



INTER
FACES
CIENTÍFICAS

EXATAS E TECNOLÓGICAS

ISSN IMPRESSO - 2359-4934

E- ISSN - 2359-4942

DOI - 10.17564/2359-4934.2018v2n3p31-44

ANÁLISE COMPARATIVA ENTRE OS MODELOS DE DESENVOLVIMENTO ASP.NET WEBFORMS E ASP.NET MVCPOR MEIO DE UM EXPERIMENTO CONTROLADO

COMPARATIVE ANALYSIS BETWEEN THE ASP.NET WEBFORMS AND ASP.NET MVC DEVELOPMENT MODELS THROUGH A CONTROLLED EXPERIMENT

ANÁLISIS COMPARATIVO ENTRE LOS MODELOS DE DESARROLLO ASP.NET WEBFORMS Y ASP.NET MVC A TRAVÉS DE UN EXPERIMENTO CONTROLADO

Anderson Santos Barroso ¹

RESUMO

No mundo corporativo atual, a plataforma de desenvolvimento ASP.NET tem sido amplamente utilizada para o desenvolvimento de aplicações Web orientadas a objetos, por se tratar de uma tecnologia que acopla vários modelos em um único ambiente de trabalho. Este artigo tem como objetivo a apresentação de um experimento controlado para a avaliação entre dois modelos de desenvolvimento ASP.NET: WebForms e MVC (Model View Controller). O artigo mostra evidências que ajudam a identificar qual modelo se enquadra melhor para a utilização em um ambiente de desenvolvimento corporativo. O experimento foi realizado em uma grande Instituição de Ensino Superior privada, com profissionais

de diferentes níveis de experiência. Como resultado, do ponto de vista da eficiência, houve evidências de que WebForms leva vantagem no quesito número de linhas e o MVC se sobressai no quesito tempo de codificação. Quanto à eficácia e ao quesito quantidade de erros, não foi possível encontrar diferenças.

PALAVRAS-CHAVE

ASP.NET, WebForms, MVC, Experimentação

ABSTRACT

In today's corporate world, the ASP.NET development platform has been widely used for the development of object-oriented Web applications, because it is a technology that couples multiple models into a single work environment. This article aims to present a controlled experiment for the evaluation between two ASP.NET development models: WebForms and Model View Controller (MVC). The article shows evidence that helps identify which model fits best for use in an enterprise development environment. The experiment was carried out in a large Private Higher Education Institution, with professionals of different

levels of experience. As a result, from the efficiency point of view, there was evidence that WebForms takes advantage of the number of lines and MVC excels in coding time. As to effectiveness and the number of errors, it was not possible to find differences.

KEYWORDS

ASP.NET. WebForms. MVC. Experimentation

RESUMEN

En el mundo corporativo actual, la plataforma de desarrollo ASP.NET ha sido ampliamente utilizada para el desarrollo de aplicaciones web orientadas a objetos, por tratarse de una tecnología que acopla varios modelos en un único ambiente de trabajo. Este artículo tiene como objetivo la presentación de un experimento controlado para la evaluación entre dos modelos de desarrollo ASP.NET: WebForms y MVC (Model View Controller). El artículo muestra evidencias que ayudan a identificar qué modelo se encuadra mejor para su uso en un entorno de desarrollo corporativo. El experimento fue realizado en una gran

Institución de Enseñanza Superior privada, con profesionales de diferentes niveles de experiencia. Como resultado, desde el punto de vista de la eficiencia, hubo evidencias de que WebForms lleva ventaja en el número de líneas y el MVC sobresale en el tiempo de codificación. En cuanto a la eficacia ya la cantidad de errores, no fue posible encontrar diferencias.

PALABRAS CLAVE

ASP.NET, WebForms, MVC, Experimentación

1 INTRODUÇÃO

O crescimento recente da economia digital, consumo de TI e do movimento de negócios de softwares aumentam cada vez mais a popularidade da web, fazendo-se necessárias aplicações efetivas, com alta escalabilidade, disponibilidade e desenvolvimento de softwares cada vez mais rápidos (GENS et al., 2011).

Aliado a isto, o desenvolvimento de aplicações web tem crescido rapidamente nas últimas décadas com a consolidação da internet, despontando em diversos meios, desde comercialização de produtos até meios de marketing digital. Com o avanço das redes e máquinas com maior poder de armazenamento e processamento, o desenvolvimento de aplicações web está tornando-se cada vez mais robusto e com maior capacidade de interação com os usuários (KERER et al., 2001; ZHUOYI et al., 2012).

Aplicações Web são sistemas que possuem um alto grau de interação, além de atender simultaneamente diversos usuários, distribuídos em locais distintos fisicamente (HENDRICKSON et al., 2002).

Ao longo dos últimos anos, a arquitetura de software sofreu alterações na sua concepção, identificando que métodos de desenvolvimento de softwares podem melhorar o nível de abstração, de reutilização e de eficiência do software (MING-XIA et al., 2010)

Atualmente, existe uma grande variedade de sistemas de tecnologia Web, para aplicações em intranets e internet, sendo que um dos mais representativos, do ponto de vista de desenvolvedor, é o ASP.NET da Microsoft (WANG; YANG, 2008).

