UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE PÓS-GRADUAÇÃO EM BANCO DE DADOS

EDUARDO PACHECO

DESENVOLVENDO WEB SERVICES COM ORACLE EM PL SQL: VANTAGENS E DESVANTAGENS

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2017

EDUARDO PACHECO

DESENVOLVENDO WEB SERVICES COM ORACLE EM PL SQL: VANTAGENS E DESVANTAGENS

Trabalho de Conclusão de Curso de especialização, apresentado à disciplina de Trabalho de Diplomação, do Curso de pós-graduação em banco de dados, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de especialista.

Orientador: Marcelo Teixeira

PATO BRANCO
2017

TERMO DE APROVAÇÃO

DESENVOLVENDO WEB SERVICES COM ORACLE EM PL SQL: VANTAGENS E DESVANTAGENS
por

EDUARDO PACHECO

Este Trabalho de Conclusão de Curso foi apresentado em 23 de fevereiro de 2017 como requisito parcial para a obtenção do título de Especialista em Banco de Dados. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Marcelo Teixeira
Prof.(a) Orientador(a)

_____
Vinícius Pegorini
Membro titular

_____
Ives Renê Venturini Pola
Membro titular

"O Termo de Aprovação assinado encontra-se na Coordenação do Curso"

RESUMO

PACHECO, Eduardo. Desenvolvendo Web Service em Oracle PL SQL: vantagens e desvantagens. 2016. 39 f. Monografia de Trabalho de Conclusão de Curso - Curso de pós-graduação em banco de dados, Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Em ambientes organizacionais, é comum nos depararmos com aplicações heterogêneas, desenvolvidas em diferentes plataformas, linguagens, estruturas de dados ou outras formas de aplicações. Na atualidade, é cada vez mais evidente a necessidade de trocar informações pelas estruturas corporativas, internamente ou externamente. Neste contexto, se tratando de aplicações heterogêneas, a troca de informações é dificultada. Uma alternativa focada em ambiente web é o Web Service, os dados podem ser transmitidos para qualquer lugar do mundo, pela rede, através de um protocolo da Web. A implantação de um Web Service inicia-se com o desenvolvimento de métodos, que podem ser escritos em uma linguagem de programação, para então serem publicados na Web. O acesso aos métodos de um Web Service ocorre através de um protocolo de internet. Dessa forma, a comunicação entre sistemas heterogêneos ocorre com a troca de arquivos de dados. Os arquivos enviados e recebidos pelo Web Service podem ser interpretados facilmente, devido ao formato de armazenamento. Um dos formatos bastante utilizado é o WSDL, no qual os parâmetros e as operações são definidos em um arquivo XML. Uma das alternativas hoje é a implementação através da própria base de dados, podendo ser implementado em um banco de dados Oracle com PLSQL. O presente trabalho visa mostrar um comparativo de performance, e as vantagens e desvantagens de cada Web Service.

Palavras-chave: Banco de Dados, Web Service, XML, HTML, linguagem de programação, plataformas Web, WSDL, SOAP.

ABSTRACT

PACHECO, Eduardo. Web Services development using Oracle PL/SQL: advantages and disadvantages. 2016. 39 f. Final Paper Work Concluding Monograph - Postgraduate course in database, Federal Technological University of Paraná. Pato Branco, 2016.

In organizational environments, it is common to come across heterogeneous applications, developed in different platforms, languages, data structures or other forms of applications. At present, the need to exchange information by corporate structures, internally or externally, is increasingly evident. In this context, when dealing with heterogeneous applications, the exchange of information is difficult. An alternative focused on web environments, is the Web Service, data can be transmitted anywhere in the world, over the network, through a web protocol. Deploying a Web Service starts with the development of methods hich can be written a programming language, and then published on the Web. Access to the methods of a Web Service occurs through an Internet protocol. In this way, communication between heterogeneous systems with the exchange of data files. Files sent and received by the Web Service can be easily interpreted due to the storage format. One of the widely used formats is WSDL, in which parameters and operations are defined in an XML file. One of the alternatives is the implementation through the database itself, and can be implemented in an Oracle database with PLSQL. The present work aims to show a comparison of performance, and the advantages and disadvantages of each Web Service.

Keywords: Database, Web Service, XML, HTML, programming language, Web platforms, WSDL, SOAP.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

DDL	*Data Definition Language*
HTML	*HiperText Markup Language*
SGBD	Sistema Gerenciador de Banco de Dados
SQL	*Structured Query Language*
XML	*Extensible Markup Language*
XLS	*Extensible Stylesheet Language*
XLST	*Extensible Stylesheet Language Transformation*
WS	*Web Service*

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do trabalho, abrangendo uma visão geral, da implementação de um Web Service com Oracle em PL SQL, aspectos práticos, os objetivos e a justificativa.

1.1 CONSIDERAÇÕES INICIAIS

De acordo com (ELMASRI, NAVATHE, 2011) em geral as aplicações Web oferecem interfaces, para acessar informações armazenadas em banco de dados, sendo comum utilizar arquiteturas de diversas camadas cliente/servidor. A linguagem mais utilizada na Web é o HTML (Hypertext markup Language), para a representação de dados, junto com sua formatação, no entanto inadequada para o compartilhamento de dados entre sistemas heterogêneos.

Uma utilizada para estruturar documentos e facilitar a troca de informações é a XML (Extensible Markup Language), os dados podem ser extraídos de um banco de dados, armazenados em um arquivo de dados XML, devidamente formatados de acordo com suas respectivas colunas. Nesse formato, os dados podem trafegar através de um protocolo de internet, e ser interpretado facilmente, ou importado para outra base de dados.

Um Web Service é uma aplicação web que permite a troca de informações através de arquivos XML, podendo ser utilizados diversos protocolos web, a aplicação de um Web Service serve para a comunicação entre diferentes aplicações através de troca de informações.

A implementação de um Web Service, através de uma linguagem de programação, requer uma implementação de métodos, sendo que esses métodos podem fazer acesso ou não ao banco de dados. É possível por meio de um banco de dados fazer a implementação de um Web Service, fazendo com que os métodos existentes, possam ser consumidos através da Web, o que pode gerar agilidade e facilidade na manutenção.

1.2 OBJETIVOS

Este trabalho analisa diferentes abordagens para o desenvolvimento de um Web Service usando diferentes tecnologias e paradigmas de desenvolvimento. A seguir serão descritos seu objetivo geral e seus objetos específicos. O objetivo geral está relacionado ao resultado principal obtido com o desenvolvimento deste trabalho e os objetivos específicos o complementam.

1.2.1 Objetivo Geral

Implementar um Web Service com Oracle utilizando as bibliotecas XDB_WEBSERVICES. Então implementar o mesmo WS usando outra tecnologia. Finalmente objetiva-se comparar as duas abordagens e identificando as vantagens e desvantagens de cada uma.

1.2.2 Objetivos Específicos

Como forma de complementar o objetivo geral foram definidos os seguintes objetivos específicos:

- Implementar um WS em Oracle PL SQL
  - Analisar a abordagem buscando vantagens e desvantagens.
  - Verificar a abordagem os métodos disponíveis.
  - Estudar formas de implementação e métodos disponíveis a serem utilizados.

- Implementar um WS em C#.
  - Criação de métodos que possam ser chamados através de uma requisição web.
  - Analisar a abordagem buscando vantagens e desvantagens.
- Consumir WS Oracle/C#.
  - Consumir WS chamando os mesmos métodos criados.

- Fazer comparações de desempenho.
- Apresentar resultados em um gráfico.

1.3 JUSTIFICATIVA

É notável o crescimento do compartilhamento de informações, entre empresas, fornecedores ou colaboradores, para melhoria de seus processos internos ou externos, independente da distancia física que estejam situados. A implementação de um WS é totalmente justificável, devido à facilidade na disponibilização de métodos, que podem ser criados, alterados e publicados instantaneamente na internet. Outro fator importante é o WS utilizar protocolos de rede, para facilitar e deixar seguro a comunicação através da internet, como também padronizar o formato dos dados, como por exemplo, o formato WSDL, que descreve o conteúdo dos dados e seus métodos.

Um WS implementado com Oracle PLSQL, facilita ainda mais, o processo de troca de informações, devido à disponibilização de funções e procedimentos já desenvolvidos no banco de dados. Ao acessar o método na Web, outra vantagem, é o uso das credenciais, para cada usuário, cadastrado no banco, aproveitando as regras de negócios.

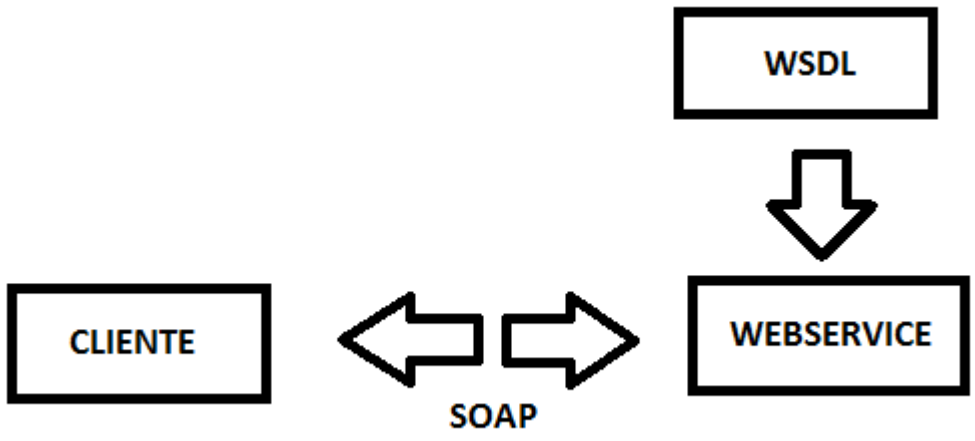

Figura 1: Descrição do funcionamento de um Web Service.

Segundo (MENSAH, ROHWEDDER, 2002) a arquitetura de um WS implementado diretamente na base de dados Oracle, descreve a funcionalidade de banco de dados e a integração de dados de serviços da Web com o motor SQL.

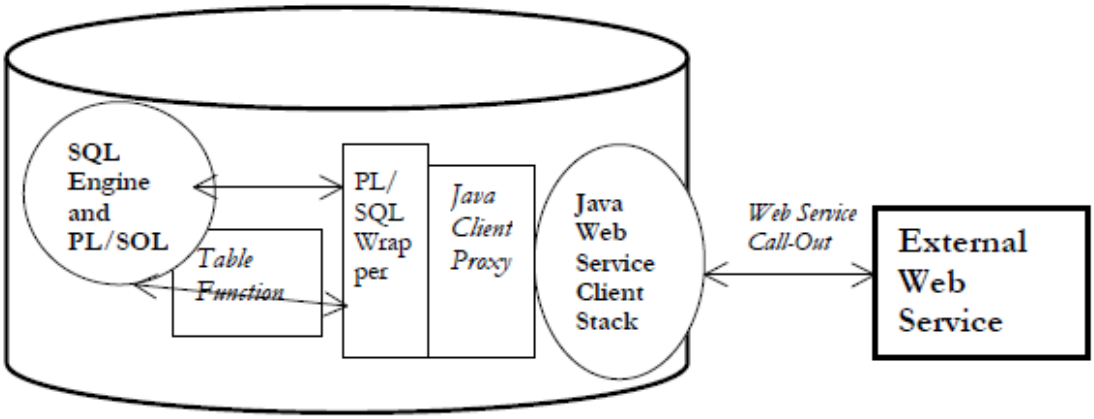

Figura 2: Estrutura de um Web Service em uma linguagem de programação.

