colloquium exactarum

O USO DE WEB SERVICES PARA INTEGRAÇÃO DE SISTEMAS ERP COM PLATAFORMA E-COMMERCE MAGENTO

THE USE WEBSERVICES FOR INTEGRATION WITH ERP SYSTEMS PLATAFORM E-COMMERCE MAGENTO

Sirley Maykelle Pacito da Silva, Francisco Assis da Silva, Francisco Virginio Maracci, Mário Augusto Pazoti

Faculdade de Informática – FIPP, Universidade do Oeste Paulista – UNOESTE
e-mail: smaykelle@hotmail.com, {chico, francisco, mario}@unoeste.br

RESUMO – Com a grande evolução tecnológica englobando diversos setores empresariais, comerciais e industriais, houve o desenvolvimento constante de diferentes aplicações nas mais diversas plataformas, gerando a necessidade de interligar esses sistemas. O uso do conceito da tecnologia de Web Service possibilita agilizar os processos com eficiência e de forma segura, reduzindo significativamente o tempo de trabalho manual. Este trabalho baseia-se na integração de dois sistemas de plataformas diferentes através de um estudo de caso real. A meta é reduzir o tempo de lançamento de pedidos de produtos e cadastro de novos clientes com o desenvolvimento do Web Service. Este consumirá os dados de uma plataforma e-commerce Magento e posteriormente incluirá os mesmos em um banco de dados Oracle de um ERP, comprovando os objetivos de minimizar os recursos.
Palavras-chave: web service; Magento; ERP; Oracle.

ABSTRACT - With the great technological developments encompassing several business, commercial and industrial sectors, there was the constant development of different applications in various platforms, creating the need to interconnect these systems. The use of the Web Service technology concept makes it possible to streamline these processes efficiently and safely, where it significantly reduces manual working time. This work is based on the integration of two different platform systems through a real case study. The goal is to reduce the time to launch product orders and register new customers with the development of Web Service. This will consume the data from a Magento e-commerce platform and will later include the same in an Oracle database of an ERP, proving the goals of minimizing the resources.
Keywords: web service; Magento; ERP; Oracle.

Recebido em: 27/07/2016
Revisado em: 23/02/2017
Aprovado em: 25/04/2017

1. INTRODUÇÃO

A necessidade da integração entre aplicações de diferentes plataformas, a customização de processos e otimização de recursos e a redução de atrasos e erros são alguns exemplos de privações encontradas em algumas empresas atualmente. Com o propósito de sanar essas questões, a tecnologia de Web Service foi criada, que permite que diferentes tipos de aplicações possam interagir, modularizar serviços e consumir informações através de uma interface baseada em tecnologias com especificação XML[1] e HTTP[2] (MENÉNDEZ, 2002).

Segundo Costa (2002), um Web Service propõe a exposição das transações e das regras de negócios por meio de protocolos que podem ser acessados e entendidos em qualquer linguagem de programação, em qualquer sistema operacional, rodando em qualquer dispositivo. Em um Web Service é criado um conjunto de funções empacotadas e distribuídas em algum lugar da rede, para que outros programas possam usá-las livremente. Desta forma, a criação de Web Services facilita a integração entre sistemas de uma forma simples, que melhora os processos de negócios e torna-os automatizados, o que permite a diminuição de custos e aumento da produtividade. Por isso, essa tecnologia cada vez mais ganha espaço nas empresas.

De acordo com Arruda (2015), muitas empresas como Microsoft, IBM, Oracle, Google, Yahoo, utilizam dessa tecnologia para integrar sistemas legados e heterogêneos de forma rápida e fácil. Para isso, essas empresas criam Web Service, podendo disponibilizá-los para que o resto do mundo possa consumi-los sem se preocupar com quem (que tecnologia) está consumindo. Dessa forma, pode-se integrar aplicações, negócios e componentes.

Para exemplificar o uso de Web Service, pode-se tomar como referência uma loja virtual que para enviar o pedido ao cliente, necessita dos dados de endereço completo. Desta forma, tem-se a opção de armazenar todos os CEPs no banco de dados e mantê-los sempre atualizados ou buscar as informações por faixa de CEP através de um Web Service disponibilizado pelo site dos correios (www.correios.com.br). Busca-se com isso agilizar os processos e evitar possíveis erros. A opção de usar um Web Service neste caso é a melhor, em que o mesmo recebe o valor com o CEP e retorna os dados referente ao endereço para, por

[1] XML: *eXtensibleMarkupLanguage* - é um formato simples de texto para representar informações estruturadas: documentos, dados, configuração, livros, transações, faturas e muito mais. (W3C, 2016)
[2] HTTP: *Hiper Text Markup Language* - Protocolo responsável por transferir hipertextos pela internet (RIOS, 2012).

exemplo, cálculo do frete para envios de encomenda.

Segundo a W3C - *World Wide Web Consortium* (WC3 BRASIL, 2015a, 2015b; WC3, 2015), principal organização de padronização da *World Wide Web*, Web Services referem-se aos projetos de comunicações na Web entre aplicações e baseiam-se em tecnologias como: HTTP, XML, SOAP[3], UDDI[4], WSDL[5].

No ano de 2000, a W3C (*World Wide Web Consortium*) aceitou a submissão do *SimpleObject Access Protocol* (SOAP). Este formato de mensagem baseado em XML estabeleceu uma estrutura de transmissão para comunicação entre aplicações (ou entre serviços) via HTTP. No decorrer do ano seguinte, o W3C publicou a especificação WSDL (*Web Services Description Language*). Uma nova implementação do XML, este padrão forneceu uma linguagem para descrever a interface do Web Service. Posteriormente, suplementada pela especificação UDDI (*Universal Description*, *Discovery and Integration*), que proporcionou um mecanismo padrão para a descoberta dinâmica (*dynamic discovering*) de descrições de serviço, a primeira geração da plataforma de Web Service foi estabelecida (ERL, 2015).