O ASP.NET oferece suporte às estruturas de *WebForms* e MVC (ModelView Controller), os quais são os dois modelos de desenvolvimento objetos de estudo do experimento controlado proposto neste artigo, perfazendo a seleção de desenvolvedores de sistemas com nível de experiência baixa, intermediária e avançada.

Segundo Basili e Weiss (1992), muitos estudos foram feitos sobre a aplicação de métricas e modelos em ambientes industriais, a fim de torná-los mais eficazes, sempre focando nos objetivos específicos aplicados a todos os ciclos de vida de produtos, processos e recursos,

interpretados com base na caracterização e compreensão do contexto da organização, ambiente e metas. Baseado nisso, o experimento aqui apresentado pretende auxiliar no entendimento e melhoria de parte do processo de software do ambiente industrial experimentado, por meio da análise de métricas para Tempo de Codificação, Quantidade de Linhas de Código e Número de Erros. Essa análise ajudará a responder as questões de pesquisas propostas.

O intuito do experimento é comparar as estruturas *WebForms* e MVC, apresentando resultados que auxiliem gerentes e desenvolvedores a tomarem decisão quanto à estrutura a ser utilizada em seu ambiente de trabalho.

Primeiramente, serão apresentados os conceitos básicos de *WebForms* e MVC, depois será definido o planejamento do experimento. O próximo passo será definir como o experimento foi executado e, por último, serão apresentadas as conclusões derivadas da análise dos resultados do experimento.

2 CONCEITOS DE WEBFORMS E MVC

2.1 WEBFORMS

WebForms são um modelo de desenvolvimento proposto pela Microsoft que introduz formulários modelos, utilizados para o desenvolvimento de aplicações Web (MING-XIA et al., 2010).

De acordo com o modelo de controle, os desenvolvedores de páginas *front-end* podem arrastar e soltar os componentes, enquanto os desenvolvedores de *back-end* podem usar qualquer tipo de linguagem na plataforma, para fazer a programação. Na sua criação, os *Webforms* pregavam uma maior facilidade para programação web, por parte dos programadores oriundos de programas desktops (ZONGYU, 2012).

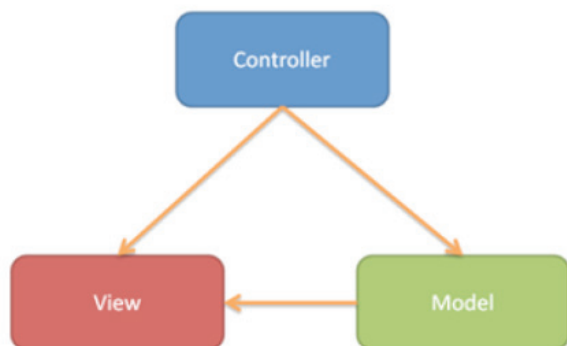
2.2 MVC

MVC define o padrão de design em três partes básicas: *Model*, *View* e *Controller*, a fim de promover o desenvolvimento Web (FIGURA 1).

O framework MVC, suportado pela plataforma ASP.

NET, é chamado de ASP.NET MVC, propondo o desenvolvimento de aplicações em um modo de acoplamento mais flexível (ROSA et al., 2015).

Figura 1 – Ilustração do modelo MVC



Fonte: ROSA et al. (2015)

A camada *model* é responsável pelo acesso a dados e retorna um conjunto de objetos que podem ser acessados pela camada *view*. Esta, por sua vez, é responsável pela interface com o usuário. A *camada-controller* é responsável pela lógica efetiva da aplicação, abrangendo a interação do usuário final com a aplicação. Esta camada manipula os dados da camada *model* e utiliza a camada *view* para mostrar o resultado para o usuário (MASOUND, 2006).

2.3 WEBFORMX MVC

Tanto *Webforms* quanto MVC possuem características próprias que podem ser avaliadas por quem desenvolve aplicações Web. Ambos geram páginas *ASPX*, rodam no servidor web IIS (Internet Information Services), acessam livremente os dados e são desenvolvidos no Visual Studio, embora para o MVC seja necessário um complemento a mais (MASOUND, 2006).

O MVC, por ser dividido em camadas, possibilita a separação de responsabilidades na aplicação, ou seja, cada camada faz a sua parte, contribuindo na facilidade de manutenção da aplicação e no total controle do *HTML*

e *JavaScript* gerado, no entanto, não disponibiliza controles prontos como o *WebForms* (ZONGYU, 2012).

3 DEFINIÇÃO E PLANEJAMENTO DO EXPERIMENTO

Nesta e nas duas próximas seções, nosso trabalho é apresentado como um processo experimental. O mesmo segue diretrizes de Wohlin e outros autores (2000). O foco desta seção será a definição do objetivo e o planejamento do experimento.

