Agência Nacional
de Vigilância Sanitária

Projeto SNGPC para Farmácias e Drogarias

Conexão ao Webservice
Manual do Desenvolvedor

Versão 2.0.1

Brasília, outubro de 2013.

Diretor-Presidente
Dirceu Brás Aparecido Barbano

Diretores
José Agenor Álvares da Silva
Jaime César de Moura Oliveira
Ivo Bucaresky
Renato Alencar Porto

Gerência Geral de Tecnologia da Informação – GGTIN
Igor Ticchetti Kishi

Núcleo de Gestão do Sistema Nacional de Notificação e Investigação em Vigilância Sanitaria - NUVIG
Maria Eugênia Carvalhaes Cury

Coordenação do Sistema Nacional de Gerenciamento de Produtos Controlados/CSGPC
Rafael Filiacci Bovi

Elaboração
Verangge Pereira Lopes Custódio – GGTIN

Sumário

1. OBJETIVO **4**
2. ACESSIBILIDADE: **4**
2.1 ACESSO AO SERVIÇO: **4**
2.2 FUNCIONALIDADES (HOMOLOGAÇÃO E PRODUÇÃO) **7**
2.3 TESTANDO OS MÉTODOS **13**

1. OBJETIVO

O serviço W*eb Service* foi desenvolvido para que os estabelecimentos que utilizam o Sistema Nacional de Gerenciamento de Produtos Controlados – SNGPC possam enviar o arquivo XML para a base de dados da Anvisa, utilizando o padrão estabelecido pela Anvisa.

2. ACESSIBILIDADE:

Caso os estabelecimentos não queiram transmitir o XML pela página de *Upload* disponibilizada pelo site da Anvisa, o sistema do estabelecimento deverá acessar ao *webservice* pelo endereço remoto, utilizando o método de transmissão.

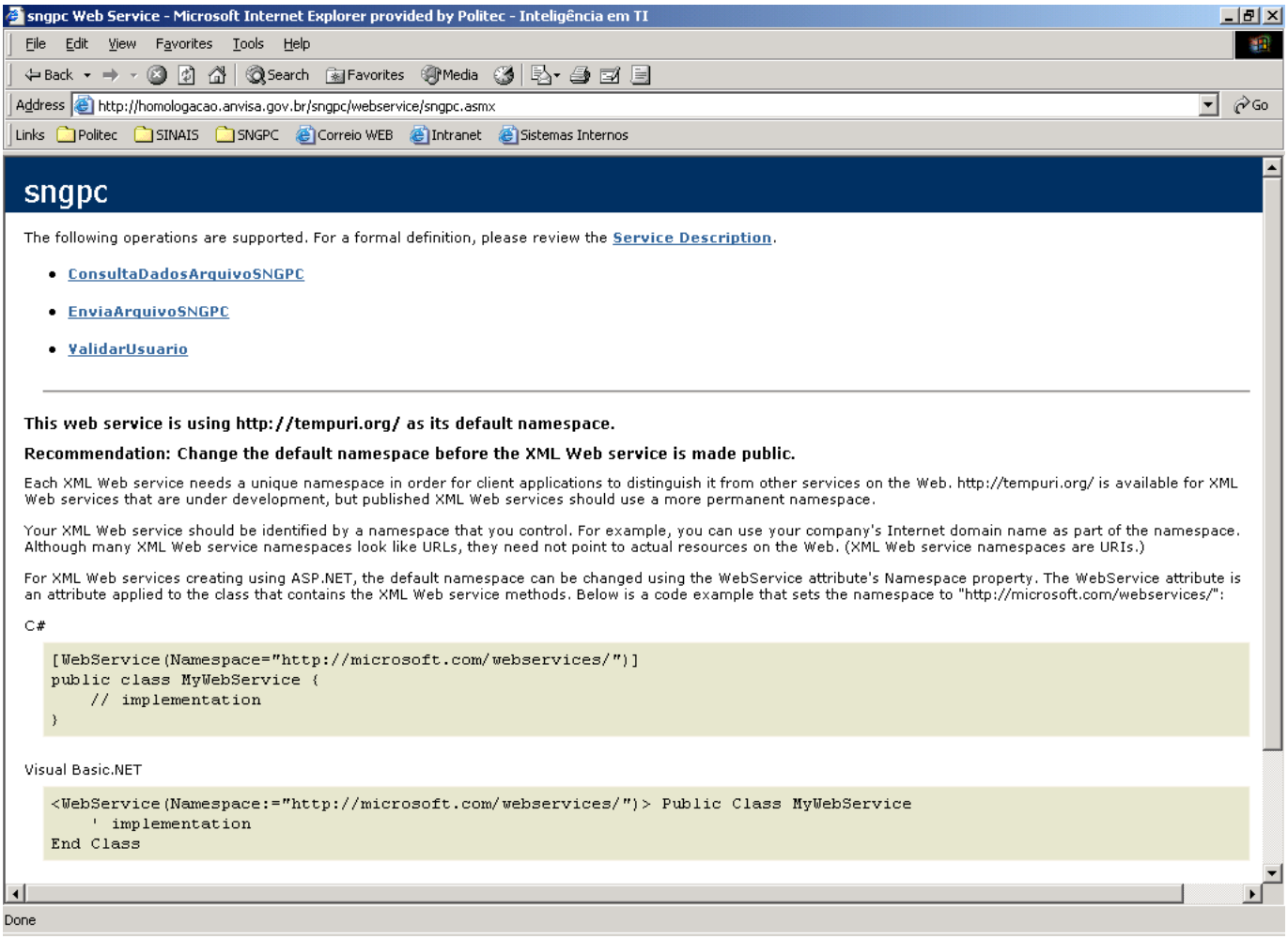
Para a versão 2.0 do sistema, há poucas mudanças no processo de transmissão do XML. O conteúdo a ser transmitido continua sendo apenas texto; porém, o arquivo XML precisa ser compactado em formato '*zip'* e depois convertido em '*base64'*.