1.4 ESTRUTURA DO TRABALHO

No Capítulo 2 o referencial teórico apresentando conceitos relacionados a banco de dados, PL SQL, WS e XML. O Capítulo 3 apresenta as ferramentas e o método utilizado no desenvolvimento do trabalho juntamente com suas características e detalhes. No capítulo 4 é realizada a apresentação do desenvolvimento e os testes realizados.

2 REFERENCIAL TEÓRICO

Este capítulo contém o referencial teórico apresenta conceitos sobre banco de dados, PLSQL, XML e WS.

2.1 BANCO DE DADOS

Segundo Date (2004) um sistema de banco de dados pode ser visto como um armário eletrônico de arquivos, ou seja, um local de armazenamento para arquivos de dados.

> “Dado é qualquer elemento identificado em sua forma bruta que, por si só, não conduz a compreensão de terminado fato ou situação.” “Informação é o dado trabalhado que permite ao executivo tomar uma decisão.” (OLIVEIRA, 2010, p.24).

Em um SGDB são armazenados estruturadamente, os quais podem ser inseridos, alterados ou deletados, como pode ser alterada a estrutura em que os dados são gravados, ou seja, um modelo relacional o qual pode ser definido conforme a necessidade de organização dos dados a serem inseridos na base.

Um Sistema Gerenciador de Banco de Dados (SGBD) tem como princípio gravar as informações de forma a garantir a integridade dos dados e a consistência, sendo que existem diversos (SGBD), cada um com suas características e recursos.

Date (1985) refere se a um sistema de banco de dados, como um software computadorizado de arquivamento de registros, muitos dos arquivos gravados em papéis, podem ser guardados de forma mais conveniente utilizando um banco de dados.

Silberschatz, Kotth e Sudarshan (1999) indicam a arquitetura de três camadas para a modelagem de um banco de dados:

a) Nível físico – descreve como esses dados estão armazenados. As estruturas dos dados.

b) Nível lógico – descreve quais dados estão armazenados no banco e os relacionamentos entre os mesmos.

c) Nível visão – descreve apenas parte do banco de dados. Dessa forma um mesmo banco de dados pode ser representado por diversas visões que são definidas de acordo com interesses.

Um relacionamento é constituído por um ou mais atributos, definido como campos que traduzem, o tipo de dados a armazenar. Cada instância do esquema (linha) é chamada de linha (registro). Um modelo é uma forma de representar a lógica de dados e os respectivos relacionamentos, sem a preocupação do armazenamento físico. Um modelo conta com operadores semânticos que fazem a representação de cada forma de informação em determinado nível de abstração (TAKAI, ITALIANO, FERREIRA, 2005).

Na atualidade os bancos de dados possuem bastante utilização e aplicabilidade nas organizações, devido a rapidez na consulta, atomicidade, consistência, isolamento, integridade, recursos como mineração e cubo de dados, com a finalidade de coleta de informações relevantes para a tomada de decisão e identificação de padrões. Alguns SGDBS permitem fazer, a integração de um banco de dados a web, publicando seus métodos através de um WS utilizando um protocolo de rede.

2.2 PL SQL

PL/SQL é uma linguagem de programação, para desenvolver procedimentos específicos para linguagem Oracle, sendo similar a linguagem SQL.

Segundo (WATSON, 2010) o PL/SQL é executado no banco de dados, porém também pode ser salvo, ou armazenado no cliente ou aplicação. Podendo ser adicionado através do SQL plus, é armazenado e nomeado, quando é compilado o processo de compilação verifica se existem erros de compilações.

Através da programação estruturada em um banco de dados é possível fazer processamento de dados, através de instruções, estruturas de repetição e recuperação dos dados armazenados.

> “Estruturas de programação adicionais, como loops e instruções condicionais, são acrescentadas à linguagem de banco de dados para convertê-la em uma linguagem de programação completa. Um exemplo dessa técnica é a PL/SQL da Oracle.” (ELMASRI, NAVATHE, 2011, p.304).

De acordo com (WATSON, 2010) o banco de dados Oracle, suporta utilizar SQL (Structured Query Language), porém para algumas situações, são necessários procedimentos mais complexos para recuperação e alteração de dados, nesse caso é necessário a utilização de uma linguagem estrutural, sendo que através do

PL/SQL se pode misturar procedimentos estruturais com consultas SQL, facilitando a criação e manutenção de rotinas armazenadas.

2.3 XML

De acordo com (ELMASRI, NAVATHE, 2011) a XML pode ser utilizada para definir dados e informações sobre estrutura de dados, pode também ser utilizado uma linguagem de formatação com XSLT, recentemente o XML foi proposto como modelo para armazenar e recuperar dados, mesmo que poucos sistemas de experimento de banco de dados foram desenvolvidos até agora para essa abordagem.

Uma das vantagens de dados em XML é que através de tags é possível especificar um modelo, de acordo com cada respectivo campo e seu tipo de dados, também nessa mesma camada definir métodos de aplicação, através de um modelo.

a)Descrição da estrutura de um XML

```
<Destinatario>
<Nome>João da Silva</Nome>
   <Endereco>
      <Rua>Rua Cinco</Rua>
      <Bairro>Centro</Bairro>
      <Numero>460</Numero>
      <Cidade>
            <Descricao>Mariópolis</Descricao>
            <UF>Paraná</UF>
       </Cidade>
  </Endereco>
</Destinatario>
```

Quadro 1: Listagem de um arquivo XML.

Extensible Markup Language (XML) é um formato de texto simples, muito flexível que veio de SGML (ISO 8879). Desenvolvido para gerar a possibilidade da publicação eletrônica em larga escala, XML também está desempenhando um papel cada vez mais importante na troca de uma ampla variedade de dados na Web (W3C, 2016).

2.4 WEB SERVICE

Um WS é um conjunto de recursos publicados na Web, que trocam informações por arquivos XML, WSDL JSO entre outros, e usam para comunicação

protocolos HTTP, na atualidade são muito utilizados por corporações, sendo que pode ser disponibilizado o recurso de um WS para o publico em geral, para resgatar ou enviar informações.

Segundo (MENSAH, ROHWEDDER, 2002) o World Wide Web Consortium (W3C) define um serviço Web como um software aplicativo ou componente com as seguintes propriedades:

· É identificado por uma URL;

· Suas interfaces e de ligação geralmente são em XML.

·Interage diretamente com serviços Web via XML, por meio de protocolos de internet.

Os termos SOAP, WSDL são essenciais para a compreensão da arquitetura de serviços da Web.

De acordo com (MENSAH, ROHWEDDER, 2002) são descritos como:

· SOAP é um protocolo de mensagem baseado em XML utilizado pelos serviços da Web. O transporte é pelo mecanismo (HTTP, FTP, SMTP, JMS).

· Web Services Description Language (WSDL) é um formato de documento XML que especifica as operações e seus parâmetros - incluindo tipos de parâmetros é fornecida por um serviço Web. Além disso, descreve a localização, o transporte protocolo, e o estilo de invocação para o serviço.

De acordo com TABOR (2002) o XML se mostrou atraente em ponto de vista técnico, com a descrição dos dados através de tags, vista como rótulos, sendo que pode ser aberto por todas as linguagens existentes de programação, com o surgimento da especificação SOAP, foi destacável o valor de um XML, ao chamar um procedimento remoto para trocar informações não visuais com uma plataforma de computação, essa tecnologia pode resolver muitos problemas se tratando da troca de informações com parceiros comerciais.

Em outras palavras em uma aplicação o SOAP é o protocolo responsável por trafegar os dados em XML, e o WSDL é responsável por definir a estrutura do WS que será acessado, para que possa ser interpretado na aplicação.

De acordo com TABOR (2002) o SOAP (Protocolo de acesso simples de objeto) é uma especificação que define como as mensagens podem ser enviadas entre dois sistemas de software, com o uso de XML. Essas mensagens normalmente seguem um padrão de requisição e de resposta. A chamada é feita por um

computador através de um método, e o outro computador executa alguma computação ou serviço e os retornos resultado para o aplicativo chamado.

Uma das vantagens do WSDL é que é possível na aplicação, saber da estrutura do WS disponível, assim se tem uma visão de todos os métodos presentes no mesmo e atributos, algumas linguagens de programação definem automaticamente essa implementação e já traz uma chamada para o método pronta, como na imagem a seguir, a linguagem C# com a IDE Visual Studio a listagens dos métodos disponíveis em cada WS.

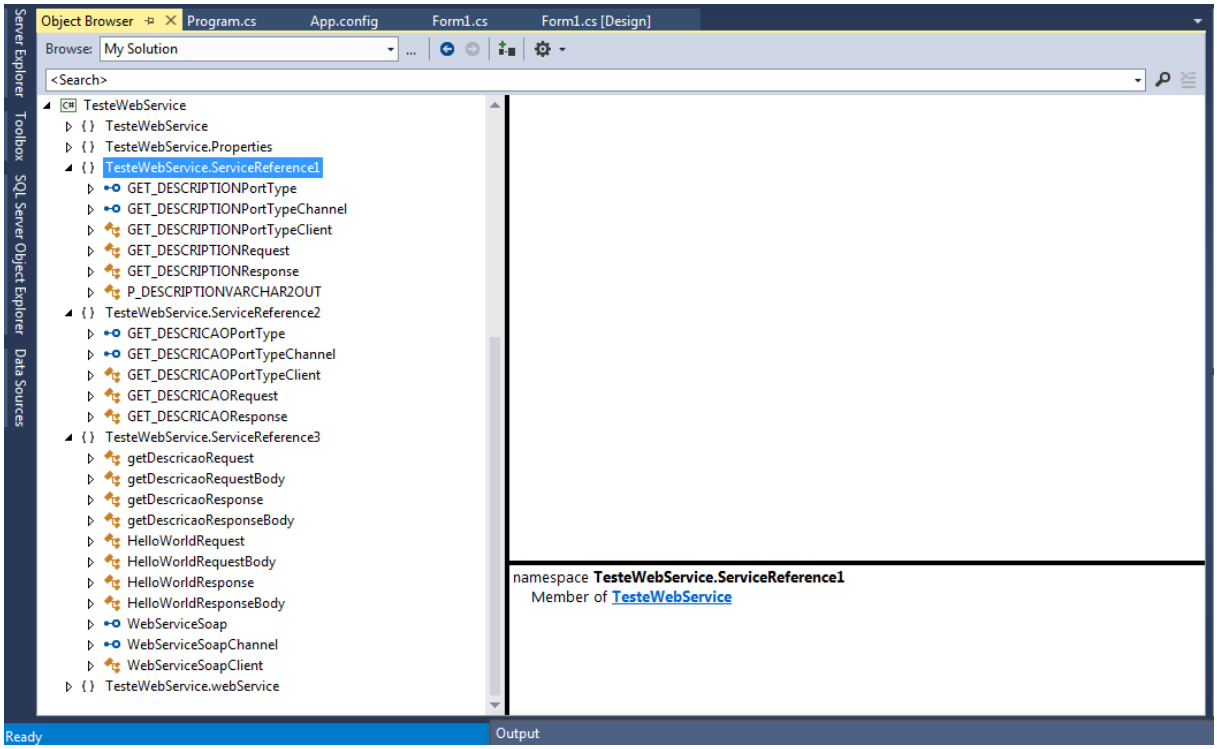

Figura 3: Métodos carregados automaticamente pelo Web Service.