3.1 DEFINIÇÃO DO OBJETIVO

Os elementos principais do experimento são as variáveis, os objetos, os participantes, o contexto do experimento, as hipóteses e o tipo de projeto do experimento. Com estes elementos, tem-se por objetivo a realização de estudos que possam evidenciar a melhora de algum processo de desenvolvimento. Esta melhora está relacionada à verificação de teorias formuladas por meio das hipóteses do experimento (TRAVASSOS, 2002).

O objetivo deste experimento é avaliar, por meio de um experimento controlado, o uso dos modelos de desenvolvimento de aplicações *WebForms* e MVC, por meio de uma plataforma de desenvolvimento orientada a objetos, o Visual Studio.

O experimento terá como alvo programadores de uma grande Instituição de Ensino, com, no mínimo, 1 ano de experiência no mercado e 6 meses de experiência em programação web. O objetivo foi formalizado, utilizando o modelo GQM proposto por originalmente por Basili (1984): **Analisar** o uso de dois modelos de desenvolvimento *ASP.NET*, **com a finalidade** de avaliar a utilização do modelo MVC **contra** o modelo *WebForms*, **com respeito à** eficiência e eficácia do processo de desenvolvimento de código, **do ponto de vista de** programadores e gestores de desenvolvimento de softwares, **no contexto de** programadores de uma Instituição de Ensino com área própria de TI.

3.2 PLANEJAMENTO

3.2.1 FORMULAÇÃO DAS HIPÓTESES

As questões de pesquisa que precisam ser respondidas com o estudo proposto, em comparação com o desenvolvimento por meio de *WebForms*, são: 1ª) O desenvolvimento utilizando ASP.NET MVC pode aumentar a produtividade dos programadores no processo de desenvolvimento? 2ª) O desenvolvimento, utilizando ASP.NET MVC pode reduzir a quantidade de erros de codificação no processo de desenvolvimento? 3ª) O desenvolvimento, utilizando ASP.NET MVC pode reduzir o número de linhas de código no processo de desenvolvimento?

Para responder a essas questões serão utilizadas métricas, da Engenharia de Software, que têm sido aplicadas na medição, tanto do processo quanto do produto de software. As métricas escolhidas foram: Tempo de codificação; Quantidade de erros e Quantidade de linhas de código, pois elas auxiliam na estimativa do custo de execução de uma tarefa.

Tendo os objetivos e métricas definidos, serão consideradas as seguintes hipóteses:

Hipótese 1

- $H0^{\text{tempo}}$: ASP.NET MVC e ASP.NET *WebForms* possuem a mesma eficiência ($\mu\text{tempoMVC} = \mu\text{tempo WebForms}$);
- $H1^{\text{tempo}}$: O desenvolvimento utilizando ASP.NET MVC é mais eficiente que o ASP.NET *WebForms* ($\mu\text{tempoMVC} < \mu\text{tempo WebForms}$);

Hipótese 2

- $H0^{\text{erros}}$: ASP.NET MVC e ASP.NET *WebForms* possuem a mesma eficácia ($\mu\text{errosMVC} = \mu\text{erros WebForms}$);
- $H1^{\text{erros}}$: O desenvolvimento utilizando ASP.NET MVC possui maior eficácia que o ASP.NET *WebForms* ($\mu\text{errosMVC} < \mu\text{erros WebForms}$);

Hipótese 3

- $H0^{\text{LOC}}$: ASP.NET MVC e ASP.NET *WebForms* geram a mesma quantidade de linhas de código ($\mu\text{codigosMVC} = \mu\text{codigos WebForms}$);
- $H1^{\text{LOC}}$: O desenvolvimento utilizando ASP.NET MVC

e *WebForms* gera números de linhas de código diferentes ($\mu\text{codigosMVC} \neq \mu\text{codigos WebForms}$);

Para as hipóteses, foram consideradas as variáveis dependentes, independentes e intervenientes a seguir:

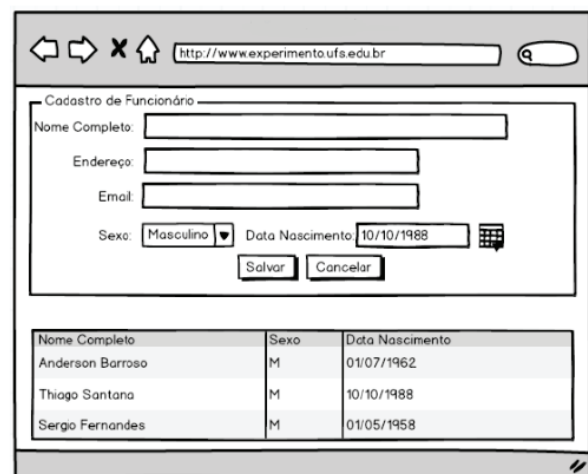
3.2.1.1 VARIÁVEIS INDEPENDENTES

A seguir, são descritas as variáveis independentes do experimento.

Descrição de Requisitos Funcionais de Cadastro utilizados no Experimento

O objetivo será criar um Protótipo de Cadastro, a ser implementado para cada modelo de desenvolvimento proposto (FIGURA 2).

