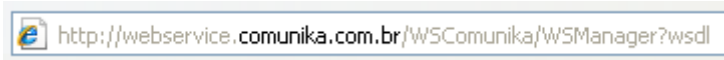


Esse arquivo descreve o que fiz para funcionar as chamadas dos dois serviços da Comunika, obedecendo-se a mesma seqüência desse arquivo.

Identificando a URL do provedor e o contexto

Acessei a primeira WSDL do manual dos web services da Comunika:



Identifiquei a URL do provedor (é o texto até a primeira barra “/”):

http://webservice.comunika.com.br

Identifiquei o contexto (é tudo o que está entre a URL do provedor e o nome do serviço):

WSComunika

Configurações do provedor de serviços

(usar proxy apenas se necessário - na Senior é necessário)

The image shows a configuration window titled "Comunika". It contains several sections for setting up web services:

- Tipo do provedor:** Three radio buttons: "Provedor de serviços Senior" (unchecked), "Provedor de serviços de terceiro" (checked), and "Mesma base" (unchecked).
- Nome do provedor:** A text box containing "Comunika".
- Timeout:** A text box containing "30".
- URL do servidor:** A text box containing "http://webservice.comunika.com.br".
- Contexto:** A text box containing "WSComunika".
- Módulo de dispatch da chamada:** An empty text box.
- Configurações de proxy:**
 - Identificador do servidor:** A text box containing "10.0.10.6".
 - Porta:** A text box containing "3128".
 - Usuário:** A text box containing "elizandro.schramm".
 - Senha:** A text box containing "*****".
- Usuário para execução dos serviços:**
 - Three checkboxes: "Utilizar autenticação HTTP" (unchecked), "Utilizar autenticação SOAP" (unchecked), and "Utilizar o usuário logado no sistema" (unchecked).
 - Usuário:** An empty text box.
 - Senha:** An empty text box.
 - A button labeled "Cabeçalho..." is located below the user and password fields.

At the bottom of the window are three buttons: "Avançar >>", "Ok", and "Cancelar".

Entendendo a porta getAuthentication do serviço WSMManager

WSDL:

The screenshot shows a web browser displaying a WSDL file from `http://webservice.comunika.com.br/WSComunika/WSManager?wsdl`. The XML content is as follows:

```
<?xml version='1.0' encoding='utf-8'?>
<types>
  <xsd:schema>
    <xsd:import namespace="http://webservice.comunika.com.br" schemaLocation="http://webservice.comunika.com.br:80/WSComunika/WSManager?xsd=1" />
  </xsd:schema>
</types>
<message name="getAuthentication">
  <part name="parameters" element="tns:getAuthentication" />
</message>
<message name="getAuthenticationResponse">
  <part name="parameters" element="tns:getAuthenticationResponse" />
</message>
<message name="cancelScheduleMessageByBatch">
  <part name="parameters" element="tns:cancelScheduleMessageByBatch" />
</message>
<message name="cancelScheduleMessageByBatchResponse">
  <part name="parameters" element="tns:cancelScheduleMessageByBatchResponse" />
</message>
<message name="cancelScheduleMessageByPeriod">
  <part name="parameters" element="tns:cancelScheduleMessageByPeriod" />
</message>
<message name="cancelScheduleMessageByPeriodResponse">
  <part name="parameters" element="tns:cancelScheduleMessageByPeriodResponse" />
</message>
<message name="cancelScheduleMessageByIds">
  <part name="parameters" element="tns:cancelScheduleMessageByIds" />
</message>
<message name="cancelScheduleMessageByIdsResponse">
  <part name="parameters" element="tns:cancelScheduleMessageByIdsResponse" />
</message>
<portType name="WSManager">
  <operation name="getAuthentication">
    <input message="tns:getAuthentication" />
    <output message="tns:getAuthenticationResponse" />
  </operation>
  <operation name="cancelScheduleMessageByBatch">
    <input message="tns:cancelScheduleMessageByBatch" />
    <output message="tns:cancelScheduleMessageByBatchResponse" />
  </operation>
  <operation name="cancelScheduleMessageByPeriod">
    <input message="tns:cancelScheduleMessageByPeriod" />
    <output message="tns:cancelScheduleMessageByPeriodResponse" />
  </operation>
  <operation name="cancelScheduleMessageByIds">
    <input message="tns:cancelScheduleMessageByIds" />
    <output message="tns:cancelScheduleMessageByIdsResponse" />
  </operation>
</portType>
</wSDL>
```