Um cenário típico do uso de Web Service é o envio de uma solicitação para um serviço publicado em uma URL[6] via HTTP, usando o protocolo SOAP. O Web Service irá receber a solicitação e processá-la; no final ele deverá retornar uma resposta também usando SOAP.

A definição da interface de um Web Service é feita através de WSDL. Além disso, existem serviços Web especializados em publicar e descobrir Web Service, chamados de Service Provider. Eles podem ser considerados como as páginas amarelas dos Web Service. Um Service Provider deve implementar o padrão UDDI para a publicação do serviço (MENÉNDEZ, 2002).

O Web Service que foi criado neste trabalho foi usado para acessar a plataforma do CMS (*Custom Management System*) do e-commerce Magento. Os CMS são sistemas de gerenciamento de conteúdo onde o usuário se preocupa com a criação/edição do conteúdo e não com detalhes técnicos por trás do funcionamento do site e a maioria são *open source* (código aberto). Os CMS mais conhecidos e utilizados são: Joomla, WordPress, Wix e Magento (CANALTECH, 2015).

[3] SOAP: *Simple Object Access Protocol* - Protocolo elaborado para facilitar a chamada remota de funções via Internet, permitindo que dois programas se comuniquem de uma maneira tecnicamente muito semelhante à invocação de páginas Web (SANT'ANNA, 2015).

[4] UDDI: *Universal Descriptionand Discovery Information* - tem como papel resolver o problema de descoberta de web services e ajudar na sua publicação. (MENÉNDEZ, 2002).

[5] WSDL: *Web Service Definition Language* - Sistema para a descrição de serviços. Através dela, são descritos os serviços externos, ou interfaces que são oferecidos por uma determinada aplicação, independentemente de sua plataforma ou linguagem de programação (RECKZIEGEL, 2015).

[6] URL: *Uniform Resource Locator* - ou Localizador Padrão de Recursos é um link de uma página de internet ou indicador de um caminho em uma rede (RIOS, 2012).

O Magento Commerce (MAGENTO, 2015a) é a principal plataforma para a inovação de comércio aberto, ou seja, atualmente é a plataforma para e-commerce mais utilizada para efetuar transações em lojas virtuais. O Magento Commerce trabalha lado a lado com os varejistas, através das indústrias B2C[7] (*Business to Commerce*) e B2B[8] (*Business to Business*) para integrar com sucesso experiências de compras digitais e físicas. Nasceu por iniciativa da Varien (empresa que criou o Magento) em 2007 para desenvolvimento de e-commerce no formato *open source*, como *osCommerce* e *Prestashop*, pioneiras neste segmento. Em 2008 a empresa lançou a primeira versão aberta do sistema e deste ano em diante a plataforma está cada vez mais firme no mercado em função de seus recursos tecnológicos e soluções avançadas para o comércio eletrônico. Posteriormente em 2011 o grupo eBay adquiriu parte das ações da Magento e em 2014 a empresa *aheadWorks*, especializada no desenvolvimento de extensões para a plataforma, divulgou na sua pesquisa anual que pelo terceiro ano consecutivo o Magento se destaca como sistema de e-commerce *open source* mais usado no mundo, com uma participação de 33,8% no universo analisado (VALLE, 2015).

Existem duas plataformas Magento (MAGENTO, 2015a), sendo Magento Community Edition, para pequenas e médias empresas, livre para download, código aberto altamente personalizável e o Magento Enterprise Edition, adaptado às maiores empresas com alto nível de vendas *on-line* (AHEADWORKS, 2016). Uma das principais características da plataforma Magento é sua estrutura composta por diversos módulos independentes que reunidos compõem o sistema. Além disso, é construído em cima de um *framework* PHP que pode ser explorado para se desenvolver aplicações e adaptações dinâmicas, conforme a necessidade (BLING, 2016).

Os módulos são conhecidos como extensões Magento ou *plug-ins*, sendo possível incorporar novas funções à loja virtual e com isso torná-las mais interessantes para quem a visita e também mais funcional para quem administra (VALLE, 2015).

O Magento API[9] SOAP oferece a capacidade de gerenciamento da lojas de comércio eletrônico, fornecendo chamadas para trabalhar com recursos, tais como

[7] B2C: *Business to Commerce* - Transação comercial entre empresa (indústria, distribuidor ou revenda) e consumidor final através de uma plataforma de E-Commerce. A Natureza dessa operação tende a ser apenas de consumo (NISSAN, 2014).

[8] B2B: *Business to Business* - Transações comerciais entre empresas. Em outras palavras, é um ambiente (Plataforma de E-Commerce) onde uma empresa (indústria, distribuidor, importador ou revenda) comercializa seus produtos para outras empresas. A natureza dessa operação pode ser revenda, transformação ou consumo (NISSAN, 2014).

[9] API: *Application Programming Interface* - exposição de uma série de ferramentas, métodos de programação e protocolos, com o objetivo de facilitar a programação de uma aplicação (PELOI, 2016).

clientes, categorias, produtos e pedidos de vendas. Ele também permite gerenciar carrinhos de compras e estoque (SAMPAIO, 2012).

Com o desenvolvimento do Web Service, a integração entre o Magento e o ERP tem o tempo de execução do processo de lançamento de pedidos e cadastro de novos clientes reduzido. O funcionário responsável pelo processo pode alocar seu tempo de trabalho em outras funções do departamento. Além disso, reduz a probabilidade de inserção de dados incorretos, como por exemplo, condições de pagamento do pedido (prazo), nome do cliente, troca do código, valores e quantidade dos produtos. Estes são alterados, após serem revistos pelo responsável do setor financeiro, onde é necessário refazer todo o processo, e então é liberado para aprovação. Esses procedimentos exigem mais alguns minutos ou até mesmo horas para o retrabalho dos setores envolvidos.

