MPE utilizarem um sistema ERP é manter sua competitividade no mercado. Para isso, é necessário deixar de lado processos manuais, que aumentam consideravelmente os riscos de erro.

As MPE crescem a partir da demanda de seus clientes e precisam estar sempre preparadas para atendê-los sem atrasos, com o risco de perderem bons negócios. Adiar a implantação de um sistema ERP pode fazer com que a empresa se torne insustentável conforme a demanda aumenta, já que fica cada vez mais difícil a organização (RAFAEL, 2014).

Pequenas empresas também precisam implantar sistemas ERP para garantir não apenas seu crescimento, como também sua sustentabilidade a médio e longo prazo em um mercado onde as grandes empresas tendem a “engolir” as micro e pequenas empresas (PRATES, 2004).

Um dos maiores ganhos trazidos pelo sistema ERP é a gestão integrada e unificada de todos os processos da empresa. Na prática, isso quer dizer que todos os setores operam em conjunto sob os olhos do gestor. O administrativo não fica distante da produção, e os dois não andam em desalinho (MEGA, 2016). Com isso, o gestor é capaz de gerenciar melhor os trabalhos, identificar pontos fracos e sanar possíveis problemas.

### **2.2.1 Consequências de utilizar ERP de terceiros**

Ao utilizar um sistema de ERP desenvolvido por terceiros a empresa está adquirindo um ERP que fora desenvolvido previamente, ou seja, um ERP padrão que pode se encaixar na cultura de diversas empresas diferentes, com módulos genéricos que não foram feitos especificamente para a cultura dela.

Ainda que sejam adquiridas algumas personalizações, o ERP não será perfeito para a empresa contratante, além de ter um custo com contrato, implantação e suporte. Caso a empresa que contratou os serviços do sistema de ERP não goste de algo ou veja, no sistema, algum problema, isso não é resolvido na hora, existe toda uma burocracia que deve ser seguida por se tratar de sistema de terceiros, além do tempo de espera para o problema ser resolvido, que tem variação muito grande, pois é um problema que é reportado para outra empresa que gerencia os chamados de todas as contratantes daquele ERP. Logo um erro simples de código pode demorar dias para ser resolvido por questão de prioridade e fila de atendimento na empresa dona do ERP.

Claro que ao implantar um ERP de outra empresa, quem contratou também tem algumas vantagens, como não ter a necessidade de uma área de TI para cuidar de análise e desenvolvimento de sistemas e não ter que contratar mais colaboradores para manter essa área a empresa.

### **2.2.2 Consequências de criar seu próprio ERP**

De acordo com BLUECORE (2016), quando uma empresa opta por desenvolver seu próprio ERP, no geral, é um ERP que abrange toda a empresa, criado por quem sabe das necessidades a serem atendidas e que conhece a cultura de onde esse ERP será implantado, desenvolvendo assim um sistema específico e exatamente como a empresa precisa, não sendo um módulo genérico, mas bem mais completo.

Tendo uma equipe, ainda que pequena, que desenvolva seu ERP interno, as MPE tendem a ganhar em agilidade de desenvolvimento e em correção de erros e bugs, pois tudo é feito internamente e a burocracia para abrir um chamado e falar com suporte/desenvolvimento é muito menor do que quando se precisa abrir um chamado em uma empresa contratada.

Quando MPE tem sua própria equipe de desenvolvimento trabalhando em um ERP, por serem equipes menores que em grandes empresas, algumas etapas de testes não são feitas, uma dessas etapas é o teste de caixa branca, que muitas vezes é deixado de lado e substituído apenas pelo teste funcional (caixa preta) que não é tão completo e, por isso, alguns erros são encontrados apenas quando o sistema já se encontra na produção (SISPRO, 2017).

## **2.3 Teste de caixa branca**

O teste de caixa branca, diferente do teste da caixa preta (também conhecido como teste funcional), visa avaliar a estrutura de um software, por esse motivo também é chamado de teste estrutural (PRESSMAN, 2006). Durante o teste de caixa branca são analisados os componentes internos de um programa para

encontrar possíveis erros que venham a prejudicá-lo durante o seu uso. Avalia-se a operação interna, os códigos-fontes, linha a linha do software.