Figura 2 – Protótipo de Cadastro a ser implementado pelos participantes



O protótipo de cadastro de funcionário apresenta um formulário com os seguintes campos: Nome Completo, Endereço, Email, Sexo (dropdown menu com 'Masculino' selecionado) e Data Nascimento (10/10/1988). Há botões para 'Salvar' e 'Cancelar'. Abaixo do formulário, há uma tabela com as seguintes informações:

Nome Completo	Sexo	Data Nascimento
Anderson Barroso	M	01/07/1962
Thiago Santana	M	10/10/1988
Sergio Fernandes	M	01/05/1958

Fonte: Elaborado pelo autor

Modelos ASP.NET

Os modelos *ASP.NET*, utilizados neste trabalho, possuem dois tipos de implementações para a realização do experimento:

- 1) *WebForms*: implementação com *ASP.NET WebForms*;
- 2) MVC: implementação com *ASP.NET MVC*;

3.2.1.2 VARIÁVEIS DEPENDENTES

As variáveis dependentes abordadas foram:

- Média de tempo de codificação para MVC e *Webforms*, calculada utilizando um cronômetro, considerando a média de tempo gasto para codificação os procedimentos;
- Média de quantidade de erros para MVC e *Webforms*, calculada totalizando o número de erros, considerando a média de quantidade de erros para codificação dos procedimentos;
- Média de quantidade de linhas de códigos digitadas para MVC e *Webforms*, calculada totalizando o número de linhas digitadas e considerando a média de quantidade de linhas de códigos digitadas para codificação dos procedimentos.

3.2.2 VARIÁVEIS INTERVENIENTES

Podemos destacar duas variáveis que podem influenciar no resultado do experimento:

- A experiência dos desenvolvedores nas estruturas experimentadas;
- O comprometimento do desenvolvedor na execução do procedimento.

3.2.3 SELEÇÃO DOS PARTICIPANTES

Os participantes são os indivíduos selecionados para conduzir o experimento. Eles são os responsáveis por informar parâmetros para o experimento, tal como o valor das variáveis (TRAVASSOS, 2002).

Ao levar em consideração o princípio da aleatoriedade, para evitar interferência no resultado do experimento (SHULL et al., 2001), o processo de seleção dos participantes ocorreu pela conveniência da disponibilidade de programadores, fornecida por uma grande Instituição de Ensino Superior e por sorteio, pelo fato desta liberação se limitar a seis programadores, entre os 10 programadores Web da empresa. Os sorteados possuíam experiência inicial, média e avançada em desenvolvimento com *WebForms* e experiência inicial e média em

desenvolvimento com MVC (TABELAS 1; 2). Os dez desenvolvedores foram classificados de acordo com as suas respectivas experiências e foram sorteados seis desenvolvedores para participar do experimento.

Os colaboradores são do Departamento de Informática da Universidade Tiradentes (www.unit.br), pertencente ao Grupo Tiradentes, o qual possui filiais brasileiras em Aracaju-SE, Feira de Santana-BA, Maceió-AL e Recife-PE. Aracaju foi a cidade escolhida para rodar o experimento, por sediar o centro de desenvolvimento de software. Esta instituição interessou-se em averiguar a melhor estrutura a ser utilizada no seu processo de desenvolvimento WEB, cooperando com os pesquisadores e autores deste artigo, os quais são totalmente imparciais, sem possuir nenhuma demanda, interesse ou preferência por alguma estrutura experimentada.

3.2.4 PROJETO DO EXPERIMENTO

O experimento foi projetado em um contexto pareado, ou seja, todos os participantes testaram os dois modelos de desenvolvimento propostos: MVC e *WebForms*, sendo que apenas três participantes executaram o experimento por vez.

Para entendimento da codificação a ser feita, foi elaborado um Requisito Funcional de Cadastro (visto na seção 3.2.1.1), o qual foi apresentado de forma bem detalhada aos programadores.

O Experimento foi projetado com dois momentos. Contendo duas opções, sendo elas: 1ª) Criação do Protótipo de Cadastro a ser Implementado, utilizando MVC e 2ª) criação do Protótipo de Cadastro a ser Implementado, utilizando *WebForms*.

Dentre os seis programadores selecionados, três foram associados ao subgrupo 1 e 3 ao subgrupo 2. Sendo assim, em um primeiro momento, o subgrupo 1 iniciou com a opção 1 e depois com a opção 2. Em um segundo momento, o subgrupo 2 iniciou com a opção 2 e depois com a opção 1. Além disso, não houve comunicação entre os subgrupos.

A ordem das rodadas será a seguinte:

- Rodada 1: O subgrupo 1 realizará o experimento em MVC;
- Rodada 2: O subgrupo 1 realizará o experimento em *WebForms*;
- Rodada 3: O subgrupo 2 realizará o experimento em *WebForms*;
- Rodada 2: O subgrupo 2 realizará o experimento em MVC;

Essa ordem se dará para mitigar o aprendizado do artefato, em favor de apenas uma das estruturas.