As demais seções deste trabalho estão organizadas da seguinte maneira: na Seção 2 são apresentados os trabalhos relacionados; na Seção 3 é apresentada a metodologia deste trabalho; na Seção 4 são apresentados os experimentos e resultados obtidos e, por fim, na Seção 5 são apresentadas as conclusões e considerações finais do trabalho.

2. TRABALHOS RELACIONADOS

Nascimento (2013) desenvolveu um Web Service para promover o intercâmbio de dados entre dispositivos móveis e um sistema ERP. O autor empregou conceitos de serialização de objetos, protocolo de transporte e servidores de aplicação. A solução promoveu a comunicação entre um sistema de vendas desenvolvido para plataforma móvel, mais especificamente o iPad, e um sistema ERP. Devido à variedade de soluções encontradas para o desenvolvimento do Web Service, o autor investigou as principais tecnologias, SOAP e [10]RESTful. Com base na pesquisa realizada foi constatado um maior apelo pelo paradigma RESTful, devido a sua simplicidade e objetividade se comparado ao SOAP, o que para um dispositivo móvel com recursos limitados, tanto de processamento quanto de conexão, pesou na escolha. O trabalho foi focado em requisitos de segurança, interoperabilidade, baixa suscetibilidade à erros e principalmente, o desempenho. A interoperabilidade, ou capacidade de integração de recursos computacionais heterogêneos foi alcançada pelo simples fato da adoção da arquitetura de serviços Web na constituição do projeto, já que esta é uma

[10] RESTful - *Representational State Transfer* - são serviços construídos com o estilo de arquitetura RESTful. A construção de Web Services com a abordagem RESTful está surgindo como uma alternativa popular ao uso de tecnologias baseadas em SOAP para implantação de serviços na Internet, por ser mais leve e ter a capacidade de transmitir dados diretamente via HTTP (NETBEANS, 2016).

das características da arquitetura, conseguida pelo método de serialização de objetos.

Santos (2003) desenvolveu um modelo de Web Service na plataforma .NET em conjunto com a tecnologia para dispositivos computacionais móveis, denominados PDA's[11]. Essa utilização conjunta de tecnologias emergentes foi exemplificada através de um aplicativo para executar em PDA's que utilizam a plataforma operacional Pocket PC. A função do aplicativo desenvolvido é de apresentar notícias localizadas em um banco de dados remoto, que são acessadas por meio dos métodos do Web Service desenvolvido. Para que as comunicações fossem usadas, as funcionalidades oferecidas pelo Web Service foram feitas por meio de documentos WSDL. Para a comunicação efetiva, foram utilizados alguns protocolos: HTTP-POST, HTTP-GET e SOAP, que permite a representação de dados complexos. Tanto estes protocolos quanto os documentos WSDL são baseados em XML, sendo este aspecto que possibilita a interoperabilidade, pois através destes artefatos é que as mensagens são transportadas.

Galhardo e Galhardo (2008) utilizaram a tecnologia de Web Service nas integrações entre sistemas de operação de redes de distribuição (baseados em geoprocessamento) com sistemas legados. A integração entre os sistemas, através de Web Service, ocorreu entre o sistema de operação de uma companhia de distribuição de energia e um sistema que controla a comunicação com as unidades móveis que executam os serviços de manutenção corretiva da rede. O sistema de operação, baseado em geoprocessamento, é composto por um conjunto de funcionalidades que permite efetuar todo o ciclo de atendimento de um serviço de manutenção corretiva da rede. Através do geoprocessamento, todos os pontos da rede são georreferenciados e as equipes de atendimento são monitoradas por GPS. O sistema GCM (Gerenciador Central de Mensagens) consiste basicamente de um site onde todas as mensagens produzidas pelos dois sistemas são armazenadas. A troca de informações, entre os sistemas, se dá através de Web Service que ciclicamente acessam as mensagens no site.

3. METODOLOGIA

A solução criada neste trabalho, para realizar a integração de dois sistemas de plataformas diferentes para reduzir o tempo de lançamento de pedidos de produtos e cadastro de novos clientes, foi o desenvolvimento de um Web Service implementado na plataforma .NET com o ambiente de programação Visual Studio 2012. Para acessar o Web Service

[11] PDA´s - *Personal Digital Assistant* - em português Assistente Digital Pessoal, são dispositivos computacionais móveis (SANTOS, 2003).

desenvolvido foi implementado um aplicativo também na plataforma .NET, que por meio de um formulário Windows, faz a requisição ao Web Service. O Web Service consome os dados dos pedidos de produtos e clientes do sistema da plataforma e-commerce Magento, que está hospedado em um VPS (*Virtual Private Server*) da empresa Hospedagem Estavel, gerenciado pela empresa OxigenWeb Marketing Digital, e os insere no banco de dados Oracle de um ERP denominado CS, desenvolvido pela empresa Compusoftware. O ERP está instalado em um Servidor Master na empresa em que foi realizado este trabalho (do ramo Agropecuário), com as seguintes configurações: HP ProLiant ML350 G5, com Sistema Operacional CentOS release 5.8, Processador Intel (R) Xeon (R) CPU E5620 @ 2.40GHz, HD com 4 discos SATA 750GB (raid 5) e memória RAM de 8GB.