Base64 é um método para codificação de dados para transferência na Internet. É utilizado para transferência de dados binários por meios de transmissões que lidam apenas com texto, como por exemplo, para enviar arquivos anexos por *email* ou métodos de *WebServices*.

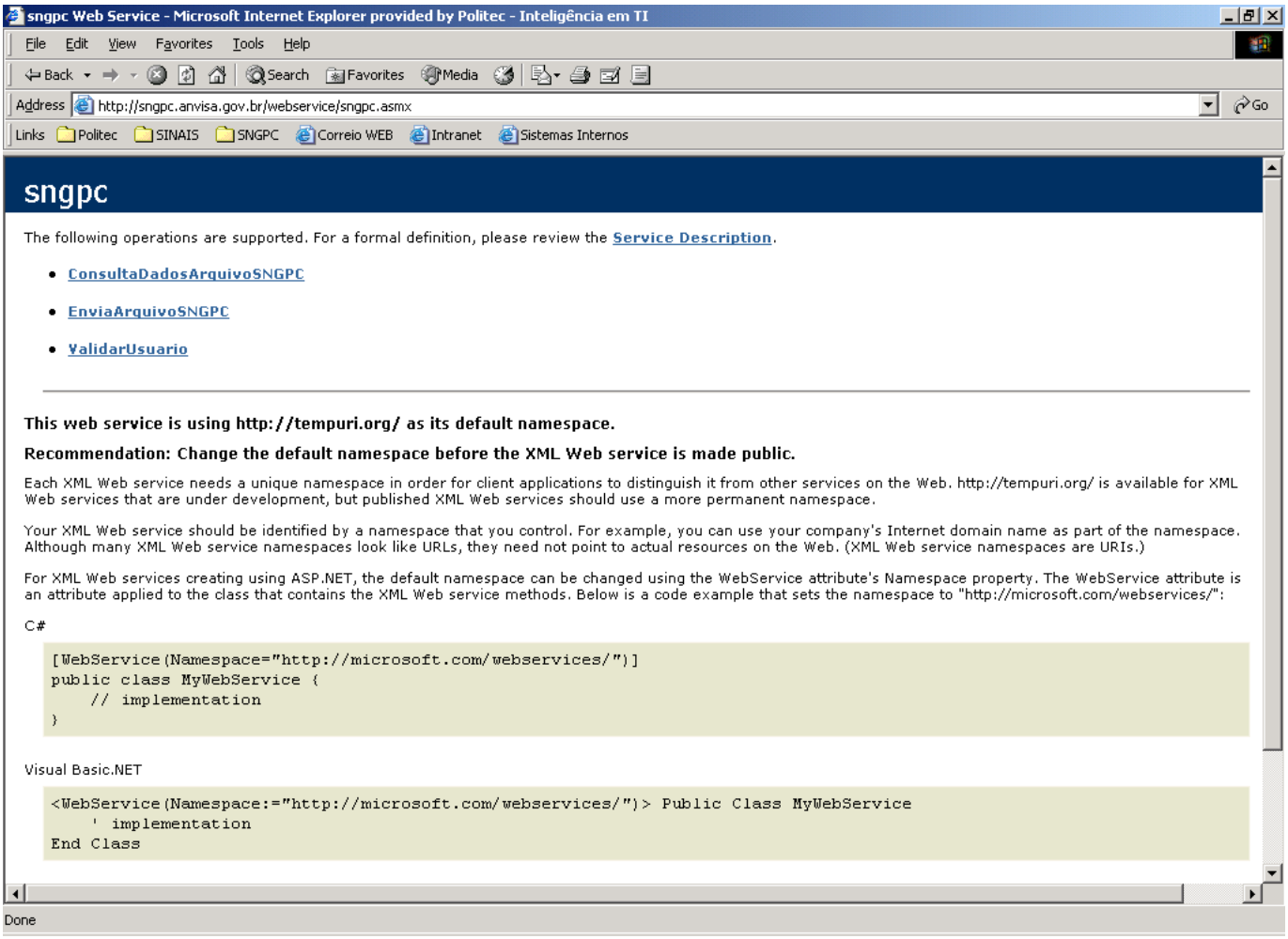
No momento da transmissão do arquivo, o SNGPC irá calcular o MD5 sobre o valor da base64 do arquivo compactado. Este valor é usado para simples comparação, um valor diferente calculado pelo usuário, não impedirá a recepção do arquivo, mas é importante ser armazenado para futura consulta a situação do arquivo.

2.1 Acesso ao serviço:

- Homologação : http://homologacao.anvisa.gov.br/sngpc/webservice/sngpc.asmx



- Produção: http://sngpc.anvisa.gov.br/webservice/sngpc.asmx



Para acessar o WSDL, ou seja, a descrição dos métodos, clique no link *Service Description*.

2.2 Métodos (Homologação e Produção)

2.2.1 ValidarUsuario

Esta funcionalidade permite a validação dos dados do usuário denominado RT Transmissor do SNGPC, são validados o e-mail do usuário e senha.

- Parâmetros:
  - *Email: string*
  - *Senha: string*

2.2.2 EnviaArquivoSNGPC

Estas funcionalidades têm por finalidade permitir a transmissão do arquivo XML selecionado pelo usuário, validar a estrutura do XML que deve estar de acordo com os *schemas* definidos e gravar o arquivo em tabela para ser, posteriormente, gravado na base de dados do SNGPC.