Annotations in the image:

- An arrow points from the `schemaLocation` attribute in the `<xsd:import>` tag to the text: "Endereço das definições dos tipos e nomes dos parâmetros de cada porta (método)!".
- An arrow points from the `<portType name="WSManager">` block to the text: "Apenas as definições das portas!".

Schema:

http://webservice.comunika.com.br/WSComunika/WSManager?xsd=1 - Windows Internet Explorer

Endereço do schema disponível da WSDL!

```
<xs:sequence>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
- <xs:complexType name="responseGeneric">
- <xs:sequence>
  <xs:element name="result" type="xs:int" />
  <xs:element name="resultDesc" type="xs:string" minOccurs="0" />
  <xs:element name="token" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="cancelReturn">
- <xs:sequence>
  <xs:element name="id" type="xs:string" minOccurs="0" />
  <xs:element name="result" type="xs:int" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="getAuthentication">
- <xs:sequence>
  <xs:element name="login" type="xs:string" minOccurs="0" />
  <xs:element name="pass" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="getAuthenticationResponse">
- <xs:sequence>
  <xs:element name="return" type="tns:wsToken" minOccurs="0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="wsToken">
- <xs:complexContent>
  <xs:extension base="tns:responseGeneric">
  - <xs:sequence>
    <xs:element name="lastUpdate" type="xs:dateTime" minOccurs="0" />
  </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
- <xs:complexType name="cancelScheduleMessageByIds">
- <xs:sequence>
  <xs:element name="token" type="xs:string" minOccurs="0" />
  <xs:element name="ids" type="xs:string" minOccurs="0" />
</xs:sequence>
```

Esses são os campos do tipo complexo (tabela) "return"!

Esses são os parâmetros de entrada e seus tipos!

Esse é o tipo do parâmetro de retorno ("return"), que é uma tabela!

Além dos três parâmetros genéricos da tabela, ele possui esse parâmetro, ficando:
result, resultDesc, token e lastUpdate