A API do Magento suporta SOAP e XML RPC, em que SOAP é o protocolo padrão. Para se conectar a um Web Service Magento SOAP, pode-se carregar o WSDL para o seu cliente SOAP a partir de qualquer uma destas URLs:

http://magentohost/api/?wsdl

http://magentohost/api/soap/?wsdl

http://magentohost/api/v2_soap?wsdl=1

Na linguagem de programação C# a chamada a API SOAP v2 pode ser feita através do exemplo a seguir:

```
MagentoServicemservice = new MagentoService();
String mlogin = mservice.login ("login", "password");
```

Através dessas chamadas, é possível consumir os dados dos pedidos de produtos dos clientes do e-commerce da plataforma Magento pelo Web Service criado neste trabalho. Desta forma, esses dados são validados e inseridos automaticamente no banco de dados do ERP nas respectivas tabelas de clientes e pedidos, e posteriormente são aprovados pelo setor financeiro da empresa. Esse processo, é realizado automaticamente pelo Web Service e reduz o tempo gasto com impressão, lançamento de pedidos e produtos, bem como cadastro de novos clientes, que é realizado manualmente, além de estar sujeito a erros com a necessidade de muitas horas para ser realizado.

O acesso às informações do banco de dados ocorre por meio de módulos e métodos da plataforma e-commerce Magento. No módulo "Sales" estão todos os dados referente ao pedido do cliente. Nesse módulo os dados são consumidos através das Entidades do Magento. Na entidade "salesOrderListEntity" (Figura 1), constam os Dados do Pedido, como informações sobre a número do pedido, código do cliente, data de criação do pedido, valor, status, tipo de frete e endereços de cobrança, e entrega do pedido e o código da fatura.

salesOrderListEntity	
Referência	**Nome do campo**
Código do Pedido	increment_id
Data do Pedido	created_at
Código do Cliente	customer_id
Valor do Frete	shipping_amount
Subtotal do Pedido	subtotal
Total do Pedido	base_grand_total
Status do Pedido	status
Código da Fatura	order_id
Forma de envio (Correios - Sedex/ PAC)	shipping_description
Endereço de Entrega	shipping_address
Endereço de Cobrança	billing_address
Telefone	telephone

Figura 1. Dados do Pedido.

Na entidade "salesOrderPaymentEntity", estão as informações sobre o tipo de pagamento, quantidade de parcelas, e se o tipo de pagamento for cartão.

salesOrderPaymentEntity	
Referência	**Nome do campo**
Tipo de Pagamento	method
Qtde. de Parcelas	additional_data

Figura 2. Tipo de Pagamento.

Na entidade "salesOrderItemEntity", as informações são dos produtos dos pedidos, tais como código, quantidade, valor, total e SKU.

No módulo "Customer" é possível obter as informações cadastrais do cliente como código, email, nome, sobrenome, CPF ou CNPJ, RG ou Inscrição Estadual, por meio da entidade "customerCustomerEntity" (Figura 4).

salesOrderItemEntity	
Referência	**Nome do campo**
Código da fatura	order_id
Código do produto	product_id
Qtde. do produto	qty_invoiced
Valor do produto	base_price
Total do produto	base_row_invoiced
SKU	sku

Figura 3. Produtos do Pedido.

customerCustomerEntity	
Referência	**Nome do campo**
Código do cliente	customer_id
Email	email
Nome	firstname
Sobrenome	lastname
CPF ou CNPJ	cpfcnpj
RG	rg
Inscrição Estadual	ie

Figura 4. Informações do Cliente.

Para que ocorra a importação dessas informações, é necessário que o campo "status" da entidade "salesOrderItemEntity" esteja como "processing" (processando). Na Figura 5 no Painel de Controle do Magento, é possível visualizar os pedidos que estão com status "processing" no painel do Magento. Quando o status do pedido consta como "pending" (pendente), a forma de pagamento é por boleto ou depósito

bancário. Neste caso, o processo é feito manualmente antes de ser importado.



Figura 5. Painel de Vendas do Magento.

No painel da plataforma do Magento, o responsável pelo lançamento de pedidos, confere a forma de pagamento do pedido e verifica junto ao setor financeiro da empesa se o boleto foi pago ou o depósito foi efetuado em conta. Após esta confirmação, no painel do Magento, o status é alterado manualmente para "processing", possibilitando realizar a integração.

Para iniciar o processo de integração dos sistemas, foi criada a função "buscarPorCodigo()", conforme Figura 6, que verifica na tabela "CSINTEGRACAO.PEDIDOS" do ERP CS se o código do pedido já existe, através de um controle SqlDataSource (controle para recuperar dados do Banco).

```
public bool buscarPorCodigo()
{
   banco.abrirConexao();

   banco.SqlCmd.CommandText =
"select * from pedidos where
ped_cod_ant = @COD";

   banco.SqlCmd.Parameters.Add
WithValue("COD", _ped_cod_ant);

   DataTable dt = new DataTable();

   dt.Load(banco.SqlCmd.Execute
Reader());

   banco.fecharConexao();

   if (dt.Rows.Count > 0)
      return true; //existe o pedido
           else
      return false; //não existe o pedido
}
```

Figura 6. Função buscarPorCodigo.

Caso o pedido não tenha sido importado, os dados são inseridos nas tabelas "CSINTEGRACAO.PEDIDOS", "CSINTEGRACAO.PARCEIRONEGOCIO" e "CSINTEGRACAO.PRODUTOS".

A Figura 7 mostra a tabela "CSINTEGRACAO.PEDIDOS", onde serão inseridas as informações do pedido.

CSINTEGRACAO.PEDIDOS	
ped_cod	number
cli_cod	number
ped_subtotal	number
ped_frete	number
ped_total	number
ped_tppagamento	number
ped_parcela	number
ped_envio	number
ped_endentrega	varchar(100)
ped_endcobranca	varchar(100)
ped_data	date

Figura 7. Tabela de pedidos do ERP CS.

A Figura 8 mostra a tabela "CSINTEGRACAO.PARCEIRONEGOCIO", onde serão inseridas as informações do cliente.