Sendo que se alguma mudança como inclusão de um método novo ou alteração de um existente for feita na estrutura do WS pode se também atualizar os métodos através do comando Update Service Reference, disponível no Visual Studio Community 2013.

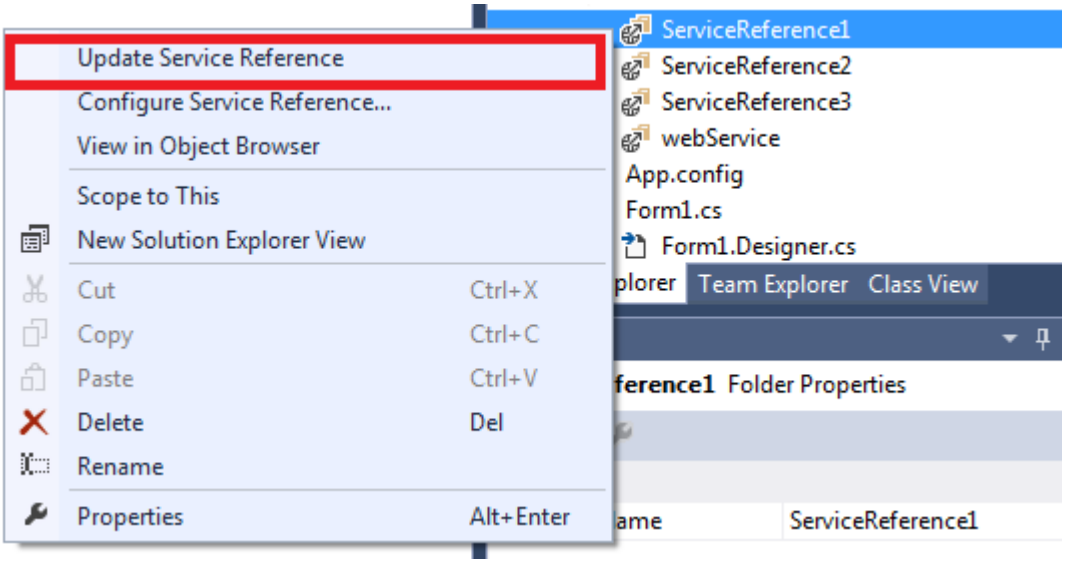

Figura 4: Atualizando as referencias.

Após a execução do comando Update Service Reference o download das especificações do WS serão realizadas automaticamente.

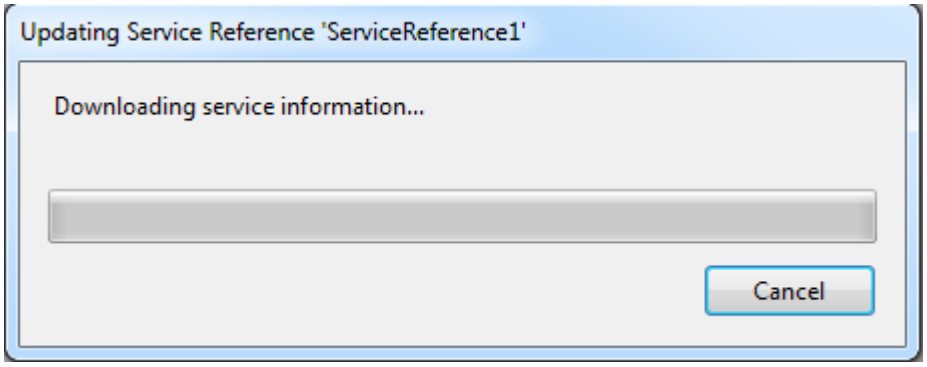

Figura 5 – Download das atualizações.

O WSDL supre essa necessidade de especificação da estrutura de dados e de métodos definindo um padrão XML para descrever serviços de rede como coleções de pontos finais de comunicação capazes de trocar mensagens. A definição do WSDL fornece um modelo para sistemas distribuídos e servem como uma receita para automatizar a forma do funcionamento da aplicação que será desenvolvida. Isso permite a reutilização de definições abstratas: mensagens que são descrições abstratas dos dados que estão sendo trocados, e os tipos de portas que são coleções abstratas de operações a serem realizadas. Uma porta está definida associando um endereço de rede com uma ligação reutilizável e um conjunto de portas de definir um serviço.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentadas as ferramentas e tecnologias utilizadas para implementar e consumir o WS proposto, sendo todas as ferramentas necessárias, desde a criação e manutenção do banco de dados e implementação do WS Oracle, até a linguagem de programação utilizada, para criar um WS em C# e para consumir os WS para que assim possa ser efetuado um comparativo de desempenho.

3.1 MATERIAIS

Serão utilizadas as seguintes ferramentas e tecnologias:

a) SQL Developer – como ferramenta administrativa do bando de dados.

b) Visual Studio – como IDE de desenvolvimento.

c) ORACLE – banco de dados utilizado.

d) C# - linguagem de programação utilizada.

3.1.1 SQL Developer

A SQL Developer foi a ferramenta escolhida para fazer o gerenciamento do banco de dados devido sua interface de fácil usabilidade e possuir todos os recursos necessários para essa implementação. Devido ser desenvolvida pela Oracle. Interface do ambiente SQL Developer conforme a Figura 6.

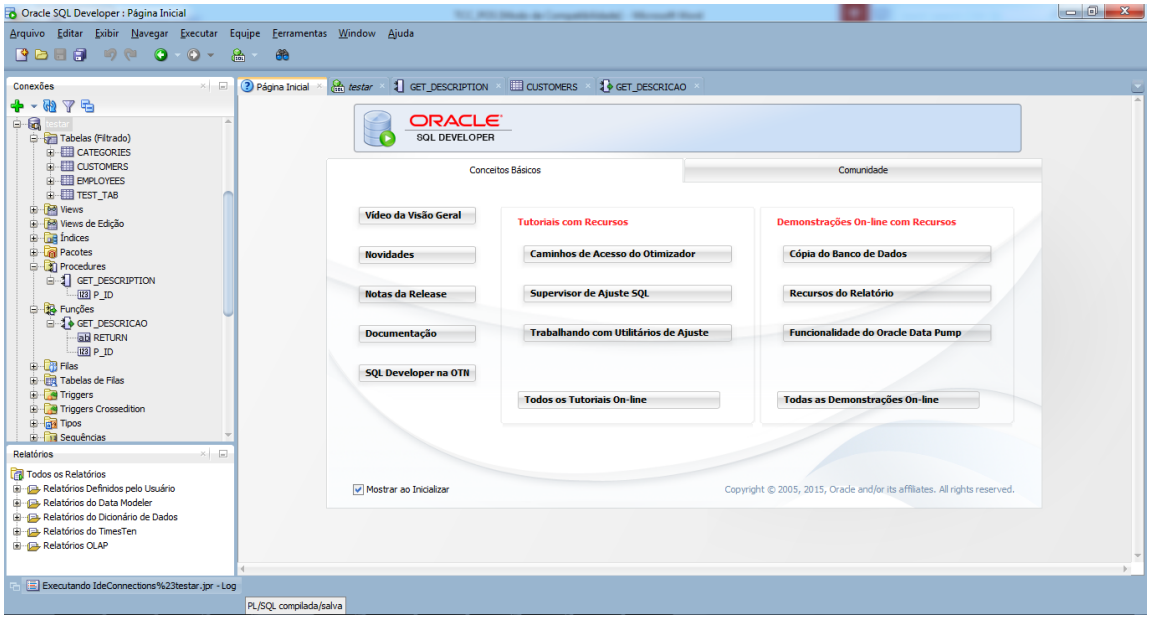

Figura 6: Ambiente e trabalho SQL Developer.

3.1.2 Visual Studio

Devido a facilidade de implementar e consumir um WS, e a facilidade de conexão com o banco de dados Oracle a IDE de desenvolvimento escolhida foi o VisualStudio 2013 Community, com a Interface conforme Figura 7.

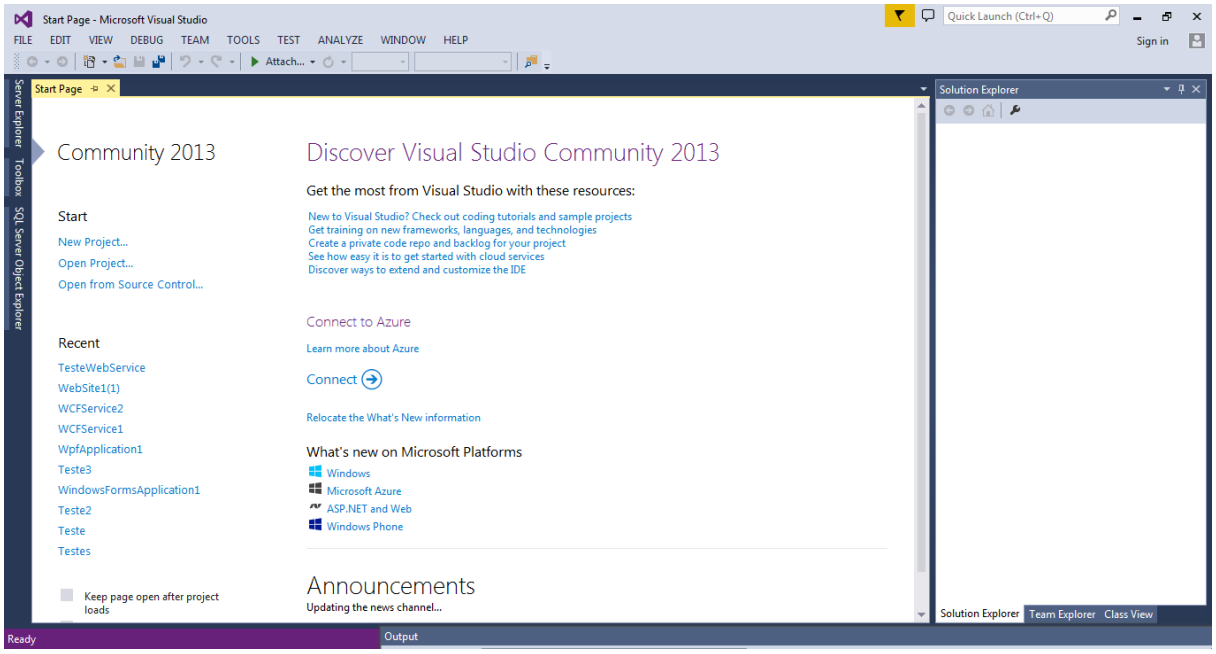

Figura 7: Ambiente de desenvolvimento Visual Studio Community 2013.

3.1.3 Banco de Dados Oracle

De acordo com (WATSON, 2016) o número de aplicações Oracle tem aumentado nos últimos anos, devido à preferência pelo SGBD ter crescido pelas empresas e organizações de médio e grande porte.