Cadastrando o serviço WSManager

WSManager

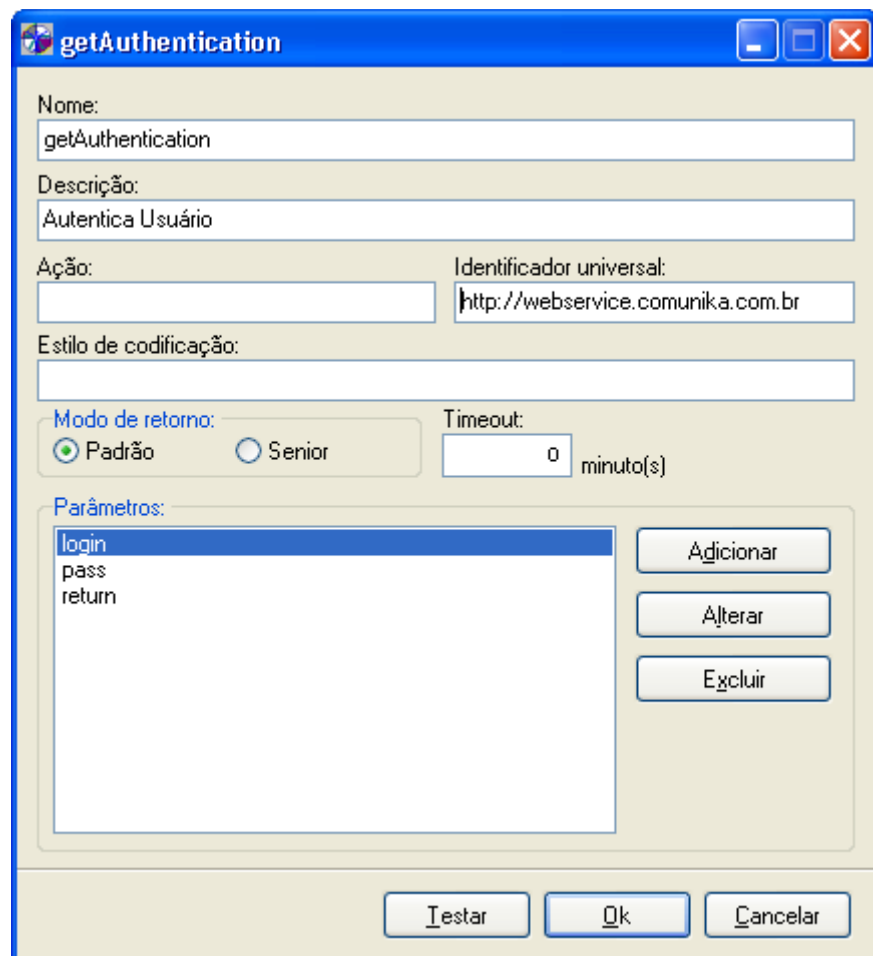
Nome:

WSManager

Ok Cancelar

Cadastrando a porta getAuthentication

(Identificador universal é o “namespace” informado na WSDL)



The image shows a Windows-style dialog box titled "getAuthentication". It contains several input fields and controls for configuring a web service port.

Nome: getAuthentication

Descrição: Autentica Usuário

Ação: [Empty text box]

Identificador universal: http://webservice.comunika.com.br

Estilo de codificação: [Empty text box]