CSINTEGRACAO.PARCEIRONEGOCIO	
ped_codigo	number
cli_codigo	number
cli_nome	varchar (50)
cli_cpf	varchar(50)
cli_rg	varchar(20)
cli_cnpj	varchar(14)
cli_ie	varchar(15)
cli_email	varchar(50)

Figura 8. Tabela de clientes ERP CS.

Por fim, a Figura 9 mostra a tabela “CSINTEGRACAO.PRODUTOS”, que terá os dados dos produtos inseridos.

CSINTEGRACAO.PRODUTOS	
ped_codigo	number
prod_codigo	number
prod_qtde	number
prod_valor	number
prod_vrtotal	number
prod_sku	number

Figura 9. Tabela de Produtos do ERP CS.

Após a inserção nessas tabelas de integração, o processo passa a ser executado pelo ERP CS, que faz a verificação do cadastro de clientes e produtos, comparando as informações das tabelas de integração com as informações das tabelas já existentes no ERP CS. Após essa verificação, as mesmas são inseridas nas respectivas tabelas de clientes e produtos.

4. EXPERIMENTOS E RESULTADOS

Para realização dos experimentos, foi criado um aplicativo com uma única tela que acessa o Web Service para integrar os pedidos da plataforma e-commerce Magento com o ERP CS, como mostra a Figura 10.

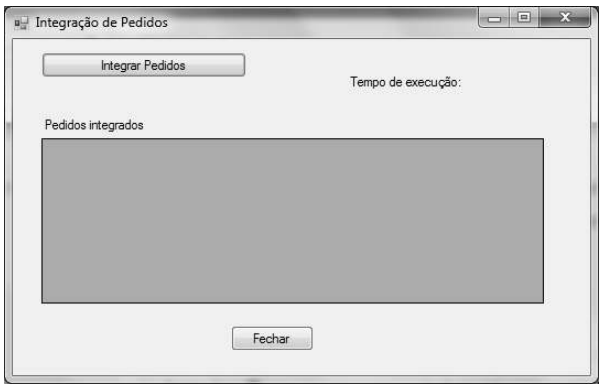

Figura 10. Inicial do aplicativo.

Após consumir os dados da plataforma e-commerce Magento, nessa mesma tela, são exibidos os pedidos que foram inseridos nas tabelas do ERP CS, conforme mostra a Figura 11. No caso será considerado o pedido “100072671” para explicar as Figuras 11, 13, 14 e 15.

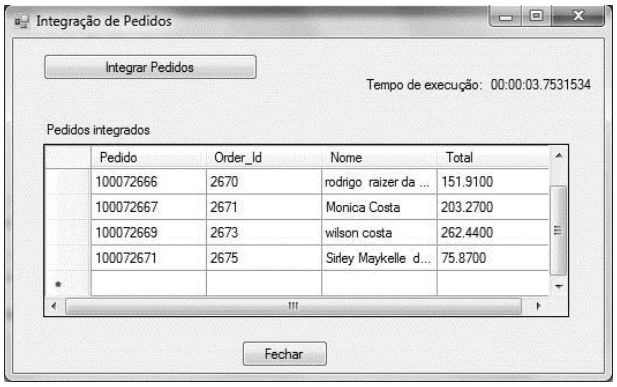

Figura 11. Tela do aplicativo após consumir dados da plataforma Magento.

Caso não existam pedidos com o status "processing" ou se os pedidos com esse status já foram inseridos nas tabelas do ERP CS (Figura 7, 8 e 9), aparece uma caixa de mensagem informando que não existem pedidos a integrar (Figura 12).

Na Figura 13 é possível visualizar um modelo XML da entidade "customerCustomerEntity" onde constam os dado do cliente (em negrito) do pedido “100072671” que serão inseridos na tabela "CSINTEGRACAO.PARCEIRONEGOCIO" do ERP CS, que é mostrada na Figura 8.

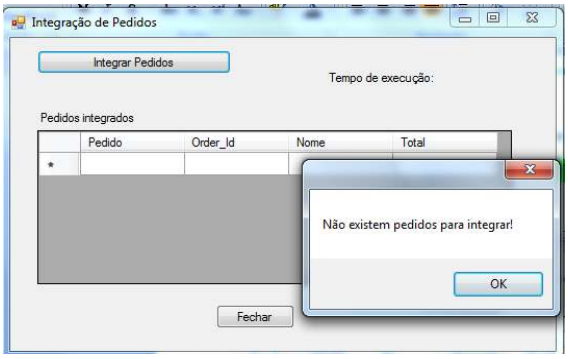

Figura 12. Caixa de Mensagem para informar que não existem pedidos para integrar.

```
1 <customerCustomerEntity
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
2 <customer_id>6</customer_id>
3 <customer_idSpecified>true</customer_idSpeci
fied>
4                          <created_at>2007-11-08
11:05:58</created_at>
5                          <updated_at>2016-07-07
20:49:50</updated_at>
6  <store_id>1</store_id>
7 <store_idSpecified>true</store_idSpecified>
8  <website_id>1</website_id>
9 <website_idSpecified>true</website_idSpecifi
ed>
10  <created_in>Portugues</created_in>
11 <email>informatica@arenales.com.br</email>
12 <firstname>Sirley Maykelle </firstname>
13 <lastname>da Silva</lastname>
14 <group_id>1</group_id>
15 <group_idSpecified>true</group_idSpecified>
16 <dob>1982-09-04 00:00:00</dob>
17 <taxvat>30728520818</taxvat>
<confirmationSpecified>false</confirmationSpec
ified>
18 <password_hash>7baa3c1b23d0a7a852ff5d9bc470
96e6:bH8it52d5PL73BJb0mu6CafVDVQuW6iC</passwor
d_hash>
19 <cpfcnpj>307.285.208-18</cpfcnpj>
20 <rg>348034714</rg>
21 'WEcommerce_Arenales.vshost.exe'    (Managed
(v4.0.30319)):                          Loaded
22 'C:\Windows\Microsoft.Net\assembly\GAC_32\S
ystem.Transactions\v4.0_4.0.0.0__b77a5c561934e
089\System.Transactions.dll', Skipped loading
symbols. Module is optimized and the debugger
option 'Just My Code' is enabled.
23 'WEcommerce_Arenales.vshost.exe'    (Managed
```

```
(v4.0.30319)):                                  Loaded
24 'C:\Windows\Microsoft.Net\assembly\GAC_32\S
ystem.EnterpriseServices\v4.0_4.0.0.0__b03f5f7
f11d50a3a\System.EnterpriseServices.Wrapper.dl
l', Skipped loading symbols. Module is
optimized and the debugger option 'Just My
Code' is enabled.
25 <ie>348034714</ie>
26 </customerCustomerEntity>
```

Figura 13. XML da entidade customerCustomerEntity.