- Parâmetros:
  - *Email: string*
  - *Senha: string*
  - *Arq: string (XML) compactada e convertida em base64*
  - *Hashindenficacacao: string (32 caracteres)*

sngpc

Click here for a complete list of operations.

EnviaArquivoSNGPC

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /sngpc/webservice/sngpc.asmx HTTP/1.1
Host: homologacao.anvisa.gov.br
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/EnviaArquivoSNGPC"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <EnviaArquivoSNGPC xmlns="http://tempuri.org/">
      <Email>string</Email>
      <Senha>string</Senha>
      <Arq>base64Binary</Arq>
      <HashIdentificacao>string</HashIdentificacao>
    </EnviaArquivoSNGPC>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <EnviaArquivoSNGPCResponse xmlns="http://tempuri.org/">
      <EnviaArquivoSNGPCResult>string</EnviaArquivoSNGPCResult>
    </EnviaArquivoSNGPCResponse>
  </soap:Body>
</soap:Envelope>
```

- Retorno:
  - *O retorno do método é do tipo string. Caso o envio seja concluído com sucesso o método irá retornar a seguinte mensagem "***Arquivo recebido com sucesso, em DD/MM/YYY, ás HH:MM:SS. O Hash calculado foi '*hash32caracteres*'***", caso contrário o retorno será uma mensagem com o erro ocorrido.*

- Exemplo:
  - *A forma como o webservice é instanciado depende da linguagem de programação utilizada.*

  - *Segue 2 exemplos: .NET e Delphi:*
    - *Visual Studio .Net*
      - *Web Reference: Adicionar uma web reference com o endereço do webservice (ver item 2.1);*
      - *Instância: Após a web reference criada, com o nome dado, criar uma instancia do webservice na aplicação.*

        sngpcService.sngpc wssngpc = new sngpcService.sngpc();
      - *Métodos: Para visualizar os métodos, criar uma variável string para receber o retorno do método.*

        string comp;
        comp =
        wssngpc.EnviaArquivoSNGPC(edemail.Value.ToLower(),
        edsenha.Text, Arquivo, HashIdentificacao);
        HashIdentificacao = Rotina MD5;

    - *Delphi*
      - *WSDL: Utilizar o Import WSDL com o endereço do webservice (ver item 2.1) para adicionar ao projeto a classe do webservice.*
      - *Instância: Após adicionar a classe ao projeto, criar uma variável com o nome da interface.*

        var ws: wssinaisSoap;
        ws := GetwssinaisSoap();
      - *Métodos: Para visualizar os métodos, criar uma variável string para receber o retorno do método.*

        string comp;

comp:= ***ws***.**EnviaArquivoSNGPC(email.ToLower(),**
edsenha.Text, Arquivo, HashIdentificacao);
HashIdentificacao = Guid();

- *HashIdentificação - Rotina MD5*

```
private String geraHash(String Arquivo)
{
        String HashGerado;
        ASCIIEncoding textConverter = new ASCIIEncoding();
        MD5CryptoServiceProvider Md5Provider = new
MD5CryptoServiceProvider();
        Byte[] ArquivoByte;
        Arquivo = Arquivo.Replace("\r","").Replace("\n","").Replace("\t","");
        ArquivoByte = textConverter.GetBytes(Arquivo);
        ArquivoByte = Md5Provider.ComputeHash(ArquivoByte);
        HashGerado = ToHexString(ArquivoByte);
        return HashGerado;
}
```

2.2.3 ConsultaDadosArquivoSNGPC

Esta funcionalidade verifica o arquivo transmitido para o SNGPC, retornando informações quanto este tiver sido validado pelo sistema.

- Parâmetros:
  - *Email: string*
  - *Senha: string*
  - *CNPJ: string*
  - *Hash: string (32 caracteres)*

sngpc

Click here for a complete list of operations.

ConsultaDadosArquivoSNGPC

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /sngpc/webservice/sngpc.asmx HTTP/1.1
Host: homologacao.anvisa.gov.br
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/ConsultaDadosArquivoSNGPC"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConsultaDadosArquivoSNGPC xmlns="http://tempuri.org/">
      <Email>string</Email>
      <Senha>string</Senha>
      <CNPJ>string</CNPJ>
      <Hash>string</Hash>
    </ConsultaDadosArquivoSNGPC>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConsultaDadosArquivoSNGPCResponse xmlns="http://tempuri.org/">
      <ConsultaDadosArquivoSNGPCResult>string</ConsultaDadosArquivoSNGPCResult>
    </ConsultaDadosArquivoSNGPCResponse>
  </soap:Body>
</soap:Envelope>
```

- Retorno:
  - *O retorno do método é do tipo string.*

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://tempuri.org/"><transmissaoSNGPC> <cabecalho>
  <CODIGOHASH>ba563f1388692720851fa813d3f7d0f8</CODIGOHASH>
  <DATATRANSMISSAO>13/11/2007</DATATRANSMISSAO> <DATAVALIDACAO>13/11/2007</DATAVALIDACAO>
  <INICIOREFERENCIA>10/11/2007</INICIOREFERENCIA> <FIMREFERENCIA>10/11/2007</FIMREFERENCIA>
  </cabecalho></transmissaoSNGPC></string>
```

- Chamada:
  - *A forma como o webservice é instanciado depende da linguagem de programação utilizada.*
  - *Segue exemplo em Delphi:*

- *Delphi*
  - *WSDL: Utilizar o Import WSDL com o endereço do webservice para adicionar ao projeto a classe do webservice.*
  - *Instância: Após adicionar a classe ao projeto, criar uma variável com o nome da interface.*
    var ws: wssinaisSoap;
    ws := GetwssinaisSoap();
  - *Métodos: Para visualizar os métodos, criar uma variável string para receber o retorno do método.*
    string comp;
    comp:= ***ws***. **ConsultaDadosArquivoSNGPC (edEmail.Text,edSenha.Text,edCNPJ.Text,edHash.Text);**

2.2.4 Exemplo em java:

Abaixo há um pequeno exemplo implementado em JAVA que demonstra como comprimir e codificar os arquivos em base64.