Modo de retorno: ☒ Padrão ☐ Senior

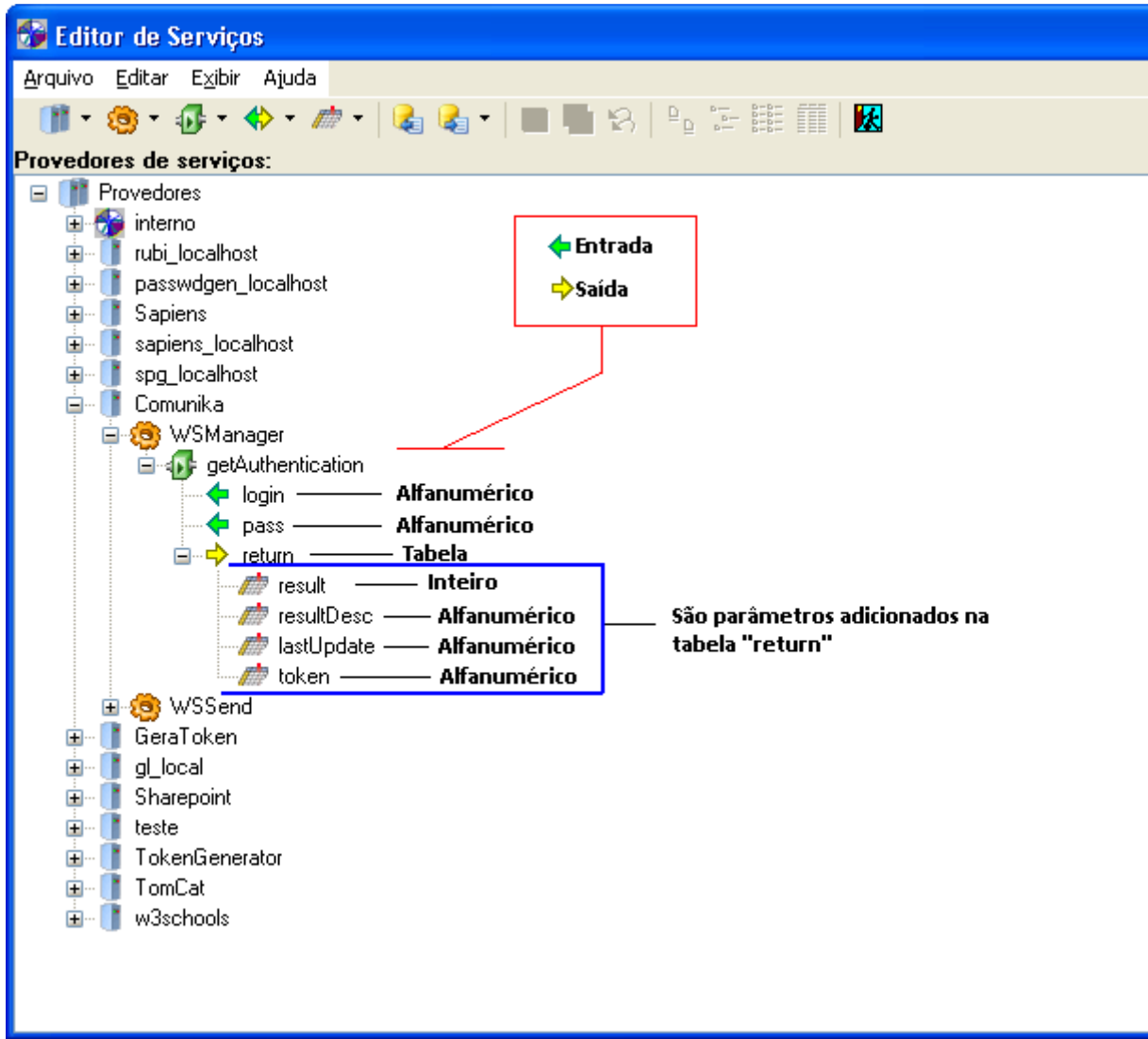
Timeout: 0 minuto(s)

Parâmetros:

login	Adicionar
pass	Alterar
return	Excluir

At the bottom of the dialog are three buttons: **Testar**, **Ok**, and **Cancelar**.

Parâmetros da porta getAuthentication



Entendendo a porta sendMessage do WSSend

WSDL:

https://webservice.comunika.com.br/WSComunika/WSSend?wsdl - Windows Internet Explorer

https://webservice.comunika.com.br/WSComunika/WSSend?wsdl

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.7-b01-. -->
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.7-b01-. -->
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://webservice.comunika.com.br" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://webservice.comunika.com.br" name="WSSend">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://webservice.comunika.com.br" schemaLocation="http://webservice.comunika.com.br:80/WSComunika/WSSend?xsd=1" />
    </xsd:schema>
  </types>
  <message name="sendMessage">
    <part name="parameters" element="tns:sendMessage" />
  </message>
  <message name="sendMessageResponse">
    <part name="parameters" element="tns:sendMessageResponse" />
  </message>
  <message name="sendBatch">
    <part name="parameters" element="tns:sendBatch" />
  </message>
  <message name="sendBatchResponse">
    <part name="parameters" element="tns:sendBatchResponse" />
  </message>
  <message name="sendSeverall">
    <part name="parameters" element="tns:sendSeverall" />
  </message>
  <message name="sendSeverallResponse">
    <part name="parameters" element="tns:sendSeverallResponse" />
  </message>
  <message name="sendSeverallWithBatch">
    <part name="parameters" element="tns:sendSeverallWithBatch" />
  </message>
  <message name="sendSeverallWithBatchResponse">
    <part name="parameters" element="tns:sendSeverallWithBatchResponse" />
  </message>
  <message name="sendScheduleBatch">
    <part name="parameters" element="tns:sendScheduleBatch" />
  </message>
  <message name="sendScheduleBatchResponse">
    <part name="parameters" element="tns:sendScheduleBatchResponse" />
  </message>
  <portType name="WSSend">
    <operation name="sendMessage">
```

Endereço das definições dos tipos e nomes dos parâmetros de cada porta (método)!

Apenas definição das portas!

Schema:

http://webservice.comunika.com.br/WSComunika/WSSend?xsd=1 - Windows Internet Explorer

Endereço do schema disponível na WSDL