Na linha 1 da Figura 13 está o nome da entidade "customerCustomerEntity". Na linha 2, tem-se ao código do cliente e na linha 11 o e-mail. Nas linhas 12 e 13 estão o nome do cliente. Na linha 16, está a data de nascimento. Na linha 19, está o número do CPF ou CNPJ e nas linhas 20 e 25 o número do RG e inscrição estadual, respectivamente.

A Figura 14 mostra a tela de pedido já preenchida com as informações que foram integradas pelo Web Service, conforme o pedido "100072671". A situação do pedido consta como "PENDENTE", pois já foi finalizado e está aguardando aprovação para ser liberado ao faturamento.

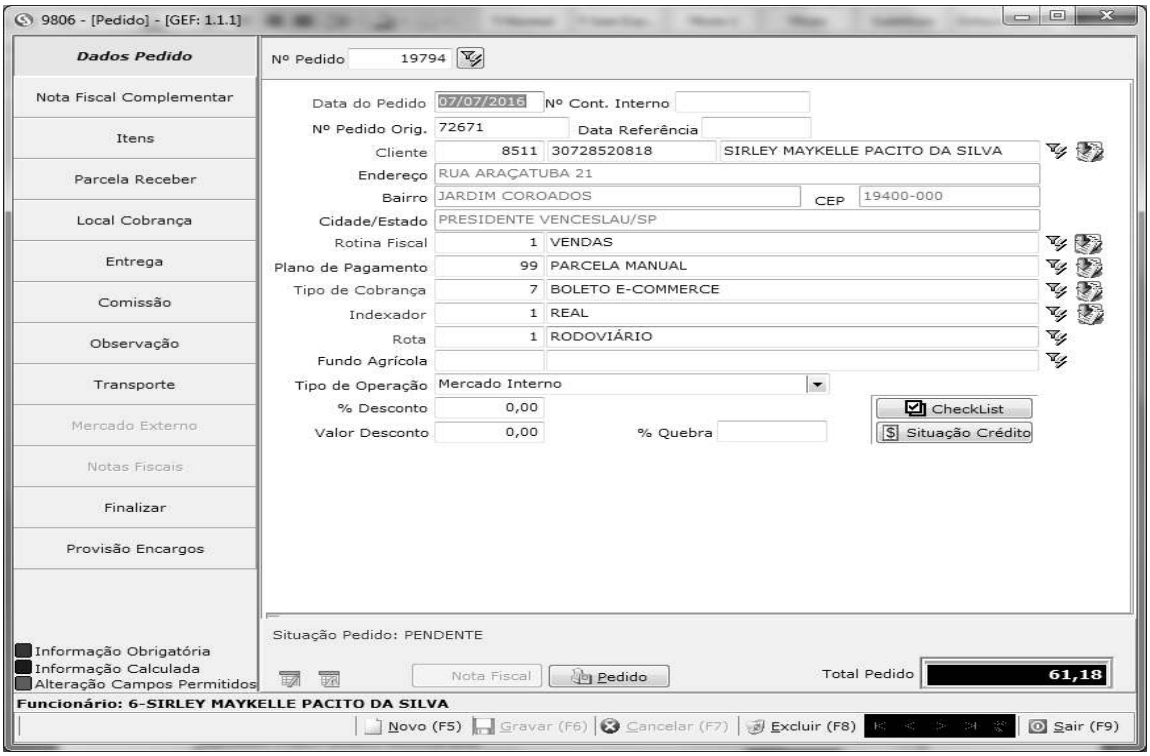

Figura 14. Tela de Pedidos do ERP CS.

No Total do Pedido, consta apenas o valor do produto, sem o valor do frete. O valor do frete é inserido na opção "TRANSPORTE", e finalizado o pedido, conforme mostra a Figura 15.