3.2.5 INSTRUMENTAÇÃO

O processo de instrumentação se deu inicialmente com a configuração do ambiente para o experimento e planejamento de coleta dos dados. Este foi realizado no departamento de informática da Universidade Tiradentes (UNIT). Tudo foi realizado em condições idênticas de trabalho, configurando um experimento *in vivo*. Seguem abaixo a tecnologia e as ferramentas usadas.

- Visual Studio 2013 *Community com update5*, que já possui suporte a *WebForms*; SQL Server 2015;
- *Razor View Engine*, que é utilizado especificamente para o desenvolvimento com MVC;
- *Scaffolding*, que é utilizado especificamente para o desenvolvimento com MVC. *Razor* e *Scaffolding* se completam no desenvolvimento com MVC.

4 OPERAÇÃO DO EXPERIMENTO

4.1 PREPARAÇÃO

As seguintes etapas foram seguidas:

Formulário de Caracterização: foi decidido que os participantes iriam responder questões sobre níveis de escolaridade, de experiência nas tecnologias propostas, de experiência na área de TI, de experiência com programação em geral, de quantidade de sistemas já mantidos em geral e discriminados entre MVC e *WebForms*.

Alocação de programadores para criação do Pro-

tótipo de Cadastro a ser Implementado: Os computadores já tinham os pré-requisitos necessários para a execução do experimento. Os participantes não precisaram realizar nenhuma configuração.

4.2 EXECUÇÃO

Após a preparação do experimento e instruções iniciais, os experimentados tiveram cinco minutos para se acomodarem, como no seu trabalho diário.

Cada participante teve dez minutos, para responder o questionário de caracterização. Nesse momento, não houve dúvidas quanto à interpretação do que estava sendo perguntado.

Todos os experimentados concordaram que a comparação entre as duas estruturas era válida e que poderia auxiliá-los em decisões de implementações futuras. Todavia, além de não terem sido treinados, pois já conheciam as ferramentas, não foram informados sobre as hipóteses exatas a serem testadas, evitando o fenômeno *Demand Characterization* (ORNE, 1962).

4.2.1 COLETA DE DADOS

Foi calculado o tempo gasto por cada programador, para codificação em MVC e *WebForms*, levando em consideração o tempo para implementação. Sob supervisão, cada programador informava o término e era registrado o tempo de um cronômetro, utilizado para este fim.

Em um segundo momento, foi totalizado o número de erros encontrados, por cada programador, para o Protótipo de Cadastro implementado.

Por fim, após o experimento, foi analisada a quantidade de linhas de códigos digitadas pelos programadores, para cada Protótipo de Cadastro implementado.

Os resultados desses dados coletados serão apresentados na seção 5 deste trabalho.

4.3 VALIDAÇÃO DOS DADOS

Para realização do experimento, foi considerado um fator, implementação do Protótipo de Cadastro para um Requisito Funcional de Cadastro e dois trata-

mentos, implementação, utilizando MVC e implementação utilizando *WebForms*. Diante deste contexto, foram computadas as médias do tempo de implementação, as médias da quantidade de erros encontrados na implementação e as médias do número de linhas da implementação.

Como auxílio para análise, interpretação e validação, foram utilizados três tipos de testes estatísticos:

- O teste *Shapiro-Wilk* foi utilizado para verificar a normalidade das amostras, pois se tratam de amostras inferiores a 30;
- O método *t-student*, para teste de significância estatística entre pares de modelos. Optamos por esse teste porque nossa amostra é menor que 30;
- O teste de *Wilcoxon*, por se tratar de teste não paramétrico, necessário para o caso no qual ocorreu ausência de normalidade nos dados.

Os testes estatísticos foram feitos utilizando a Ferramenta SPSS – IBM (IBM SPSS, 2015).

5 RESULTADOS

5.1 ANÁLISE E INTERPRETAÇÃO DE DADOS

Para responder às questões, foram analisadas as seguintes variáveis dependentes: a) o tempo para criação de cada procedimento; b) os erros encontrados na criação de cada procedimento e; c) a quantidade de linhas digitadas na implementação de cada procedimento.

Os resultados serão demonstrados por meio de tabelas, para uma melhor compreensão dos dados levantados.

5.1.1 PARTICIPANTES X ESCOLARIDADE X EXPERIÊNCIA

A Tabela 1 representa a experiência profissional do programador. Notem que os programadores possuem escolaridades e experiências variadas.

Tabela 1 – Participantes x Escolaridade x Experiência

Participante	Escolaridade	Experiência com Programação Geral	Experiência com Programação Web Geral	Experiência com os Modelos	
				<i>WebForms</i>	MVC
01	Graduação - Incompleto	Iniciante	Iniciante	Iniciante	Iniciante
02	Pós-Graduação	Avançado	Avançado	Avançado	Intermediário
03	Graduação	Intermediário	Intermediário	Intermediário	Iniciante
04	Técnico Informática	Iniciante	Iniciante	Iniciante	Iniciante
05	Graduação	Intermediário	Intermediário	Intermediário	Intermediário
06	Graduação - Incompleto	Iniciante	Iniciante	Iniciante	-

Até 2 anos: iniciante De 3 a 5 anos: intermediário A partir de 6 anos : avançado

Fonte: Dados da pesquisa.