```
<xs:element name="sendMessage" type="tns:sendMessage" />
<xs:element name="sendMessageResponse" type="tns:sendMessageResponse" />
<xs:element name="sendScheduleBatch" type="tns:sendScheduleBatch" />
<xs:element name="sendScheduleBatchResponse" type="tns:sendScheduleBatchResponse" />
<xs:element name="sendSeveral" type="tns:sendSeveral" />
<xs:element name="sendSeveralResponse" type="tns:sendSeveralResponse" />
<xs:element name="sendSeveralWithBatch" type="tns:sendSeveralWithBatch" />
<xs:element name="sendSeveralWithBatchResponse" type="tns:sendSeveralWithBatchResponse" />
<xs:complexType name="sendMessage">
  <xs:sequence>
    <xs:element name="token" type="xs:string" minOccurs="0" />
    <xs:element name="sender" type="xs:string" minOccurs="0" />
    <xs:element name="target" type="xs:string" minOccurs="0" />
    <xs:element name="idMessage" type="xs:string" minOccurs="0" />
    <xs:element name="sched" type="xs:string" minOccurs="0" />
    <xs:element name="activePeriodFirst" type="xs:int" />
    <xs:element name="activePeriodLast" type="xs:int" />
    <xs:element name="expireTime" type="xs:string" minOccurs="0" />
    <xs:element name="message" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="sendMessageResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:responseSendMessage" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="responseSendMessage">
  <xs:complexContent>
    <xs:extension base="tns:responseGeneric">
      <xs:sequence />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="responseGeneric">
  <xs:sequence>
    <xs:element name="result" type="xs:int" />
    <xs:element name="resultDesc" type="xs:string" minOccurs="0" />
    <xs:element name="token" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="sendScheduleBatch">
  <xs:sequence>
    <xs:element name="token" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

Esses são os parâmetros de entrada e seus tipos

Esse é o tipo do parâmetro de retorno ("return"), que é uma tabela!

Esses são os campos do tipo complexo (tabela) "return"!

Cadastrando o serviço WSSend

WSSend

Nome:

WSSend

Ok Cancelar

Cadastrando a porta sendMessage

sendMessage

Nome:

sendMessage

Descrição:

Ação:

Identificador universal:

http://webservice.comunika.com.br

Estilo de codificação:

Modo de retorno:

☒ Padrão

☐ Senior

Timeout:

0

minuto(s)

Parâmetros:

token

sender

target

idMessage

return

sched

activePeriodFirst

activePeriodLast

expireTime

message

Adicionar

Alterar

Excluir

Testar

Ok

Cancelar

Parâmetros da porta sendMessage

Editor de Serviços

Arquivo Editar Exibir Ajuda

Provedores de serviços:

Provedores

interno

rubi_localhost

passwdgen_localhost

Sapiens

sapiens_localhost

spg_localhost

Comunika

WSSend

sendMessage

GeraToken

gl_local

Sharepoint

teste

← Entrada

→ Saída

token — Alfanumérico

sender — Alfanumérico

target — Alfanumérico

idMessage — Alfanumérico

return — Tabela

result — Inteiro

resultDesc — Alfanumérico

token — Alfanumérico

sched — Alfanumérico

activePeriodFirst — Inteiro

activePeriodLast — Inteiro

expireTime — Inteiro

message — Alfanumérico

São parâmetros adicionados na tabela "return"