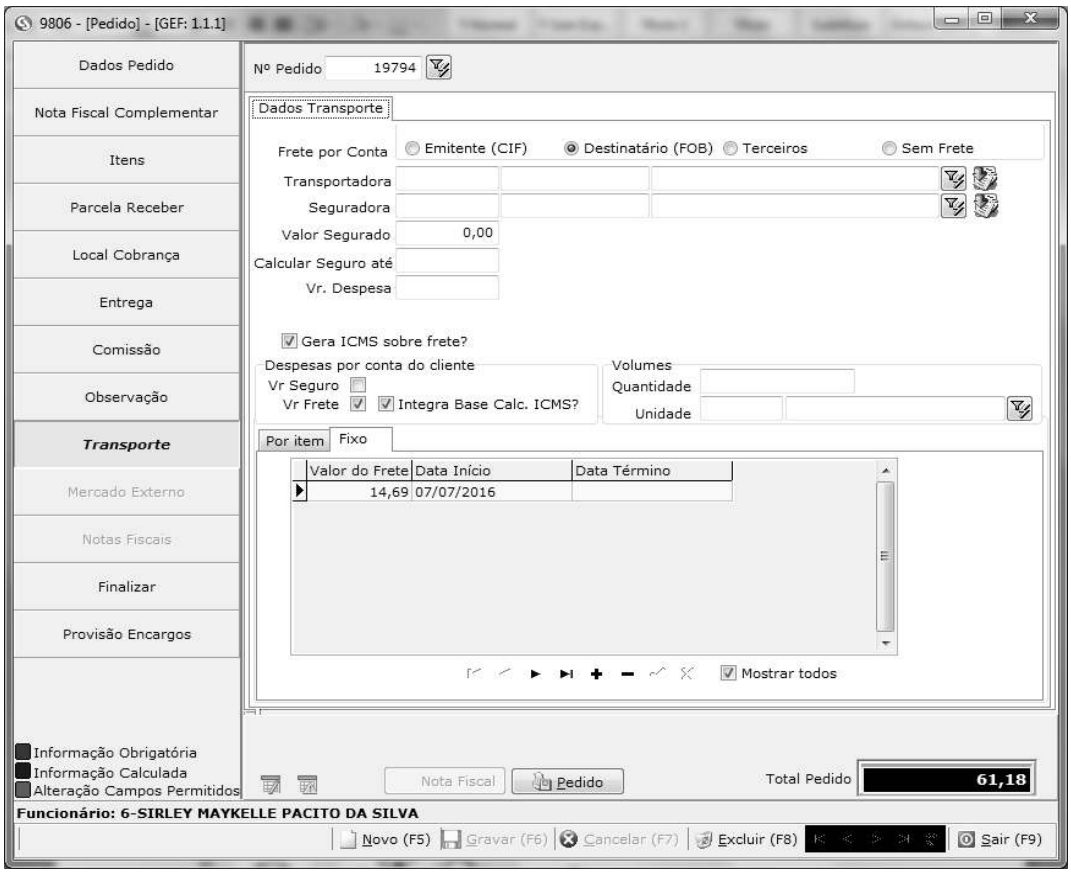

Figura 15. Tela de Pedidos – Transporte.

Para analisar o tempo de execução de todo o processo de integração foi usada a classe "StopWatch" do componente "System.Diagnostics" da biblioteca de classes .NET Framework no aplicativo desenvolvido, onde ao requisitar o acesso ao Web Service, inicia-se a contagem do tempo de execução, em que a mesma é finalizada ao gravar as informações no banco de dados Oracle do ERP CS. Para o processo manual, foi usado um cronômetro que inicia com o lançamento das informações na tela de Pedidos do ERP CS.

A análise foi realizada em dois tipos de lançamentos: para novo cliente, em que foi necessário cadastrar novo cliente e para cliente já cadastrado, no lançamento de apenas um pedido. Conforme o gráfico "Tempo de Execução" da Figura 16, é possível fazer uma análise e constatar que o tempo de execução manual é muito superior ao tempo de execução do Web Service. Para o primeiro caso, de novo cliente, o tempo de execução manual é de aproximadamente 14 minutos e 52 segundos e para cliente já cadastrado, é de 7 minutos e 32 segundos. Em relação a estes processos, o Web Service executou o processo para novo cliente em 15 segundos e para cliente já cadastrado em 7 segundos. Comparando as duas formas de execução dos processos, pode-se verificar que o Web Service é 64,57 vezes mais rápido para

clientes já cadastrados e 59,46 vezes mais rápido para novos clientes.

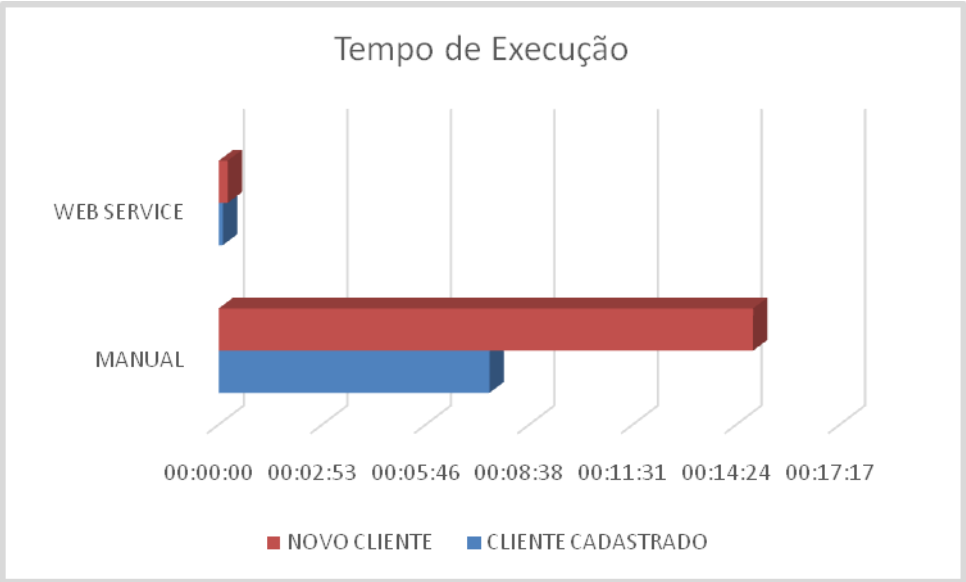

Figura 16. Tempo de Execução.

Os testes foram realizados em um Work Station HP Compaq 8200 Elite SFF PC, com processador Intel Core i7-2600 CPU 3.40 GHz, 8 GB de Memória, com Sistema Operacional Windows 7 Professional.

5. CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi mostrar que, com o desenvolvimento do Web Service é possível integrar sistemas de plataformas diferentes, reduzindo o tempo gasto com o lançamento de pedidos e cadastros de clientes. O Web Service demonstrou rapidez e eficiência em relação ao processo manual, comparando o lançamento de apenas um pedido, visto que, na empresa onde foi realizado o experimento, o número de pedidos diários é superior a dez. Desta forma, conclui-se que o objetivo foi alcançado, pois o processo manual executado em média de uma hora e cinquenta minutos, foi reduzido para uma média de onze segundos, além do que, quando digitados manualmente, pode haver inconsistência de dados, com possiblidade de erros. Com o Web Service, todo o processo é seguro e consistente.