```java
package br.gov.anvisa.util.zip;


import java.io.ByteArrayOutputStream;

import java.util.zip.Deflater;
import java.util.zip.DeflaterOutputStream;
import java.util.zip.Inflater;
import java.util.zip.InflaterOutputStream;

import org.apache.commons.codec.binary.Base64;
import org.apache.log4j.Logger;


/**
 * Esta classe fornece suporte para compressão de propósito geral, utilizando a biblioteca de compressão popular ZLIB. A
 * biblioteca de compressão ZLIB foi inicialmente desenvolvido como parte dos gráficos PNG padrão e não está protegida
 * por patentes.
 *
 * Além da compressão, também realiza codificação em base 64 para que os dados comprimidos possam ser utilizados como
 * sequência de caracteres, por exemplo, em um arquivo XML. Isso é realizado com o uso da API Apache Commons Codec (TM),
 * que implementa seção 6.8 da Base64 Content-Transfer-Encoding from RFC 2045 Multipurpose Internet Mail Extensions
 * (MIME) Part One: Format of Internet Message Bodies by Freed and Borenstein.
 *
 * @author ANVISA - Agência Nacional de Vigilância Sanitária
 * @version 2.0
 * @see org.apache.commons.codec.binary.Base64
 */
public class CompressionUtil {
    private static final Logger logger = Logger.getLogger(CompressionUtil.class);

    /**
     * Comprimi o texto fornecido em formato ZLIB com compressao máxima e codifica os {@code bytes} em base 64.
     *
     * @param text O texto a ser comprimido.
     * @return O texto fornecido comprimido e codificado em base 64.
     */
    public static String encodeBase64Zlib(String text) {
        try {
            Deflater def = new Deflater(Deflater.BEST_COMPRESSION);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            DeflaterOutputStream dos = new DeflaterOutputStream(baos, def, 4 * 1024);

            dos.write(text.getBytes());
            dos.flush();
            dos.close();

            return Base64.encodeBase64String(baos.toByteArray());
        } catch (Exception e) {
            if (logger.isDebugEnabled()) {
                logger.fatal(e.getMessage(), e);
            }

            throw new RuntimeException(e);
        }
    }

    /**
     * Decodifica o texto em base 64 fornecido e interpreta seu conteudo como um arquivo compactado em ZLIB. O conteúdo
     * desse arquivo é descompactado e o texto resultante é retornado.
     *
     * @param base64ZlibText O texto em base 64 a ser descompactado.
     * @return O texto descompactado.
     */
    public static String decodeBase64Zlib(String base64ZlibText) {
        try {
            Inflater inf = new Inflater();
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            InflaterOutputStream ios = new InflaterOutputStream(baos, inf, 4 * 1024);

            ios.write(Base64.decodeBase64(base64ZlibText.getBytes()));
            ios.flush();
            ios.close();

            return baos.toString();
        } catch (Exception e) {
            if (logger.isDebugEnabled()) {
                logger.fatal(e.getMessage(), e);
            }

            throw new RuntimeException(e);
        }
    }
}
```

Abaixo há um teste simples para a classe exemplo.

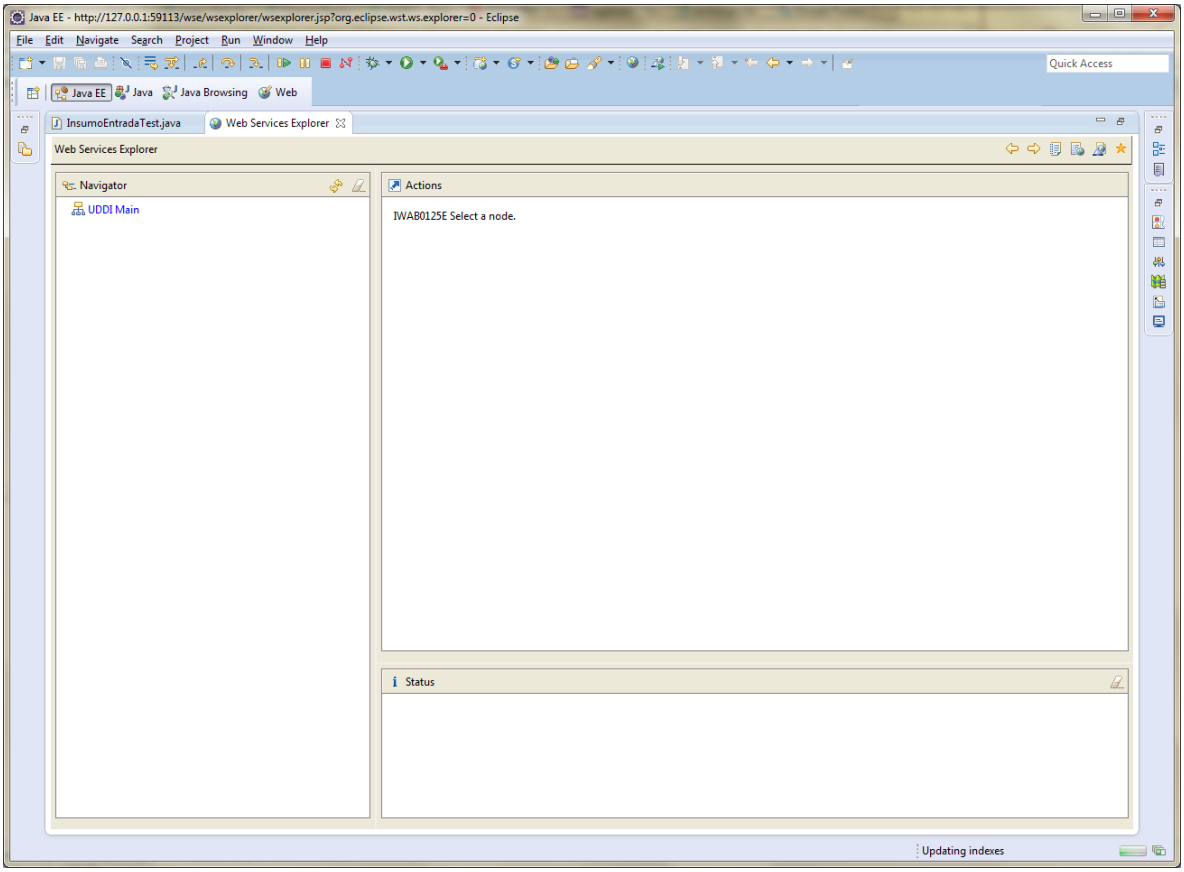
```
package br.gov.anvisa.util.zip;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 * Testes unitários para a classe CompressionUtil.
 *
 * @author ANVISA - Agência Nacional de Vigilância Sanitária
 * @since 2.0
 * @see CompressionUtil
 */
public class CompressionUtilTest {