O banco de dados Oracle foi escolhido para implantação de um WS, devido possuir os recursos integrados necessários, sendo assim adequado para a implementação, e já através da mesma ser possível acessar todos os métodos disponíveis no banco, sendo que para utilizar os recursos mencionados acima pode se contar com os métodos presentes na biblioteca DBMS_XDB.

3.1.4 Web Service Oracle

Conforme MENSAH e ROHWEDDE (2002) fornecedores de banco de dados começaram a oferecer acesso a procedimentos armazenados disponibilizados, através de interfaces de serviços Web. Banco de dados de WS representa uma

visão do banco de dados de serviços Web e de trabalho representam em duas direções:

- Acessar os recursos do banco de dados como um serviço Web, e consumir um WS externo a partir de um procedimento no banco de dados. Girando o banco de dados Oracle em um serviço Web.
- Desenvolvimento de pacotes PL / SQL, consumirem serviços Web externos do próprio banco de dados e integração com o SQL.

De acordo com Oracle um WS em um banco de dados é baseado nos mecanismos padrão dos protocolos de rede, independente da linguagem de aplicação, ele é capaz de definir e publicar métodos na web, podendo ser acessado por usuários ou sistema.

Um WS implementado em uma base de dados pode ser consumido normalmente, da mesma forma que um implementado em uma linguagem de programação, o que difere é que os blocos escritos em PL/SQL, procedimentos, funções podem ser aproveitados, diminuindo o custo de manutenção.

Segundo MENSAH e ROHWEDDER (2002) Web Service de banco de dados se baseia em mecanismos padrões e protocolos, serviços Web dessa forma não dependem de linguagem de programação, o WS publicado é capaz de ser consumido por serviços internos em sua aplicação da mesma forma que pode se consumir um WS através de um bloco PL/SQL. Os procedimentos podem ser escritas em PL/SQL ou em Java e após isso serem publicados como serviços na internet.

De acordo com MENSAH e ROHWEDDER (2002) algumas vantagens de serviços da Web de banco de dados são:

- Operações de banco de dados de pedidos de serviços da Web fornecida através de um padrão.
- Forma segura e controlada de abertura a lógica de dados banco de dados.
- Compartilhamento de dados, devido aproveitamento dos controles existentes.

3.1.5 Linguagem de Programação C#

A linguagem de programação C# foi escolhida devida a facilidade de acesso a um WS sendo ele implementado em diversos modelos e com protocolos sendo WSDL ou SOAP, e a facilidade de criação de um WS através da mesma linguagem,

com componentes prontos, robustos, e também podendo contar com o componente Chart para a exibição dos gráficos necessários.

Segundo Microsoft (2016) é uma linguagem de programação com tipos protegidos, orientada a objetos e que possibilita a construção de aplicações confiáveis e compatíveis com o .NET Framework. Pode ser usado para aplicativos Windows, serviços Web XML, componentes distribuídos, aplicativos de cliente-servidor, aplicativos para banco de dados.

3.2 MÉTODO

Inicialmente foi necessário estudar as tecnologias envolvidas, nas etapas do desenvolvimento e consumo do WS.

Após estudos o WS foi implementado na linguagem C# e também em Oracle, após testes foram realizados consumindo, o WS nas duas implementações, e feito uma analise, com base nos dados coletados.

4 ESTUDO DE CASO

Este capítulo apresenta como foi implementado e o que foi obtido, como resultado da realização deste trabalho, que é um comparativo de desempenho entre tecnologias e a analise da coleta de dados.

4.1 DESENVOLVIMENTO DO WEB SERVICE EM ORACLE

É necessário iniciar o Oracle XML DB HTTP, para ativar os recursos do Web Service Oracle, a configuração da porta HTTP, uma das mais utilizadas em aplicações Web é a 8080, pode ser configurado através da chamada a seguir executada através do usuário sysdba: EXEC dbms_xdb.sethttpport(8080);

Para verificar se as alterações foram efetuadas e a porta alterada realizar a seguinte leitura: SELECT dbms_xdb.gethttpport FROM dual;

De acordo com Oracle é necessário adicionar o Servlet e configurar o ORAWSV Servelet, o procedimento pode ser feito através desse procedimento, conectado com o usuário sysdba, como no código apresentado no Quadro 2.

a)Procedimento executado para adicionar o Servlet e configuração do ORAWSV com usuário sysdba

```
DECLARE
  l_servlet_name VARCHAR2(32) := 'orawsv';
BEGIN
  DBMS_XDB.deleteServletMapping(l_servlet_name);

  DBMS_XDB.deleteServlet(l_servlet_name);

  DBMS_XDB.addServlet(
    name     => l_servlet_name,
    language => 'C',
    dispname => 'Oracle Query Web Service',
    descript => 'Servlet for issuing queries as a Web Service',
    schema   => 'XDB');

  DBMS_XDB.addServletSecRole(
    servname => l_servlet_name,
    rolename => 'XDB_WEBSERVICES',
    rolelink => 'XDB_WEBSERVICES');

  DBMS_XDB.addServletMapping(
    pattern => '/orawsv/*',
    name    => l_servlet_name);
END;
```

Quadro 2: Código utilizado para adicionar ServLet.

De acordo com Oracle é necessário a alteração do arquivo de configuração através de uma query, que pode ser executada como sysdba, conforme descrito no Quadro 3.

a) Alterar arquivo de configuração através de uma query

```
SET LONG 10000
XQUERY declare default element namespace "http://xmlns.oracle.com/xdb/xdbconfig.xsd"; (:
:)
      (: This path is split over two lines for documentation purposes only.
         The path should actually be a single long line. :)
      for $doc in fn:doc("/xdbconfig.xml")/xdbconfig/sysconfig/protocolconfig/httpconfig/
       webappconfig/servletconfig/servlet-list/servlet[servlet-name='orawsv']
      return $doc
```

Quadro 3: Query para alterar arquivo de configuração.

Para utilização do WS é necessário a criação do usuário e atribuir as permissões devidas ao mesmo.

Comando de criação do usuário:

```
CREATE USER testar IDENTIFIED BY test QUOTA UNLIMITED ON users;
```

Adicionar permissão ao usuário para conectar, criar tabelas e procedimentos:

```
GRANT CONNECT, CREATE TABLE, CREATE PROCEDURE TO testar;
```

Adicionar permissão ao usuário à biblioteca XDB_WEBSERVICES:

```
GRANT XDB_WEBSERVICES TO testar;
```

Adicionar permissão ao usuário sobre o HTTP:

```
GRANT XDB_WEBSERVICES_OVER_HTTP TO testar;
```

Essa permissão é opcional caso queira tornar publico:

```
GRANT XDB_WEBSERVICES_WITH_PUBLIC TO testar;
```

Então são criadas tabelas e definidos os métodos no usuário testar para efetuar os testes, conforme o Quadro 4.

a) Alterar arquivo de configuração através de uma query

```
CONN testar/test

CREATE TABLE test_tab (
  id          INTEGER,
```

```
  description VARCHAR2(50),
  CONSTRAINT test_tab_pk PRIMARY KEY (id)
);
INSERT INTO test_tab (id, description) VALUES (1, 'ONE');
INSERT INTO test_tab (id, description) VALUES (2, 'TWO');
INSERT INTO test_tab (id, description) VALUES (3, 'THREE');

CREATE OR REPLACE PROCEDURE get_description (
  p_id          IN  test_tab.id%TYPE,
  p_description OUT test_tab.description%TYPE) AS
BEGIN
  SELECT description
  INTO   p_description
  FROM   test_tab
  WHERE  id = p_id;
END;
```

Quadro 4: Query para alterar arquivo de configuração.

Os métodos criados estarão disponibilizados a partir do seguinte interesse e retornaram um XML, que poderão ser acessados diretamente ou por uma linguagem de programação onde pode existir uma interface a serem informados os parâmetros do procedimento.

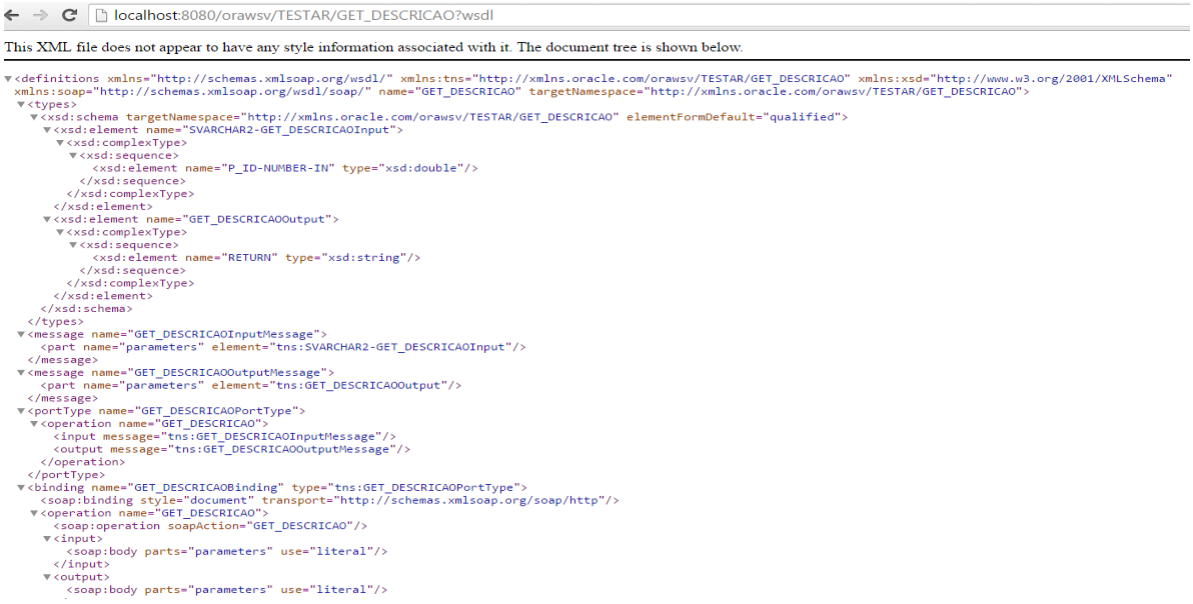

Figura 8: Chamada de método GET_DESCRICAO() do Web Service.

4.2 CONSUMINDO WEB SERVICE ATRAVÉS DE UMA APLICAÇÃO C#.

Através da linguagem de Programação C# foi desenvolvido uma chamada para a GetDescricao() definida no Oracle, e uma interface passando o número do ID e chamando o evento de um botão para pegar a descrição.