A Tabela 2 representa a quantidade de sistemas já mantidos por cada participante.

Tabela2 – Participantes x Sistemas Mantidos

Participante	Quantidade de Sistemas Mantidos		
	Geral	WebForms	MVC
01	6	2	1
02	23	13	5
03	10	6	2
04	9	3	1
05	8	5	2
06	3	1	0

Fonte: Dados da pesquisa.

5.1.2 TEMPO DE CODIFICAÇÃO DO EXPERIMENTO

Para responder à primeira questão, a qual relacionada ao tempo de codificação para executar o procedimento, é apresentada a Tabela abaixo:

Tabela3 –Tempo (em minutos) de execução do experimento

Participante	WebForms	MVC	Participante	WebForms	MVC
01	17	9	04	20	45
02	18	13	05	21	22
03	19	22	06	22	42

Fonte: Dados da pesquisa.

A média do tempo, para a execução do procedimento em *WebForms*, foi de 39,83 minutos e, para execução em MVC, foi de 25,5 minutos.

Esse cálculo indica que a média de tempo de codificação gasto, utilizando *WebForms*, é superior à

média de tempo de codificação gasto utilizando MVC, apesar de haver notória maior experiência em *WebForms* dos participantes (TABELA 3). Com isso, podemos supor que a resposta para a primeira questão seria “sim”, ou seja, o uso de MVC pode aumentar a produtividade dos programadores no processo de desenvolvimento de software.

Para embasamento, utilizamos evidências estatísticas conclusivas. Primeiramente, foi definido um nível de significância de 0,05 para todo o experimento, bem como foi aplicado o teste de Shapiro-Wilk, para analisar a distribuição normal da amostra. Foi constatado que o *p-value* da amostra de *WebForms* foi 0.9606 e da amostra de MVC foi 0,2953. As duas amostras tiveram valores maiores que o nível de significância adotado, caracterizando assim, que ambas possuem distribuição de dados normais.

Com a evidência de que as amostras são normais, o teste de hipótese aplicado foi o Teste T, haja vista que a nossa amostra é inferior a 30.

Após aplicação do Teste T, foi verificado que o *p-value* é de 0.004, inferior ao nível de significância adotado. Desta forma, confirmou-se uma evidência de diferença entre as médias de 14,33, o que implica dizer que há uma diferença de tempo entre a execução do experimento, utilizando *WebForms* e MVC. Diante do exposto, a hipótese H_0^{tempo} foi rejeitada. Em tempo, não podemos rejeitar a hipótese H_1^{tempo} .

5.1.3 NÚMERO DE ERROS DO EXPERIMENTO

Como resposta à segunda questão – que trata a quantidade de erros encontrada na execução do experimento – foram coletadas as informações da Tabela 5.

Tabela 5 – Número de erros na execução do experimento

Participante	WebForms(erros)	MVC (erros)	Participante	WebForms(erros)	MVC (erros)
01	0	0	04	2	2
02	2	0	05	1	1
03	2	1	06	3	1

Fonte: Dados da pesquisa.

Ao analisar a Tabela 5, foi verificada a ausência de normalidade dos dados, com p-value abaixo do nível

de significância. Para testar a hipótese em questão, foi utilizado um teste não paramétrico para amostras emparelhadas: o teste *de Wilcoxon*.

Ao aplicar o teste de *Wilcoxon*, foi obtido o *p-value* de 0,102, o qual é superior ao nível de significância de 0,05 adotado. Diante desta evidência, não se pode afirmar que as médias são estatisticamente diferentes.

Após a análise, a hipótese H_0^{erros} não será rejeitada, pois não há evidência estatística suficiente para garantir que o desenvolvimento em MVC gera número de erros diferente do desenvolvimento em *WebForms*.

normais, bem como o teste T, para verificar se a hipótese H_0^{LOC} será ou não rejeitada.

Percebemos que a média da amostra 1 é 168,50, enquanto da amostra 2 é de 259,0. Ao aplicar o teste de *Shapiro-Wilk* foi encontrado um *p-value* de 0.389 para a amostra de *WebForms*, e 0.621, para a amostra de MVC, o que caracteriza que ambas são normais (TABELA 7).

Com a certeza de serem amostras normais, foi aplicado o teste T, o qual resultou em um *p-value* de 0,000, considerando uma diferença de média de 90,5.

5.1.4 NÚMERO DE LINHAS DE CÓDIGO DO EXPERIMENTO

A terceira questão indaga se o desenvolvimento, utilizando ASP.NET MVC pode reduzir o número de linhas de código no processo de desenvolvimento. Para responder essa questão, iremos analisar a Tabela 6.

Tabela 6 – Número de linhas de código (LOC) geradas pelo experimento

Participante	WebForms LOC	MVC LOC	Participante	WebForms LOC	MVC LOC
01	120	172	04	180	234
02	171	295	05	191	264
03	154	275	06	195	314

Fonte: Dados da pesquisa.