    private static final String text = "Testando a classe CompressionUtil:\\n  - String 1;\\n  - String 2.";
    private static final String base64ZlibText = "eNoLSS0uScxLyVdIVEjOSSwuTlVwzs8tKEotLs7MzwstycyxislTUNBVCC4pysxLVzC0RuEa6QEA6NkVkQ=="

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testEncodeBase64Zlib() {
        System.out.println("###>>> Testando encodeBase64Zlib...");
        String result = CompressionUtil.encodeBase64Zlib(text);
        assertEquals(base64ZlibText, result);
    }

    @Test
    public void testDecodeBase64Zlib() {
        System.out.println("###>>> Testando decodeBase64Zlib...");
        String result = CompressionUtil.decodeBase64Zlib(base64ZlibText);
        assertEquals(text, result);
    }
}
```

2.3 Testando os métodos do *Web Service*

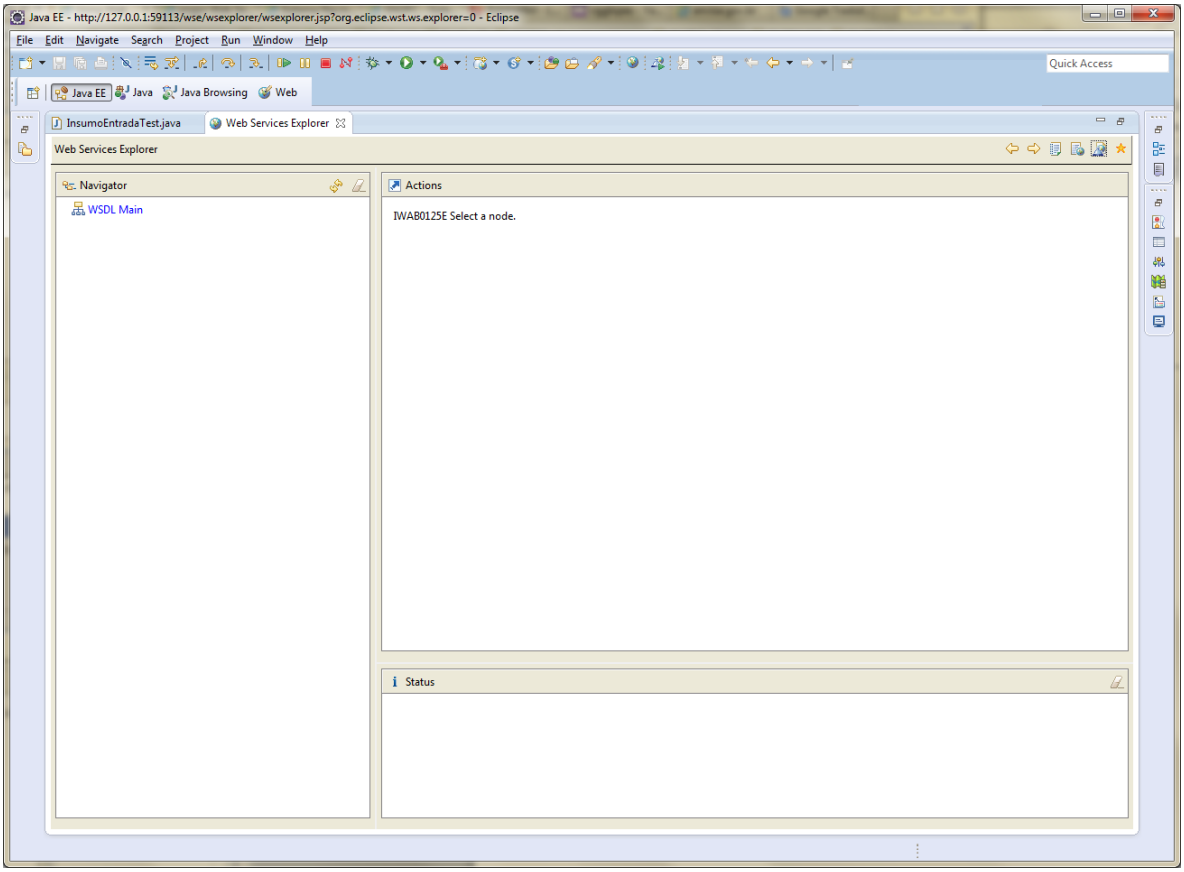
Para testar os métodos do *Web Service* do SNGPC, sugerimos utilizar a ferramenta denominada *Web Services Explorer* por meio da plataforma Eclipse (www.eclipse.org). Sugerimos o uso desta ferramenta por ser gratuita e de fácil entendimento, porém há várias outras disponíveis gratuitamente, ou não, na Internet.

O passo a passo abaixo demostra como usar o *Web Service Explorer* para testar um Web Service via WSDL nativo e SOAP. Ele demonstra como usar a *Web Services Explorer* para invocar os métodos do *Web Service* do SNGPC chamados 'ValidarUsuario', 'EnviaArquivoSNGPC' e 'ConsultaDadosArquivoSNGPC'.

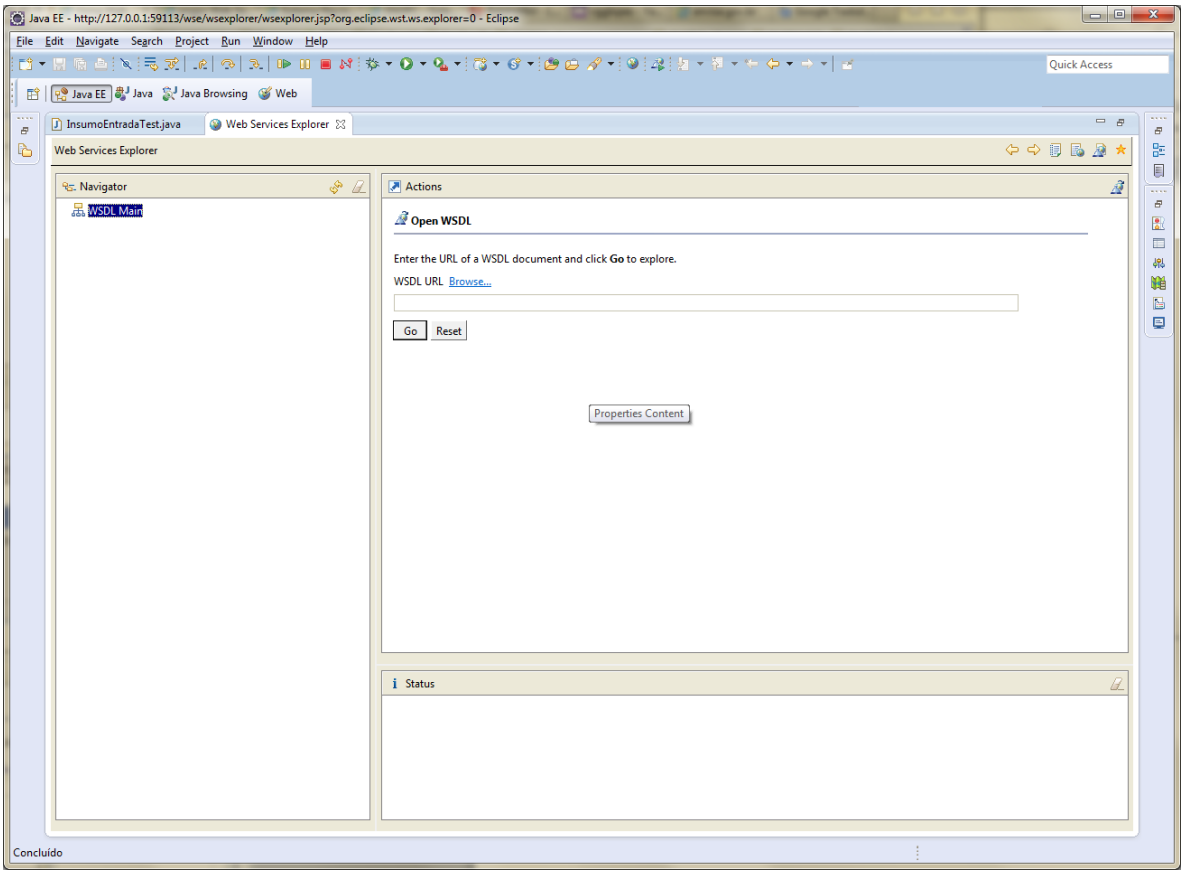
1. Inicie o Eclipse (utilizamos a versão Eclipse Java EE IDE for Web Developers. Version: Juno Service Release 1);