Configuração da referencia do serviço:

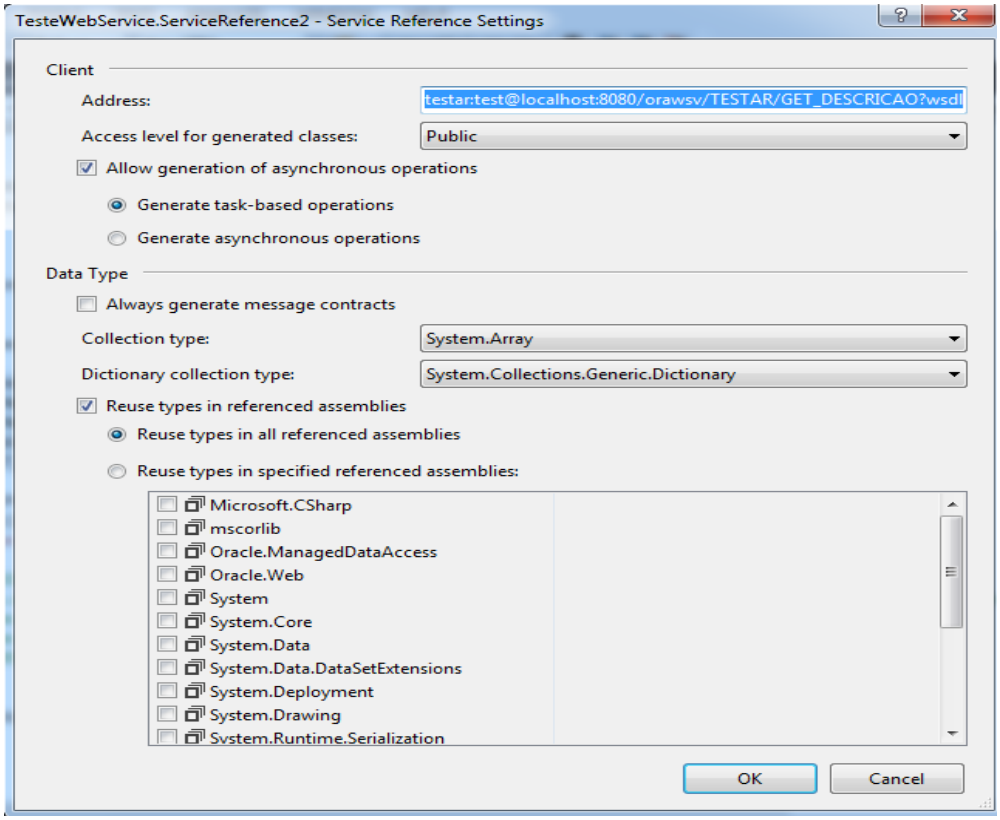

Figura 9: Configurando acesso ao serviço.

Acessando o Método getDescricao() através do C#:

a) Método btnPegaDescricao_Click

```
var ws = new ServiceReference2.GET_DESCRICAOPortTypeClient();
ws.ClientCredentials.UserName.UserName = "testar";
ws.ClientCredentials.UserName.Password = "test";
double id = Convert.ToDouble(txtID.Text);
txtDescricao.Text = ws.GET_DESCRICAO(id);
```

Quadro 5: Método adicionado ao botão para chamado do Web Service.

Exemplo do funcionamento do programa em execução fazendo a chamado do WS implementado em Oracle:

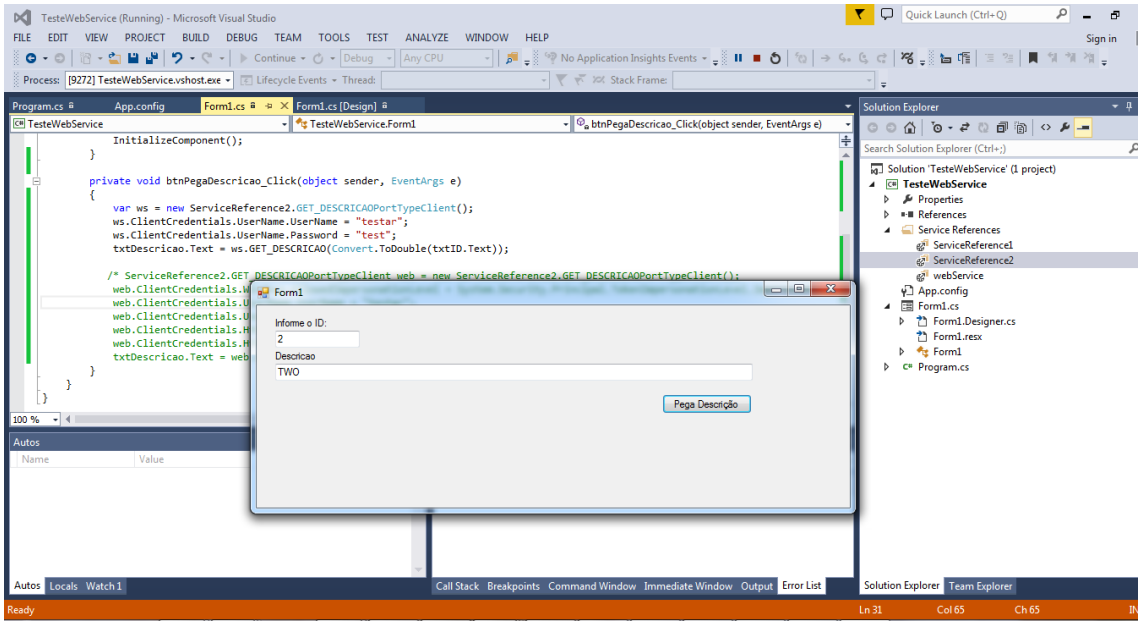

Figura 10: Chamada Web Service Oracle pelo C#.

4.3 CRIANDO UM WEB SERVICE COM VISUALSTUDIO C#.

Foi necessário desenvolver um WS com C# o qual faz a chamada ao mesmo método que está disponível no WS Oracle, o método getDescricao(), o primeiro passo foi criar a aplicação Web, que por padrão já vem com o método HelloWord(), para que seja possível fazer um comparativo entre um WS Oracle e um implementado em C#.

O componente WebService.asmx foi adicionado ao projeto e foi cria a classe com o nome do WebService, tendo sido criada como herança da biblioteca “System.Web.Services.WebService” como padrão ao adicionar o componente o método HelloWord vem criado, após isso após cada método adicionado a classe do WS foi acrescentado o ‘[WebMethod]’ para que possa ser reconhecido como um método interno, para que possa ser chamado a partir de uma nova aplicação, ou mesmo da página principal da Web ao iniciar a aplicação.

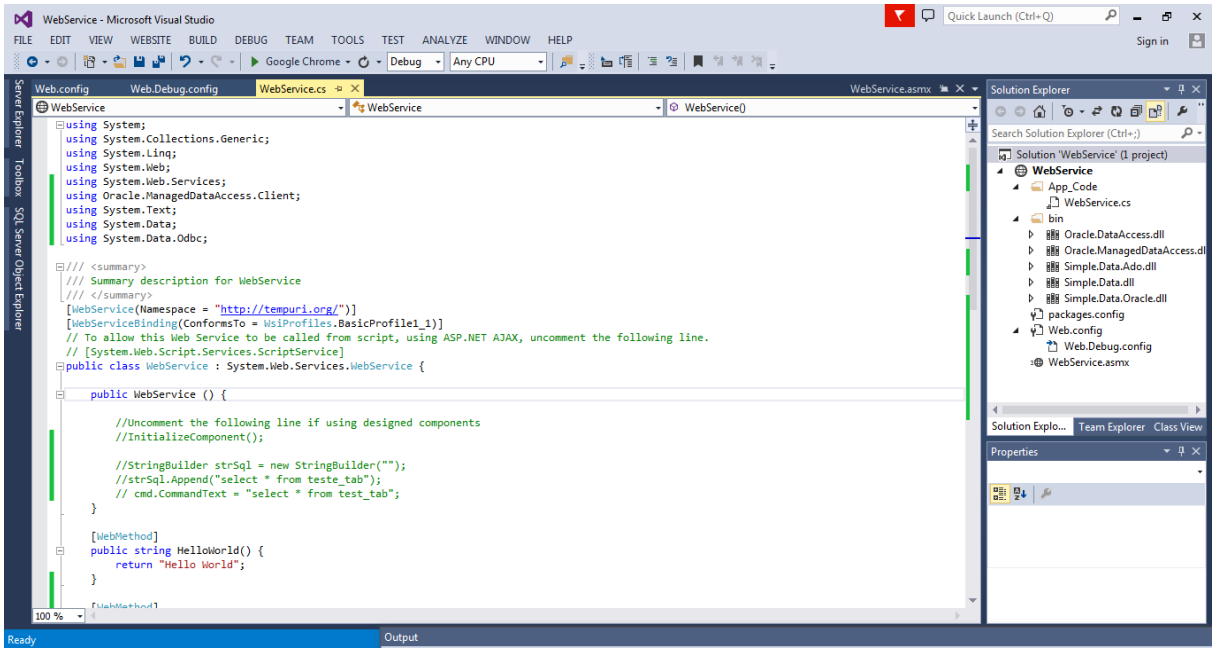

Figura 11: Web Service C# implementado.

Foi necessário desenvolver a chamado do método GetDescricao() da seguinte forma:

a) Método GetDescricao() C#

```
 [WebMethod]
 public string getDescricao(Decimal id)
 {

        String connection =
            "Data Source=localhost:1522/orcl;User ID=TESTAR; Password=test;";
        OracleConnection conn = new OracleConnection(connection);

        var cmd = new OracleCommand();
        cmd.Connection = conn;

        cmd.CommandText = "get_descricao";
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("GET_DESCRICAO",OracleDbType.Varchar2,
ParameterDirection.ReturnValue);
        cmd.Parameters["GET_DESCRICAO"].Size = 50;
        cmd.Parameters.Add("P_ID", id);

        cmd.Connection.Open();
        cmd.ExecuteNonQuery();
        String descricao = cmd.Parameters["GET_DESCRICAO"].Value.ToString();
        conn.Close();
        return descricao;
  }
```

Quadro 6: Método GetDescricao() Web Service C#.

É necessário deixar o WS em execução ou publicar para que os métodos estejam visíveis através de um protocolo de internet, que pode ser acessado através de um endereço HTTP.

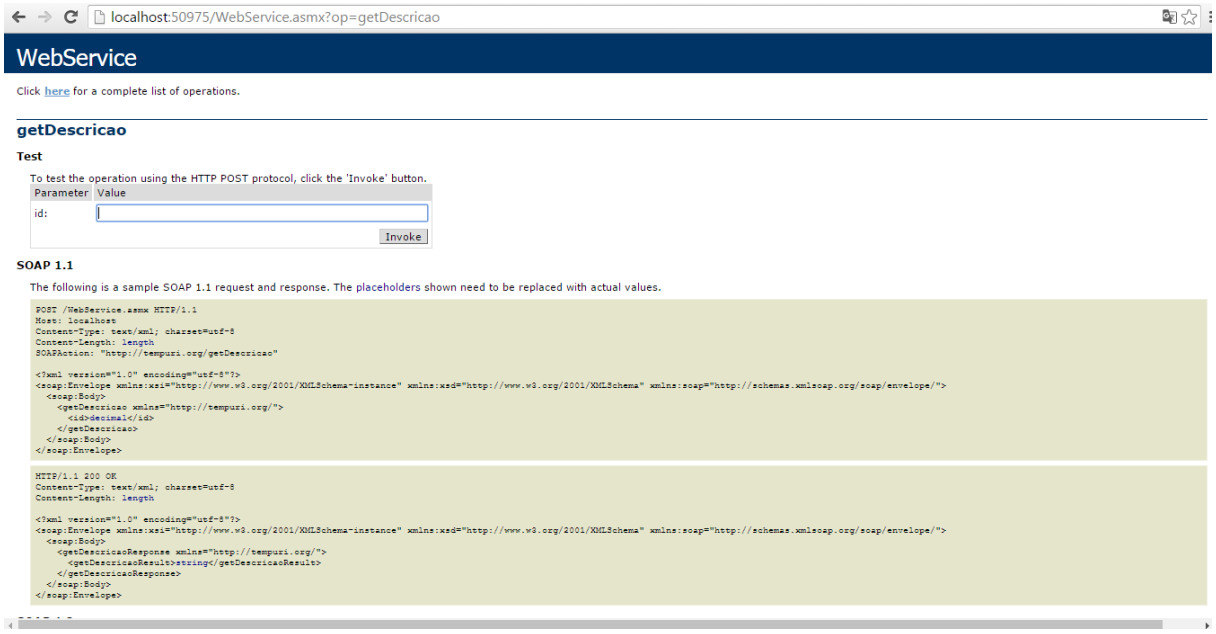

Figura 12: Tela para chamada do método.