Para que o teste da caixa branca seja feito é preciso conhecimento técnico, já que serão avaliados os códigos e a estrutura do programa. No momento de montar o teste, é necessário avaliar a criticidade do programa que está sendo testado, sua complexidade e sua estrutura para determinar quais os critérios que serão utilizados..

O teste da caixa branca diferencia-se do da caixa preta em relação ao objetivo. Enquanto o de caixa branca é usado para avaliar a estrutura interna de um software, o da caixa preta foca-se na externa (MIRANDA, 2017). No teste da caixa preta as interfaces são avaliadas para se verificar se apresentam erros na hora em que é simulado o uso por um usuário final. Nesse tipo de teste é importante avaliar como os usuários finais utilizariam o programa, quais os possíveis caminhos que eles poderiam fazer e como o programa responderia a isso. É um teste que não necessita de grande domínio técnico, pois é mais simples de ser analisado. No teste da caixa-branca é diferente, já que é necessário entender de códigos e conhecer o código fonte do programa para que se possa fazer o teste.

### **Exemplo de teste de caixa branca:**

Para facilitar o entendimento de como funciona o teste de caixa branca, abaixo temos, como exemplo, o código de uma função que realiza uma busca de um elemento dentro de uma lista ou de um vetor, e utilizaremos este código para mostrar todos os passos necessários para a realização do teste.



```

Public static int buscaValor(int[] array, int valor){
    Int inicio = 0;
    Int final = array.length-1;
    Int return = -1;
    while(inicio <= final){
        int meio = (inicio + final)/2;
        if(array[meio] == valor){
            return = meio;
            fim = -1;
        } else if(valor > array[meio]){
            inicio = meio + 1;
        } else {
            fim = meio - 1;
        }
    }
    return return;
}

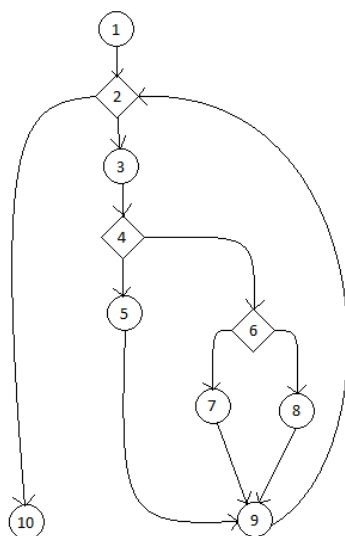
```

```

Public static int buscaValor(int[] array, int valor){
    1  Int inicio = 0;
        Int final = array.length-1;
        Int return = -1;
    2  while(inicio <= final){
    3      int meio = (inicio + final)/2;
    4      if(array[meio] == valor){
    5          return = meio;
    6          fim = -1;
    7      } else if(valor > array[meio]){
    8          inicio = meio + 1;
    9      } else {
    10         fim = meio - 1;
    }
    }
    return return;
}

```

A primeira coisa que devemos fazer após olhar o código é separá-lo em diferentes blocos e numerá-los. O que determina esses blocos são os comandos de decisões e repetições, no caso do código acima esses comandos são “if” e “while”, pois é a partir desses comandos que ocorre uma divisão de fluxo. Com essa divisão em blocos feita, podemos criar um fluxograma do código.



Agora que o fluxo do código foi mapeado é necessário calcular a complexidade do grafo, que é a quantidade de caminhos que o código pode ter até chegar ao seu fim, assim saberemos a quantidade de testes necessária para verificar no código.

A fórmula para calcular os caminhos é  $C = A - N + 2$ , onde C é a quantidade de caminhos, A é o número de arestas e N é o número de nós que temos no fluxograma.

Para o exemplo a conta ficaria  $C = 12 - 10 + 2$ , logo  $C = 4$ , o que significa que temos 4 caminhos independentes para chegar ao fim do código, esses caminhos são:

1 – {1,2,10}

2 – {1,2,3,4,5,9,2,10}

3 – {1,2,3,4,6,7,9,2,10}

4 – {1,2,3,4,6,8,9,2,10}

Depois de definir os caminhos que o código pode percorrer, temos que fazer uma análise para que possamos definir os valores necessários para testar cada um dos caminhos, sabendo que os valores que podemos inserir são o tamanho do vetor e o número que está sendo buscado. Definiremos os valores necessários para testar cada caminho.