2. Na barra de menu, selecione **Run** -> **Launch the Web Services Explorer;**
3. Após o Web Browser ser aberto, maximize a tela, da seguinte forma**:**



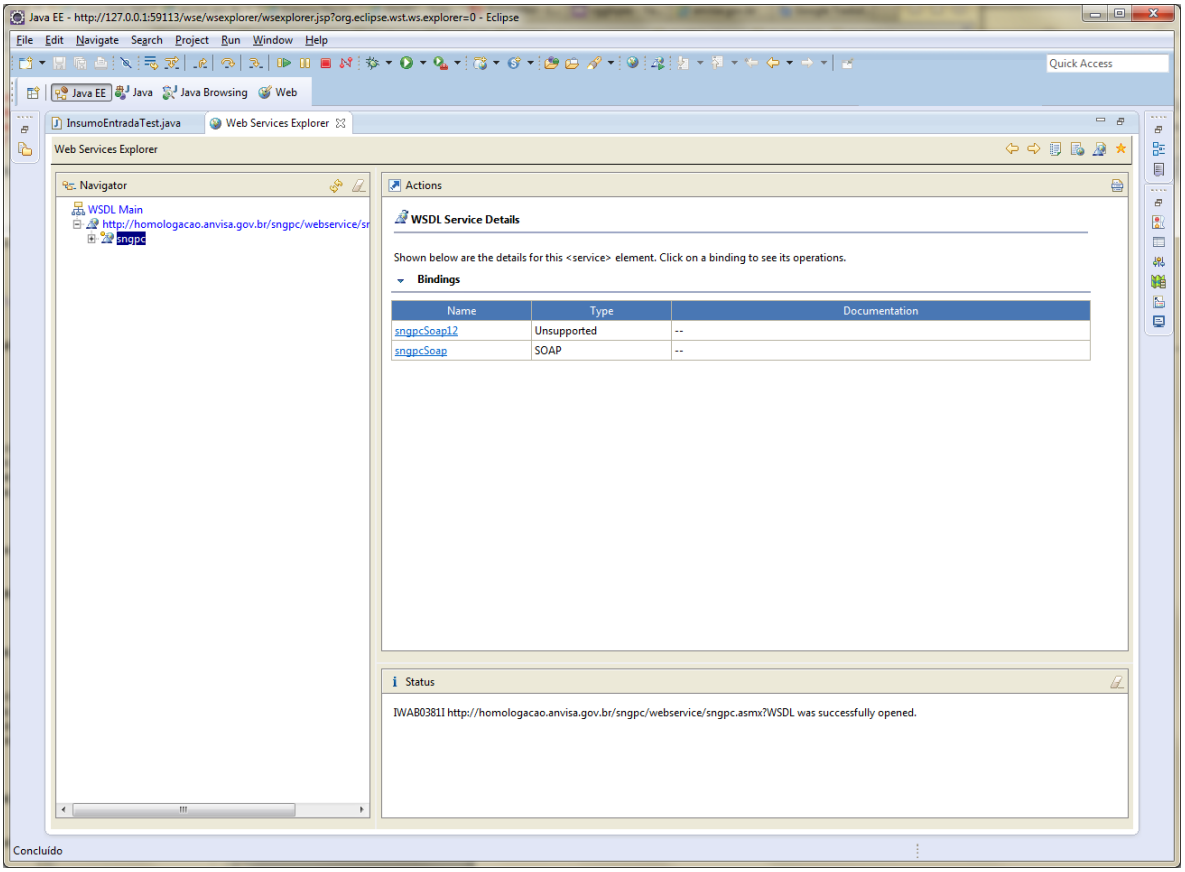
4. No canto superior direito da barra de menu do Web Service Explorer, click no ícone do WSDL PAGE . Resultado:



5. Clique no ícone do **WSDL Main** . Resultado:

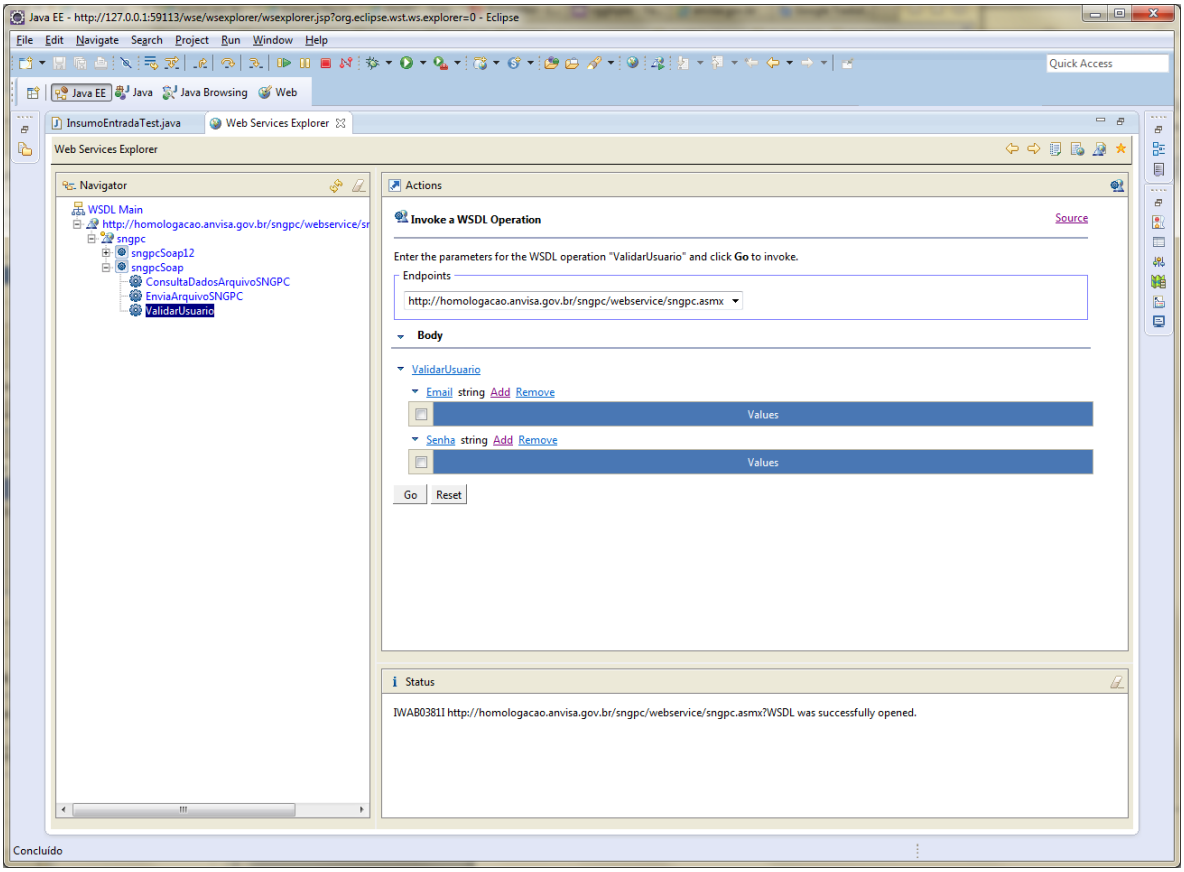


6. No campo da **WSDL URL**, digite a url do Web Service de homologação do SNGPC. http://homologacao.anvisa.gov.br/sngpc/webservice/sngpc.asmx?WSDL, depois clique em **Go**. Resultado:

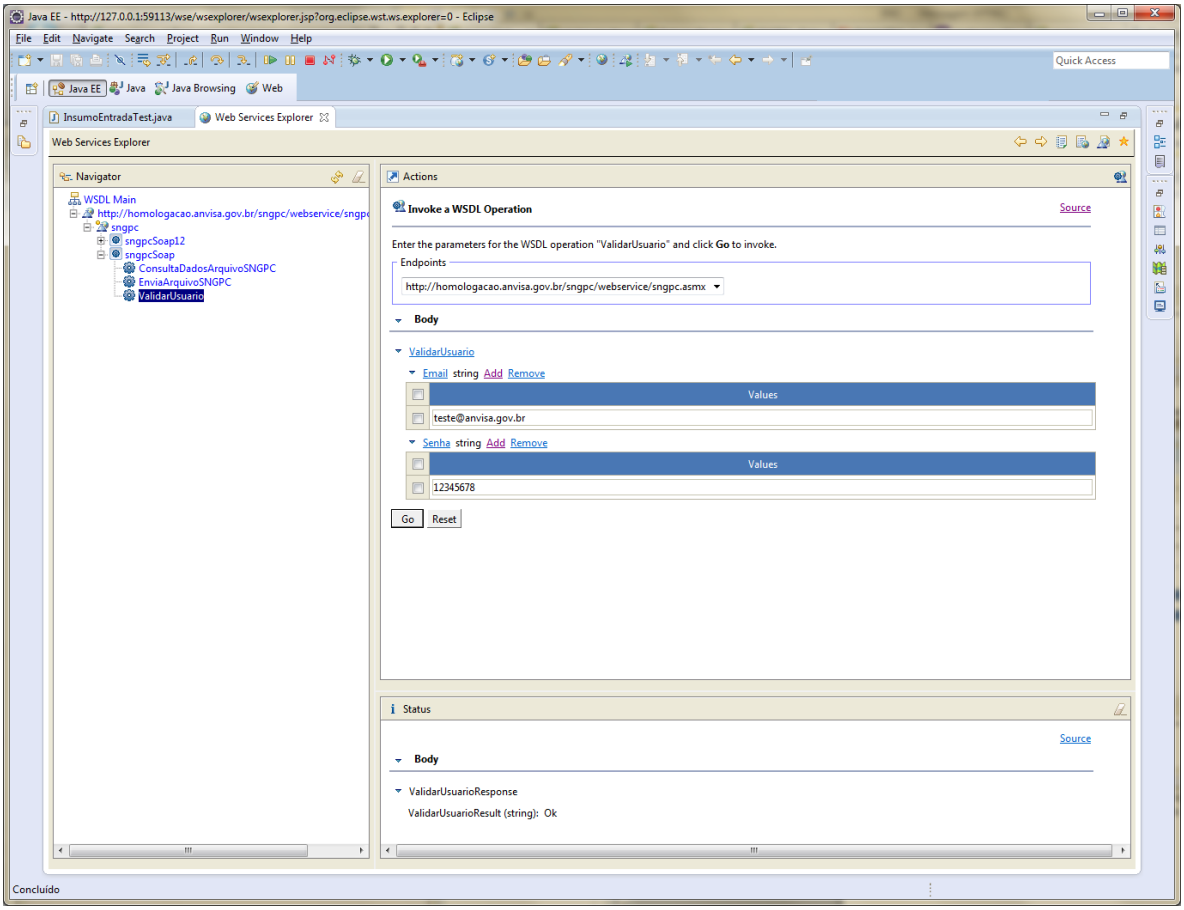


7. Selecione a biblioteca sngpcSoap;

8. Selecione uma das operações para testar, por exemplo, 'ValidarUsuario', será exibida a tela abaixo:



9. Nos campos email e senha, clique em Add para que seja exibido o campo para inserção e informe os dados válidos no ambiente de homologação. Em seguida, clique em ***GO*** para envio e teste. No painel de status, será exibido o retorno do Web Service:



Para testar as demais informações, repetir os passos 8 e 9, selecionando a operação que deseja testar.

3. CONSIDERAÇÕES FINAIS

Esta versão está sendo publicada visando auxiliar os desenvolvedores dos sistemas utilizados pelas farmácias e drogarias a se comunicarem com o SNGPC. As contribuições devem ser enviadas através do formulário fale conosco do site da Anvisa ou pela central de atendimento – 0800-6429782.

O presente documento será atualizado conforme necessidades identificadas pela autoridade sanitária