Após a chamada do método, o retorno será através de uma estrutura XML como a mostrada na figura 13.

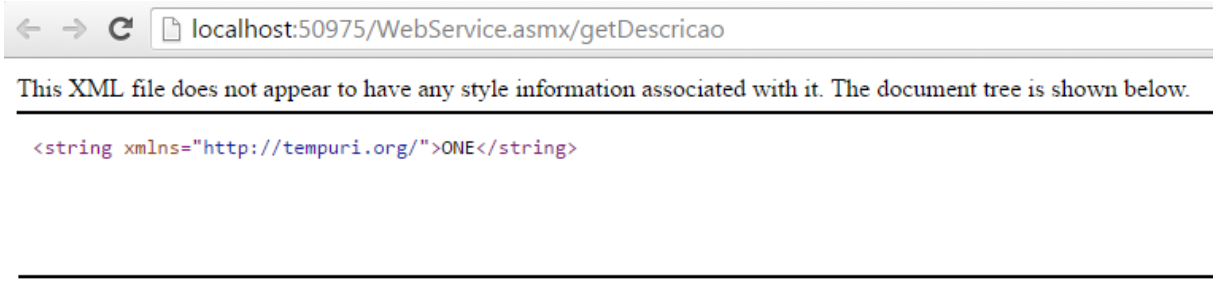

Figura 13: Formato do retorno do método getDescricao().

4.4 COMPARATIVOS ENTRE WEB SERVICE C# E WEB SERVICE ORACLE.

Para realizar um comparativo foi desenvolvida uma aplicação capaz de fazer a chamada ao método getDescricao() implementadas no WS Oracle, e no WS desenvolvido no C#.

A seguir, apresenta-se o método implementado, para fazer a chamada do WS C# com o número de requisições definido para aumentar o uso do WS, e adicionar o resultado em milissegundos em um gráfico.

a) Chamada método GetDescricao() C# adicionando o número de requisições

```
private void btnPegaDescricaoC_Click(object sender, EventArgs e)
{
    Stopwatch sw = new Stopwatch();
    sw.Start();
    int requisicoes = 1;
    if (chbRequisicoes.Checked)
    {
        requisicoes = Convert.ToInt32(txtRequisicoes.Text);
    }
    for (int i = 0; i < requisicoes; i++)
    {
        var ws = new ServiceReference3.WebServiceSoapClient();
        ws.ClientCredentials.UserName.UserName = "testar";
        ws.ClientCredentials.UserName.Password = "test";
        Decimal id = Convert.ToDecimal(txtID.Text);
        txtDescricao.Text = ws.getDescricao(id);
    }
    sw.Stop();
    TimeSpan tempo = sw.Elapsed;
    grafico.Series.Add("WebService C#").Points.AddXY("C#", tempo.TotalSeconds);
}
```

Quadro 7: Chamada do método GetDescricao() Web Service C# por requisições.

No Quadro 7, foi desenvolvido uma chamada, para método GetDescricao(),possa consumir o Web Service Oracle. Utilizando a mesma lógica do método a ser utilizado pelo Web Service C# o número de requisições como parâmetro.

a) Chamada método GetDescricao() Oracle adicionando o número de requisições

```
private void btnPegaDescricao_Click(object sender, EventArgs e)
{
    Stopwatch sw = new Stopwatch();
    sw.Start();
    int requisicoes = 1;
    if (chbRequisicoes.Checked)
    {
        requisicoes = Convert.ToInt32(txtRequisicoes.Text);
    }

    for (int i = 0; i < requisicoes; i++)
    {
        var ws = new ServiceReference2.GET_DESCRICAOPortTypeClient();
        ws.ClientCredentials.UserName.UserName = "testar";
        ws.ClientCredentials.UserName.Password = "test";
        double id = Convert.ToDouble(txtID.Text);
        txtDescricao.Text = ws.GET_DESCRICAO(id);
    }
    sw.Stop();
    grafico.Series.Add("WebService                    ORACLE").Points.AddXY("ORACLE",
tempo.TotalSeconds);
}
```

Quadro 8: Chamada método GetDescricao() Web Service Oracle por requisições.

O Primeiro teste foi uma chamada com uma requisição para cada WS. A requisição demorou menos de um segundo para ser executada em ambas as tecnologias. O WS Oracle obteve o retorno mais rápido, com uma diferença em milissegundos em relação ao WS C#, conforme Figura 14.

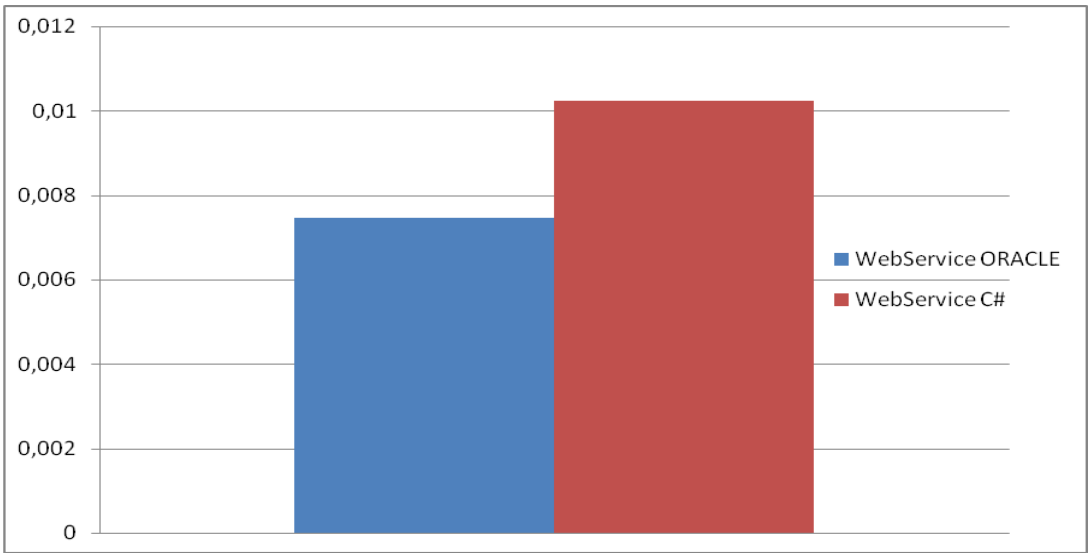

Figura 14: Comparação do tempo de resposta do Web Service Oracle e C# com uma repetição em milissegundos.

No sentido de consolidar o primeiro experimento foi realizada uma sequência de cinco chamadas, de uma requisição cada chamada, para cada WS. O resultado é mostrado na Figura 15.

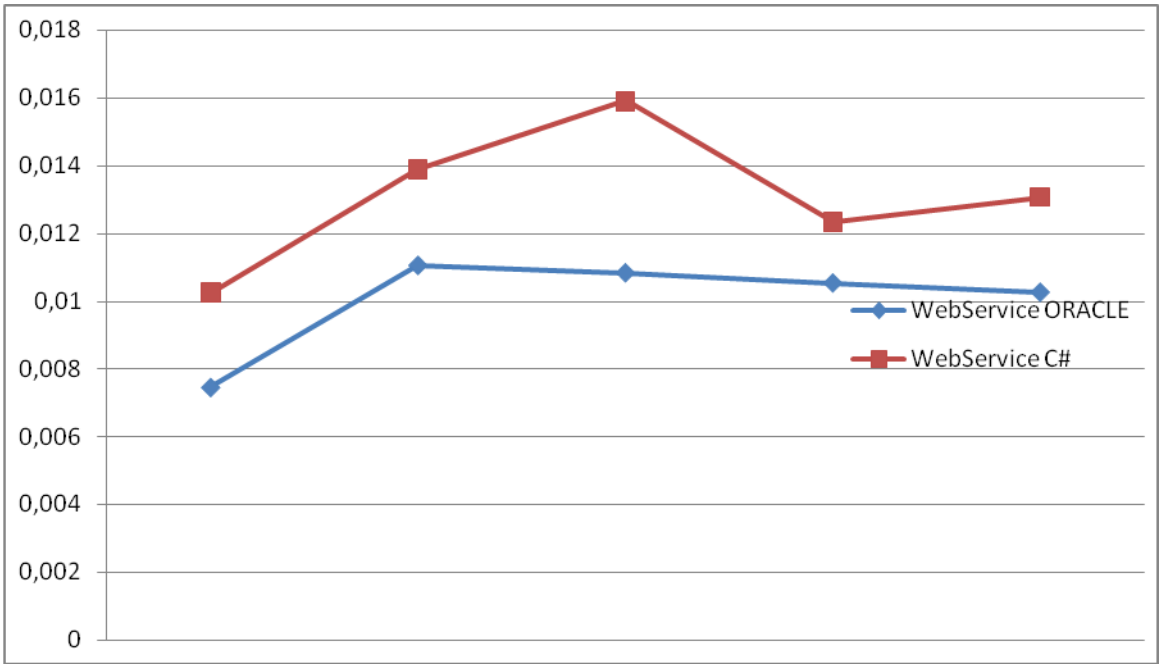

Figura 15: Replicação do experimento com carga de uma requisição com tempo em milissegundos.

É possível observar que o desempenho, em termos de tempo de resposta de cada WS, sofre pequenas variações. Em geral o WS em Oracle retorna uma resposta (para uma única requisição, ou seja, sem filas) em média em 0.01 segundos. Já o WS em C# demora em média 0.06 segundos para retornar.

O Segundo teste foi o consumo de uma chamada para cada WS com dez mil repetições. O intuito desse experimento é gerar fila de requisições para analisar como cada tecnologia se comporta.