Para essa análise, utilizamos novamente o teste de *Shapiro-Wilk*, para averiguar se as amostras são

Tabela 7 – *Shapiro-Wilk Normality Test* para a amostra coletada

Nº de Linhas de Código	Shapiro-Wilk	
	W	p-value
WebForms	0,902	0,389
MVC	0,935	0,621

Fonte: Dados da pesquisa.

Com isso, rejeitamos a hipótese H_0^{Loc} , evidenciando que a média de linhas de código utilizando MVC é maior que a média utilizando *WebForms*.

5.2 AMEAÇAS À VALIDADE

Apesar de terem sido alcançados resultados satisfatórios no experimento, não há como desconsiderar as seguintes ameaças à validade.

Ameaças à validade interna: como os experimentados fizeram os dois Protótipos de Cadastros no mesmo dia e em sequência, eles podem ter aproveitado a experiência ao executar a primeira tarefa para auxiliar na execução da segunda.

Isto foi mitigado com a separação em grupos e com a inversão da ordem entre eles. Além disso, conforme visto anteriormente, entre os intervalos, não houve troca de experiências entre os grupos.

Ameaças à validade externa: como no ambiente experimentado havia poucos programadores aptos a executarem as tarefas nas duas estruturas propostas, os resultados poderiam ter sofrido alterações, se o número de participantes fosse maior.

Ameaças à validade de construção: o Protótipo de Cadastro que os experimentados deveriam reproduzir pode não ter ficado suficientemente intuitivo. Isto foi mitigado com uma explicação minuciosa dos requisitos.

6 TRABALHOS RELACIONADOS

Foi encontrado apenas um trabalho científico que trata de comparação entre os modelos de desenvolvimento *WebForms* e MVC, mas sem fazer experimento controlado e sem efetuar cálculos estatísticos.

Ming-Xia e Keming(2010) fazem um estudo comparativo entre *WebForms*, MVC e MVP (*ModelViewPresenter*), chegando à conclusão de que a escolha do modelo depende das necessidades reais de desenvolvimento, ou seja, se o objetivo é ter mais controle do HTML e acessibilidade, o ideal é MVC. Se o objetivo for um maior controle nas etapas de desenvolvimento, o ideal é *WebForms*. Por fim, se o desenvolvimento não precisar de muita lógica, o ideal é MVP.

O diferencial do trabalho aqui apresentado está no fato de termos utilizado um experimento controlado, para comparar os modelos *Webforms* e MVC. Combinando os dois resultados, podemos associar o trabalho de Ming-Xia e Keming ao nosso experimento e, com isso, gerar subsídios quanto ao modelo ASP.NET ideal a ser utilizado. Com nosso experimento, a instituição colaboradora planejará o uso de MVC, inicialmente em alguns pequenos projetos, haja vista a satisfação com os resultados alcançados.

7 CONCLUSÃO E TRABALHOS FUTUROS

Diante da crescente necessidade da indústria de desenvolver softwares cada vez mais eficiente e eficaz, surgiu a necessidade de se testar possibilidades para que o desenvolvimento tenha sucesso na relação custo *versus* tempo (GENS et al.,2011).

Essa relação fica mais clara no ambiente web, no qual as aplicações suportam acessos de qualquer local do mundo e a performance passou a ser item primordial no desenvolvimento deste tipo de aplicação. Pensando nisso, a *Microsoft* criou uma plataforma de desenvolvimento web (*ASP.NET*) totalmente integrada. Neste contexto, o objetivo deste trabalho foi realizar uma avaliação, utilizando dois modelos de desenvolvimento desta plataforma: *WebForms* e MVC.

O método escolhido para fazer a comparação foi um experimento controlado que analisou três hipóteses distintas. O experimento permitiu a análise comparativa em relação ao tempo de codificação, ao número de linhas de código e à quantidade de erros encontrados, para executar a criação de um Protótipo

de Cadastro nos dois modelos.

Antes de executar o experimento, os participantes preencheram um questionário de caracterização e, após o experimento, responderam um questionário de *feedback*.

Como resposta à primeira pergunta, que foi “o desenvolvimento utilizando ASP.NET MVC pode aumentar a produtividade dos programadores no processo de desenvolvimento?” – foi identificado através do teste T que a hipótese nula foi rejeitada, evidenciando que desenvolver em MVC requer menos tempo de codificação do que realizar a mesma tarefa em *WebForms*.

A segunda pergunta, “o desenvolvimento utilizando ASP.NET MVC pode reduzir a quantidade de erros de codificação no processo de desenvolvimento?” – foi analisada por meio do teste de *Wilcoxon*, o qual evidenciou que não podemos rejeitar a hipótese nula, não havendo evidências da redução de erros com MVC.

Para a terceira pergunta, “O desenvolvimento utilizando ASP.NET MVC pode reduzir o número de linhas de código no processo de desenvolvimento?” – houve rejeição da hipótese nula, todavia, ficou evidenciado que o desenvolvimento em *WebForms* gera um menor número de linhas que o MVC.