Para testarmos o caminho número 1 sabe-se que o tamanho do vetor tem que ser menor que o valor “início” para que o código não entre no “while”, a única forma de isso acontecer é se o tamanho do vetor for nulo (esteja vazio), já que a variável “início” é iniciada com valor 0.

Para testar o caminho número 2, sabe-se que o código entra no “while” logo o tamanho do vetor tem que ser maior ou igual a 1, ou seja, ter pelo menos um elemento. Nesse caminho a condição do nó 4 deve ser verdadeira, para isso o valor que está sendo procurado tem que estar no meio do vetor, logo nosso vetor precisa conter um número ímpar de valores e o procurado precisa, necessariamente estar no meio do vetor, como por exemplo:

Array[4,5,6,7,8] e o valor procurado ‘6’.

Para testarmos o caminho 3, é necessário que a condição do nó 4 seja falsa e a do nó 6 seja verdadeira, logo, precisamos que o valor procurado seja maior que o valor do meio do vetor, como por exemplo:

Array[7] e o valor procurado '8'.

Já para testarmos o caminho 4, precisamos que a condição no nó 4 seja falsa e a do nó 6 também, então podemos utilizar um exemplo ao contrário do caminho 3, onde o valor procurado é menor que o do meio do vetor, como:

Array[8] e valor procurado '7'.

Assim, temos os valores necessários para testar todas as possibilidades de caminho que um código pode seguir, prevenindo erros que podem passar despercebidos em um teste de caixa preta, onde a preocupação é apenas a saída e não todo o caminho realizado.

## **2.4 Como utilizar o teste de caixa branca em pequenas empresas**

Nas MPE, mesmo tendo uma equipe menor do que em médias e grandes empresas, não é complicado realizar o teste de caixa branca, apenas é necessário escolher alguém da equipe de desenvolvimento que tenha mais conhecimento e familiaridade com códigos e designar a essa pessoa a função de realizar esse teste (SILVA, 2015).

O processo de desenvolvimento não sofreria uma mudança drástica, apenas seria acrescentado um novo teste que, caso a equipe prefira, pode até substituir o teste de caixa preta. Assim, após o desenvolvimento de cada módulo, nova função ou correção de algum código, o desenvolvedor criará um plano de teste, dizendo o que o código deve realizar em cada caminho possível, outro desenvolvedor, com o plano de teste em mãos, aplicaria o teste de caixa branca, verificaria cada linha e se há a necessidade de enviar o código de volta para o desenvolvedor para que ele aplique alguma correção. Caso não haja a necessidade de correções o programa já pode ser passado para a próxima fase, a de implementação.

### **3. Conclusão**

O trabalho desenvolvido procurou explicar um pouco sobre as micro e pequenas empresas e sua grande participação na economia do Brasil e como com os anos vem crescendo cada vez mais. Além disso, frisar que um sistema integrado (ERP) é de muita importância para a organização dessas empresas ainda que não sejam de grande porte, por isso precisando sempre prezar pela qualidade do processo de teste para entregar sempre um resultado confiável para seu cliente final (mesmo que o cliente final seja a própria empresa).

Sabendo que algumas dessas MPE, ao invés de contratar um serviço de ERP optam por ter uma equipe de desenvolvimento criando um software ERP próprio (mesmo que a empresa não seja de software). Foi apresentada, no artigo, a opção de, na fase de teste do software, utilizar o teste de caixa branca, muitas vezes não utilizado em micro e pequenas empresas por falta de conhecimento ou experiência da equipe de desenvolvimento.

A ideia de utilizar o teste de caixa branca nessas MPE sugere que, ao incluir este teste nos processos de desenvolvimento de um software, o resultado final não terá, ou terá bem menos, bugs ou erros no código, pois ao realizar o teste de caixa branca todas as linhas do código seriam testadas, assim evitando que algo passe despercebido como poderia passar em um teste funcional simples (teste de caixa preta).

## 4. Referências

BANTERLI, F. R; MANOLESCU, F. M. K. **As micro e pequenas empresas no Brasil e sua importância para o desenvolvimento do país**, 2007. Disponível em: <[http://www.inicepg.univap.br/cd/INIC\\_2007/trabalhos/sociais/inic/INICG00512\\_01O.pdf](http://www.inicepg.univap.br/cd/INIC_2007/trabalhos/sociais/inic/INICG00512_01O.pdf)> Acesso em 11 de junho de 2017.