REFERÊNCIAS

AHEADWORKS. **What is Magento?** 2016. Disponível em: <http://ecommerce.aheadworks.com/>. Acesso em: 12 jan. 2016.

ARRUDA, F. **Artigo WebMobile 15 - Integrando tecnologias heterogêneas com Web**. 2015. Disponível em: <http://www.devmedia.com.br/artigo-webmobile-15-integrando-tecnologias-heterogeneas-com-web-services/8054> Acesso em: 18 nov. 2015.

BLING. **Magento para desenvolvedores**. 2016. Disponível em <http://manuais.bling.com.br/manual/?item

=magento-para-desenvolvedores>. Acesso em: 12 jan. 2016.

CANALTECH. **O que é CMS (Content Management System)?** 2015. Disponível em: <http://corporate.canaltech.com.br/o-que-e/programacao/o-que-e-cms-content-management-system/>. Acesso em: 07 dez. 2015.

COSTA, G. O Modelo de Web Services — como desenvolver aplicações em uma nova arquitetura de software. **Promon Business & Technology Review Series**, n.4, 2002.

ERL, T. Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI - Parte1. 2015. Disponível em: <http://www.devmedia.com.br/introducao-as-tecnologias-web-services-soa-soap-wsdl-e-uddi-parte1/2873>. Acesso em: 18 nov. 2015.

GALHARDO, M. R.; GALHARDO, R. R. Utilização de Web Services como meio de integração de Sistemas. 2008. In: SEMINÁRIO NACIONAL DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA, 18. **Anais...** Olinda – PE, 2008.

MAGENTO. **About Magento**. 2015a. Disponível em: <http://magento.com/company/about-us>. Acesso em: 07 dez. 2015.

MAGENTO. **Introduction to the Magento 1.x SOAP API**. 2015b. Disponível em: <http://devdocs.magento.com/guides/m1x/api/soap/introduction.html>. Acesso em: 07 dez. 2015.

MENÉNDEZ, A. I. M. **Uma ferramenta de apoio ao desenvolvimento de Web Services**. 2002. 97p. Dissertação (Mestrado em Informática) – Universidade Federal de Campina Grande, Campina Grande, 2002.

NASCIMENTO, M. E. H. **Uma arquitetura de serviços web como meio de intercâmbio de dados entre sistemas heterogêneos.** 2013. 67p. Trabalho (Conclusão de Curso) - Universidade Federal de Santa Catarina, Curso Tecnologias da Informação e Comunicação, 2013.

NETBEANS. **Introdução aos Web Services RESTful.** Dez. 2016. Disponível em: <https://netbeans.org/kb/docs/websvc/rest_pt_BR.html> Acesso em: 12 dez. 2016.

NISSAN, M. **Qual a diferença entre B2B e B2C?** Jan. 2014. Disponível em: <http://ecommercenews.com.br/artigos/cases/qual-e-a-diferenca-entre-b2b-e-b2c>. Acesso em: 09 mar. 2016.

PELOI, R. **O que são API's – Part 1 : Introdução**. 2016. Disponível em <http://sensedia.com/blog/apis/o-que-sao-apis-parte-1-introducao/>. Acesso em: 13 dez. 2016.

RECKZIEGEL, M. **Descrevendo um Web Service – WSDL**. 2015. Disponível em: <http://imasters.com.br/artigo/4422/web-services/descrevendo-um-web-service-wsdl/> Acesso em: 07 dez. 2015.

RIOS, R. O. **Protocolos e Serviços de Rede.** Colatina–ES, 2012. Curso Técnico em Informática. Disponível em: <http://ead.ifap.edu.br/netsys/public/livros/LIVRO%20MANUTEN%C3%87%C3%83O/Modulo%20III/Protocolo_servicos_redes.pdf>. Acesso em: 13 dez. 2016.

SAMPAIO, T. **Entendo as APIs do Magento – SOAP – Introdução.** Out. 2012. Disponível em: <http://tiagosampaio.com/entendendo-as-apis-do-magento-soap-introducao/>. Acesso em: 13 jan. 2016.

SANT'ANNA, M. **SOAP e Webservices**. 2015. Disponível em: <http://www.linhadecodigo.com.br/artigo/38/soap-e-webservices.aspx#ixzz3rrgXSAyA>. Acesso em: 18 nov. 2015.

SANTOS, M. S. **Utilização de Web Services na plataforma .NET para a criação de um aplicativo visualizador de notícias para dispositivos móveis**. 2003. 89p. Monografia (Trabalho de Conclusão) – Centro Universitário Luterano de Palmas, 2003. Disponível em: <http://www.significados.com.br/>. Acesso em: 18 nov. 2015.

VALLE, A. **O que é Magento – Plataforma de e-commerce.** 2015. Disponível em: <http://www.guiadeecommerce.com.br/o-que-e-magento/> Acesso em: 07 dez. 2015.

W3C BRASIL. **Materiais, cursos e publicações**. 2015a. Disponível em: <http://www.w3c.br/Materiais/>. Acesso em: 18 nov. 2015.

W3C BRASIL. **Padrões.** 2015b. Disponível em: <http://www.w3c.br/Padroes> Acesso em: 18 nov. 2015.

W3C. **Web of Services**. 2015. Disponível em: <http://www.w3.org/standards/webofservices/>. Acesso em: 18 nov. 2015.

W3C. **XML Essentials.** 2016. Disponível em: <https://www.w3.org/standards/xml/core>. Acesso em: 12 dez. 2016.