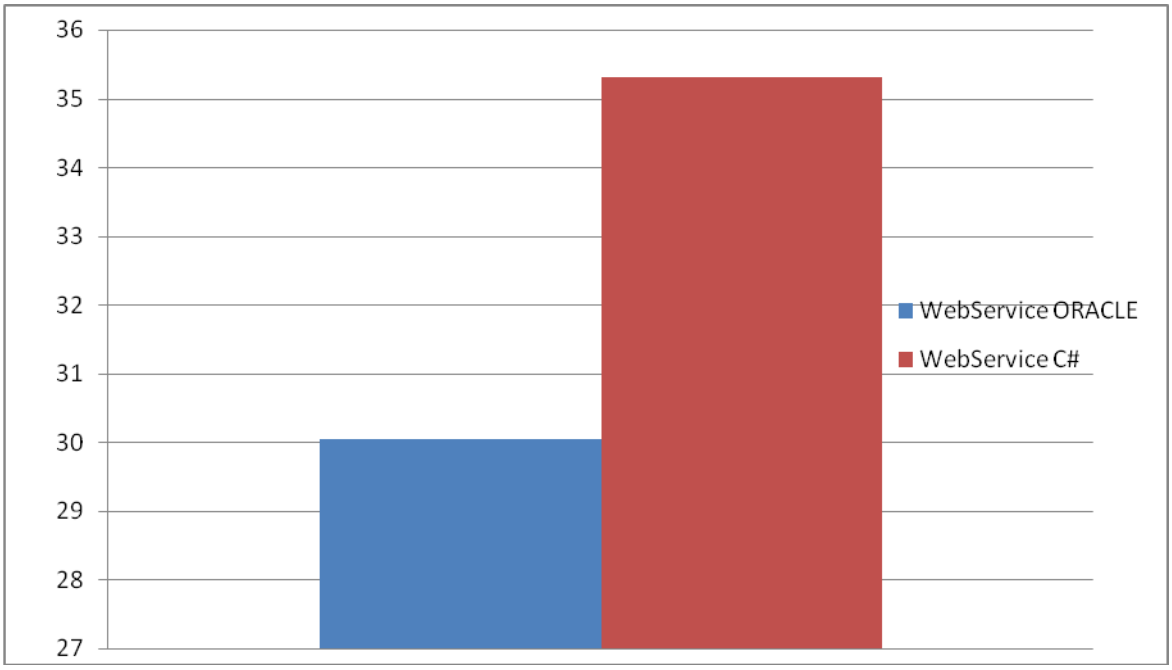

Figura 16: Comparação do tempo de resposta do Web Service Oracle e C#, com carga de 10000 repetições.

Conforme Figura 16, o WS Oracle foi mais rápido, em relação ao WS C#, resultando uma diferença de aproximadamente 5 segundos.

Foi repetido o experimento com carga de dez mil repetições, cinco vezes para sua analise estatística. Pode-se observar que o WS Oracle novamente, retornou mais rápido que o WS em C#, conforme mostra a Figura 17.

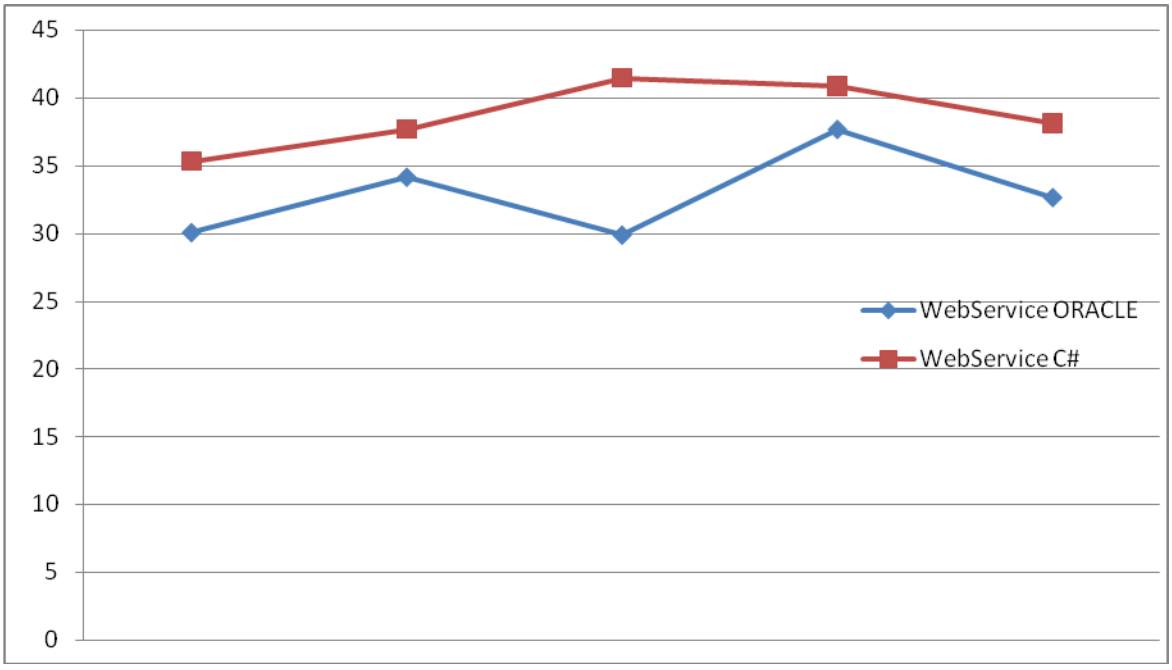

Figura 17: Replicação do experimento com carga de 10000 repetições.

Como exibido na Figura 17 a média geral das cinco amostras, ficaram em aproximadamente em 5 segundos, similar a Figura 16.

Na sequência foi aumentado o número de repetições para vinte mil, para que seja possível analisar em uma amostra maior, se existem diferenças mais significativas, ou se o crescimento da diferença é exponencial, o resultado desse experimento será mostrado na Figura 18.

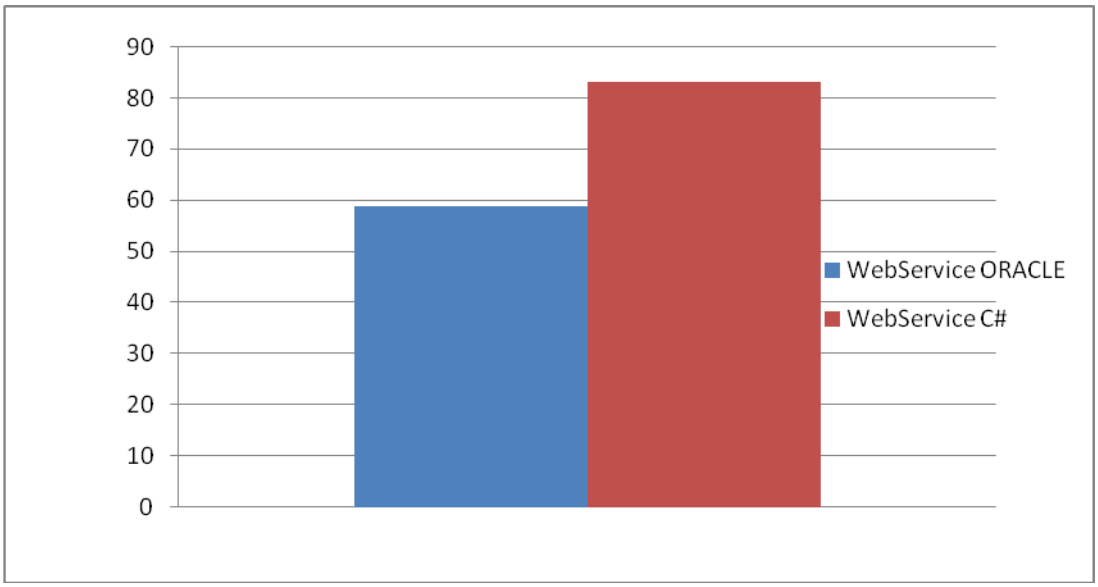

Figura 18: Comparação do tempo de resposta do Web Service Oracle e C# em segundos com 20000 repetições.

No primeiro teste com vinte mil chamadas, novamente o WS Oracle, retornou mais rápido, em um total de 58 segundos, enquanto WS C# retornou em 82 segundos.

Comparativamente, em relação ao teste com dez mil chamadas, em que a diferença ficou em aproximadamente 5 segundos, nesse caso a diferença ficou em 24 segundos, percebe-se que ocorreu um aumento significativo na diferença em segundos.

O experimento com carga de vinte mil repetições, cinco vezes para sua analise estatística. Pode-se observar que o WS Oracle novamente, em todas as amostras, retornou mais rápido que o WS em C#, conforme mostra a Figura 19.

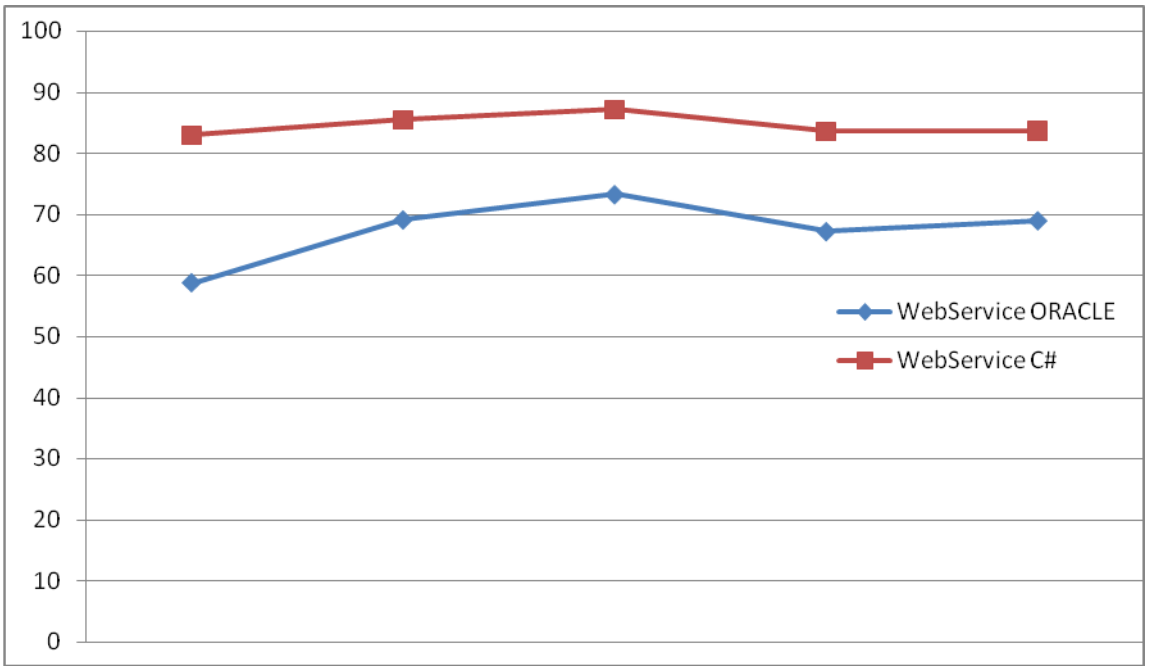

Figura 19: Replicação do experimento com carga de 20000 repetições.

Neste caso o consumo do WS em Oracle teve uma média das amostras em 67 segundos, enquanto o C# em 84 segundos e a média da diferença em 17 segundos.

Para verificar, se com um número mais elevado ainda de requisições os resultados continuam seguindo os mesmos padrões, um teste foi realizado, com cem mil repetições.