BLUECORE. **ERP, as vantagens de criar o seu próprio**, 2016. Disponível em: <<https://www.bluecore.com.br/erp/vantagens-de-criar-proprio-erp>> Acesso em 11 de junho de 2017.

BÓ, R. A. D. **A importância das micro e pequenas empresas para a economia do país**, 2010. Disponível em: <<http://www.administradores.com.br/artigos/economia-e-financas/a-importancia-das-micro-e-pequenas-empresas-para-a-economia-do-pais/48688/>> Acesso em 21 de maio de 2017.

LEITE, M. **ERP para pequenas empresas: por que implantar?**, 2014. Disponível em: <<http://www.artsoftsistemas.com.br/blog/erp-para-pequenas-empresas-por-que-implantar>> Acesso em 18 de junho de 2017.

LIMA, E.O. **As definições de micro, pequena e média empresas brasileiras como base para a formulação de políticas públicas**, 2001. Disponível em: <<http://www.anegepe.org.br/edicoesanteriores/londrina/GPE2001-03.pdf>> Acesso em 12 de maio de 2017.

MAPARI. **Definição de Grande, Média, Pequena e Microempresa**, 2016. Disponível em: <<https://economiasfinancas.com/2016/definicao-grande-media-pequena-microempresa/>> Acesso em 21 de maio de 2017.

MEGA. **5 vantagens de implantar um ERP**, 2016. Disponível em: <<https://www.mega.com.br/blog/5-vantagens-de-implantar-um-erp-2825/>> Acesso em 10 de julho de 2017.

MIRANDA Jr, Ottoni. **Testes de Caixa Branca e Caixa Preta**. Disponível em: <<http://www.tesestec.com.br/pasteurjr/TCPB.pdf>> Acesso em 01 de junho de 2017.

PRATES, G.A; Ospina, M. P. **Tecnologia da informação em pequenas empresas: fatores de êxito, restrições e benefícios**, 2004. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1415-65552004000200002](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-65552004000200002)> Acesso em 10 de maio de 2017.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill, 2006, 720p.

RAFAEL, G.C. **A realidade de ambientes de TI em Micro e Pequenas Empresas (MPE)**, 2014. Disponível em: <<https://www.profissionaisiti.com.br/2014/02/a-realidade-de-ambientes-de-ti-em-micro-e-pequenas-empresas-mpe/>> Acesso em 15 de maio de 2017.

SEBRAE. **Micro e pequenas empresas geram 27% do PIB do Brasil**. Disponível em: < <https://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/micro-e-pequenas-empresas-geram-27-do-pib-do-brasil,ad0fc70646467410VgnVCM2000003c74010aRCRD>> Acesso em 10 de junho de 2017.

SEBRAE. **Participação das Micro e Pequenas Empresas na Economia Brasileira**, 2015.

Disponível em:

<[http://www.bibliotecas.sebrae.com.br/chronus/ARQUIVOS\\_CHRONUS/bds/bds.nsf/e55cdb1932bc40120b21bf4d277bb6ea/\\$File/5307.pdf](http://www.bibliotecas.sebrae.com.br/chronus/ARQUIVOS_CHRONUS/bds/bds.nsf/e55cdb1932bc40120b21bf4d277bb6ea/$File/5307.pdf)> Acesso em 10 de julho de 2017.

SILVA, Dianne Dias. **Melhoria do Processo de Teste para as Micro e Pequenas Empresas Brasileiras**, 2015. Disponível em: <

[http://www.inf.ufg.br/mestrado/sites/portal.inf.ufg.br/mestrado/files/uploads/Disserta%C3%A7%C3%A3o\\_Dianne%20Dias%20Silva\\_Vers%C3%A3o%20Final.pdf](http://www.inf.ufg.br/mestrado/sites/portal.inf.ufg.br/mestrado/files/uploads/Disserta%C3%A7%C3%A3o_Dianne%20Dias%20Silva_Vers%C3%A3o%20Final.pdf) >

Acesso em 10 de junho de 2017.

SISPRO. **5 desvantagens que a empresa assume ao desenvolver seu próprio**

**ERP**. Disponível em: <<http://www.sispro.com.br/blog/erp/5-desvantagens-que-a-empresa-assume-ao-desenvolver-seu-proprio-erp/>> Acesso em 09 de julho de 2017.