Em resumo, encontramos evidências que *WebForms* leva vantagem no quesito número de linhas e o MVC se sobressai no quesito tempo. Quanto ao quesito quantidade de erros, não foi possível verificar o melhor.

Como sugestão de trabalho futuro, indicamos realizar um novo experimento, nos mesmos moldes, incluindo outros requisitos funcionais, para analisar o comportamento dos componentes de acesso a banco nos dois modelos.

REFERÊNCIAS

BASILI, V.; WEISS, D. A Methodology for Collecting Valid Software Engineering Data. **IEEE Transactions on Software Engineering**, v.10, n.3, p.728-738, november 1984.
BASILI, R. **Software Modeling and Measurement: The Goal Question Metric Paradigm**, Computer

Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland: College Park, Md., Sept., 1992.

DAWEINI; LUGAO. Based on the MVC model of teachers' workload management system development, Computer Science and Information Processing (CSIP). **International Conference on**, p.1392-1395, 24-26 aug. 2012

FANGYUAN, Zheng; HAO, Hu; JIAN, Lu. Applying a Component Behavior Model to MVC Pattern, Young Computer Scientists, 2008. ICYCS 2008. **The 9th International Conference for**, p.1106-1111, 18-21, nov. 2008

GENS, F; LEVITAS, D; SEGAL, R. **Closing the “Consumerisation gap”**. IDC, 2011.

GU MING-XIA; TANG, Keming. Comparative analysis of WebForms MVC and MVP architecture, Environmental Science and Information Application Technology (ESIAT), **International Conference on**, v.2, p.391-394, 17-18 July 2010.

HENDRICKSON, E.; FOWLER, M. The software engineering of internet software: Guest editors' introduction. **IEEE Softw.**, v.19, n.2, p.23-24, 2002.

KERER, C.; KIRDA, E. Layout, content and logic separation in web engineering. In: **Web Engineering, Software Engineering and Web Application Development**, London, UK: Springer-Verlag, 2001. p.135.

MASOUND, F.A.; HALABI, D.H. ASP.NET and JSP Frameworks in Model View Controller Implementation. **Information and Communication Technologies**, ICTTA '06.2nd, v.2, p.3593-3598, 2006.

ORNE, M.T. **Sobre a psicologia social da experiência psicológica**: com referência particular para exigir características e suas implicações, 1962.
ROSA, J.; SILVA, H.; MATIAS, R. A web-based framework using a Model-View-Controller

architecture for human motion analysis, Bioengineering (ENBENG), IEEE **4th Portuguese Meeting on**, p.1-2; 26-28, feb. 2015

SELF, D.M.; CARRILLO, M.; DEL ROCIO BOONE, M. A Database and Web Application Based on MVC Architecture, Electronics, Communications and Computers, 2006. **CONIELECOMP 2006. 16th International Conference on**, p.48-48, 27-01 feb. 2006

SHULL, F.; CARVER, J.; TRAVASSOS, G.H. An empirical methodology for introducing software processes. **SIGSOFT Softw.Eng. Notes**, v.26, n.5, p.288-296, 2001.

SPSS, IBM SOFTWARE. RETRIEVED AUGUST 5, 2015 FROM [HTTP://GOO.GL/FZEMB4](http://goo.gl/FZEMB4).

TRAVASSOS, G. H. **Introdução à engenharia de software experimental**. [S.l.], 2002.

XIAOYUE, Wang; BINXU; RUIGU, The Application of Code Reuse Technology Based on the MVC

Framework, Computer Sciences and Applications (CSA), **International Conference on**, p.534,537, 14-15 dec. 2013

YONGLEI, Tao, Component- vs. application-level MVC architecture, **Frontiers. Education, 2002**. FIE 2002. 32nd Annual, v.1, p.T2G-7,T2G-10, 2002

WANG, H.; YANG, J. Research and application of web development based on ASP.NET 2.0+Ajax. **Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on**, p.857-860, 3-5 june 2008.

WOHLIN, C. *et al.* **Experimentation in Software Engineering**: An introduction. USA: Kluwer Academic Publishers. 2000.

ZHUOYI, Chen; LIMIN, Jia; WEIHUA, Zheng, Research and development of the long distance coach management system based on ASP.net technology. **Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on**, p.1992-1995, 21-23 april 2012

Recebido em: 11 de Julho de 2017
Avaliado em: 13 de Julho de 2017
Aceito em: 14 de Agosto de 2017

1 Meste em Ciência da Computação pela Universidade Federal de Sergipe (UFS), pós-graduação em Sistemas de Informação para Web (2004) e graduação em Ciência da Computação pela Universidade Tiradentes (2002). Atualmente é professor assistente III da Universidade Tiradentes(SE) e Analista de Sistemas SR do departamento de informática da Universidade Tiradentes(SE). Tem experiência na área de Ciência da Computação, com ênfase em Programação para Web. Tem experiência em implantação e manutenção da plataforma ERP Protheus. E-Mail: giga.anderson@gmail.com