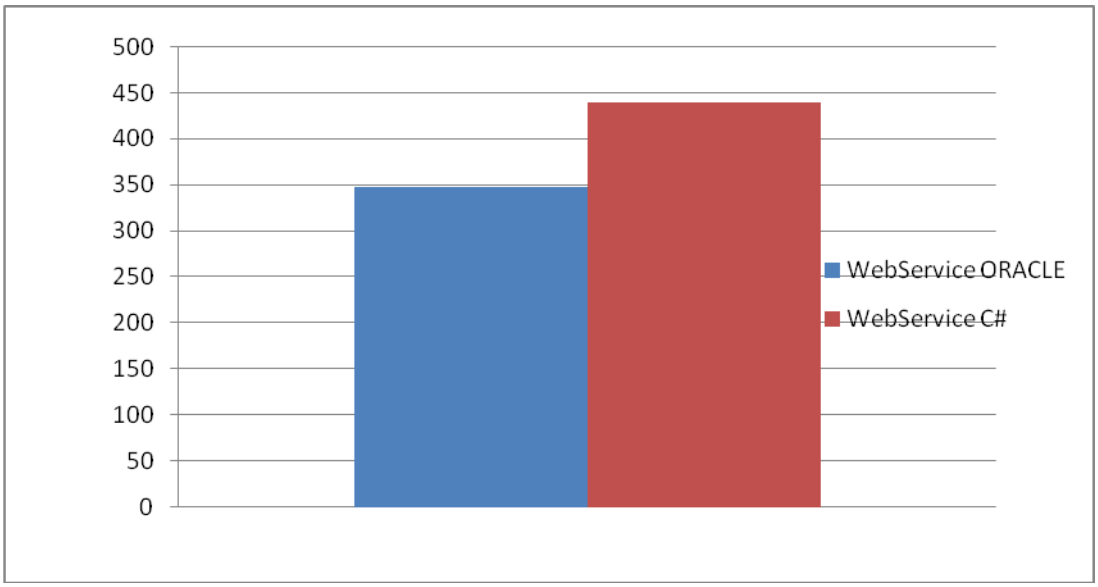

Figura 20: Comparação do tempo de resposta do Web Service Oracle e C# em segundos com 100000 repetições.

Note que, conforme a Figura 20, na chamada com cem mil requisições, o WS Oracle novamente retornou mais rápido, com uma diferença de 131 segundos, ou 23 % mais rápido do que o WS C#.

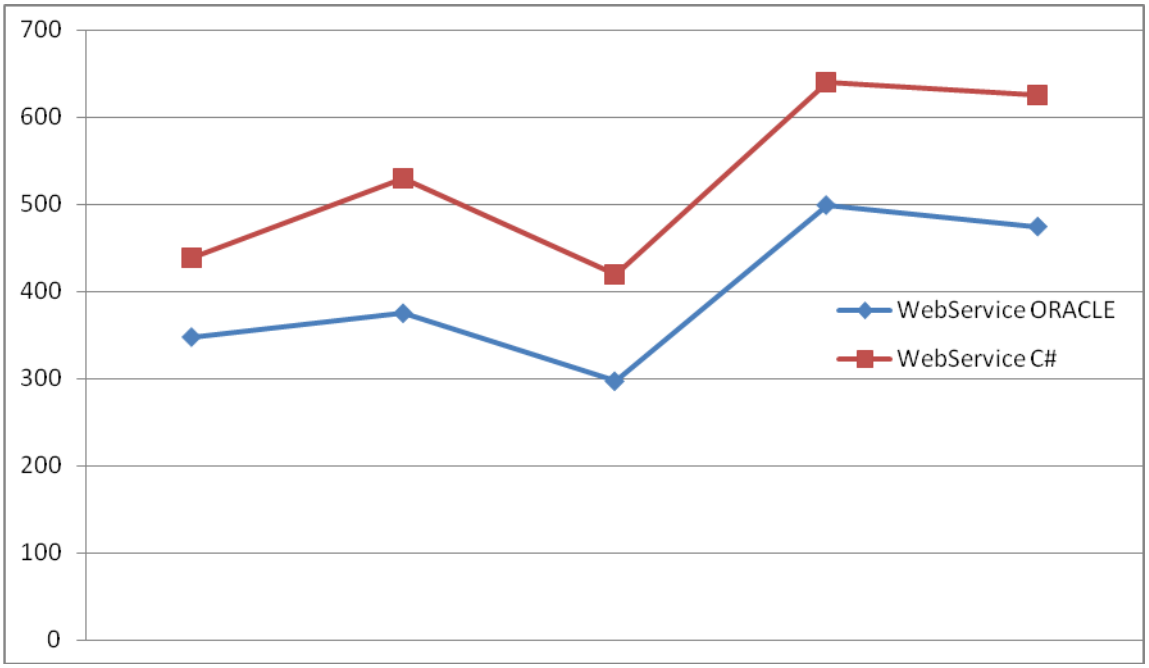

Figura 21: Replicação do experimento com carga de 100000 repetições.

Observou uma variação no uso processamento da máquina, durante os testes realizados com cem mil repetições, porém a diferenças permaneceram semelhantes.

Para fins comparativos, a Figura 22 ilustra as diferenças entre os tempos de respostas coletados em cada tecnologia.

Pode-se notar que o aumento do da diferença em segundos é crescente, de acordo com o aumento da carga.

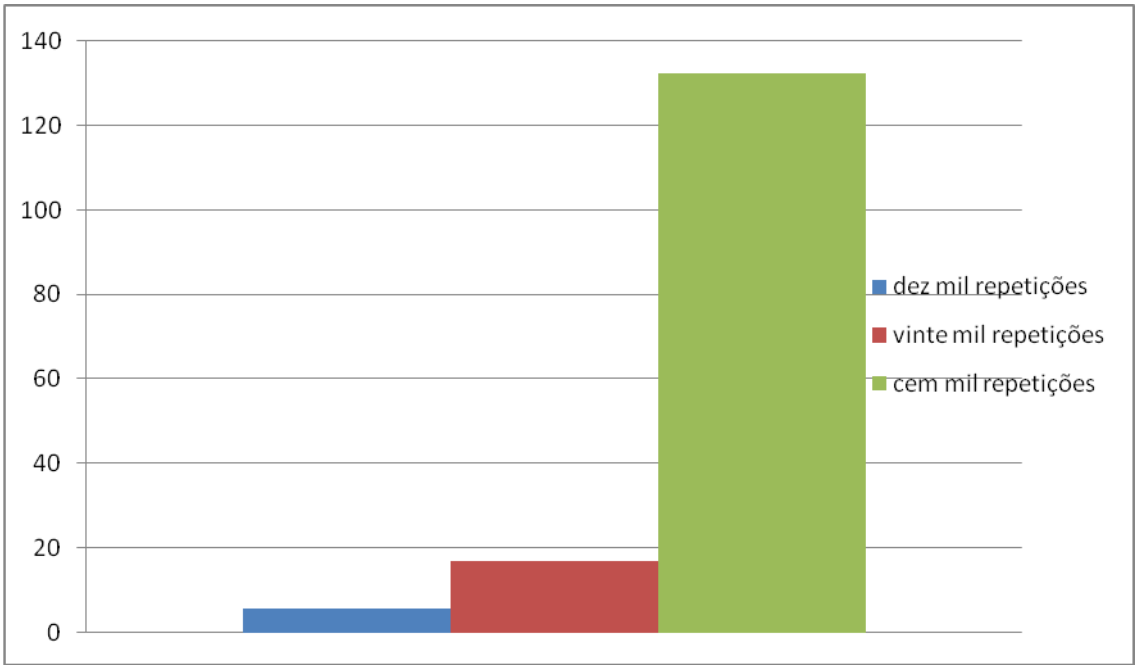

Figura 22: Diferença em segundos da média das amostras.

Note que, com o aumento da carga, o WS em C# apresenta uma degradação de desempenho muito mais acentuada do que o WS em Oracle, a diferença em segundos aumenta, em proporção ao número de repetições.

Outras formas de consumo devem ser levadas em consideração, com cargas maiores de dados ou novas condições, dependendo da necessidade de cada organização.

Nota-se que o tempo de execução pode variar, dependendo do uso do CPU no momento ou da capacidade de processamento da máquina, porém em todos os momentos, a diferença se manteve similares.

Em um ambiente organizacional o desempenho é um fator determinante, para a competitividade de mercado. Assim, estima-se que esse trabalho contribui com estimativas que podem ser decisivas para a tomada de decisões organizacionais, especialmente no que tange a alocação de recursos computacionais, dentre outros.

5 CONCLUSÃO

A implementação de um WS com Oracle PLSQL, conta com várias vantagens em relação a aplicações convencionais implementadas através de linguagens de programações. Diminui a carga de trabalho, os métodos existentes no banco de dados, automaticamente passam a serem publicados na web, não sendo necessário desenvolver uma chamada para estes métodos. Outra questão é a herança da lógica de aplicações baseada em permissões de usuários, sendo que cada método ao ser acessado poderá receber por parâmetro, as credenciais do usuário, limitando assim somente acessar métodos e aplicações que o usuário tenha acesso. O consumo de um WS em Oracle PLSQL, pode ser acessado da mesma forma através das linguagens de programação devido à portabilidade do arquivo WSDL.

O comparativo de desempenho, no consumo de um WS, implementado em C# e um em Oracle, chamando o mesmo método nas mesmas configurações de máquina. O primeiro teste de desempenho, com uma interação, não apresentou diferença significativa de desempenho, nos testes posteriores, com um número elevado de requisições, o WS Oracle retornou mais rápido, em todos os casos. A aplicação futuramente pode ser utilizada para testes, a fim de fazer novas comparações, em outra amostra ou em outras condições, com cargas maiores de dados.

Os conceitos avaliados no presente trabalho, considerando a necessidade, de aplicações, acessáveis na web, surgem diversas questões, em relação às formas de implementação e suas tecnologias. O surgimento de tecnologias emergentes, como o OData, onde um banco de dados e seus métodos podem ser acessados através de um WS, retornando um modelo orientado a objetos, pode alterar significativamente as escolhas de tecnologias a serem usadas. Assim, aconselha-se que os experimentos conduzidos nesse trabalho sejam reproduzidos para cada cenário, conforme os recursos a serem utilizados.

REFERÊNCIAS

DATE, C. J. **Banco de Dados: Fundamentos.** Rio de Janeiro: Campus, 1985.

DATE, C. J. **Introdução a sistemas de banco de dados.** Rio de janeiro: Elsevier 2004.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas, Organização e Métodos: Uma Abordagem Gerencial**. 19ª ed. São Paulo: Editora Atlas, 2010.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron, 1999.

TAKAI, Osvaldo Kotaro; ITALIANO, Isabel Cristina; FERREIRA, João Eduardo**. Introdução a banco de dados.** 1995. Disponível em: <http://www.ime.usp.br/~jef/apostila.pdf>. Acesso em: 4 abr. 2013.

ELMASRI, Ramez; NAVATHE, Shamkan B. **Sistema de banco de dados.** 6.ed São Paulo: Pearson 2011.

WATSON, John. **OCA Oracle Database 11g Administração I.** São Paulo: Artmed 2010.

MENSAH, Kuassi; ROHWEDDER, Ekkehard. **Database Web Services.** Redwood Shores: Oracle Corporation 2002.

W3C. **Extensible Markup Language (XML).** Disponível em <http://www.w3.org/XML> Acessado em 29/08/2016.

Web Services Description Language. Disponível em <http://www.w3.org/TR/wsdl> Acessado em 29/08/2016.

SOAP Specifications. Disponível em <http://www.w3.org/TR/soap/> Acessado em 29/08/2016.

Microsoft. **Introduction to the C# Language and the .NET Framework.** Acessado em 30/08/2016.

TABOR, Robert. **Microsoft .NET XML WEB SERVICES.** Indianápolis: Sams, 2002.