

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 “/”):

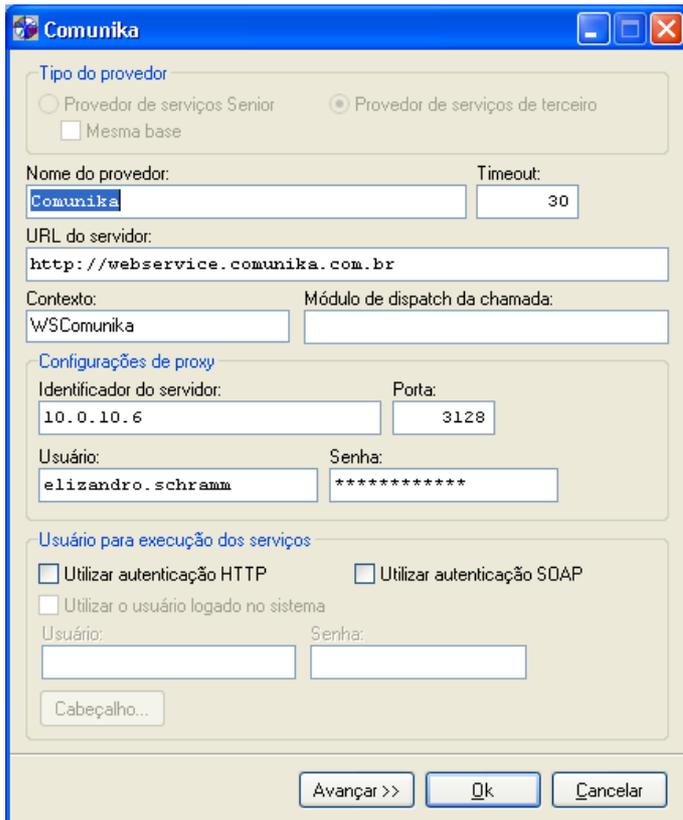
http://webservice.comunika.com.br

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)



# Entendendo a porta getAuthentication do serviço WSMManager

WSDL:

http://webservice.comunika.com.br/W5Comunika/WSManager?wsdl - Windows Internet Explorer

http://webservice.comunika.com.br/W5Comunika/WSManager?wsdl

```
<types>
- <xsd:schema>
  <xsd:import namespace="http://webservice.comunika.com.br" schemaLocation="http://webservice.comunika.com.br:80/W5Comunika/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">
```

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

Apenas as definições das portas!

Schema:

```
<?xml-stylesheet type="text/xsl" href="http://webservice.comunika.com.br/WSComunika/WSManager?xsd=1" />
<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>
</xs:complexType>
```

Endereço do schema disponível da WSDL!

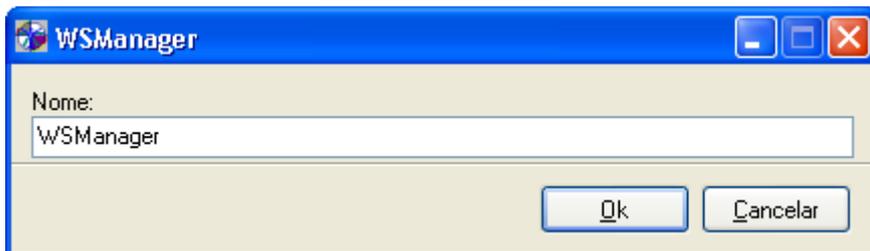
Esses são os campos do tipo complexo (tabela) "return"!

Esses são os parâmetros de entrada e seus tipos!

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

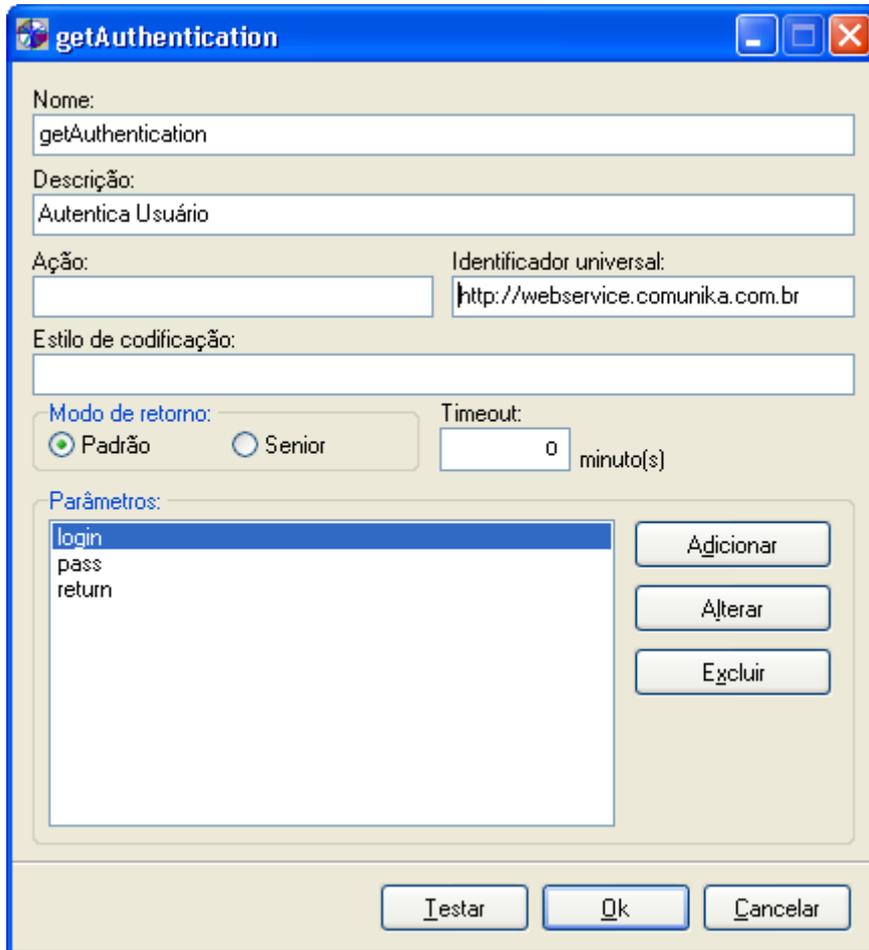
Além dos três parâmetros genéricos da tabela, ele possui esse parâmetro, ficando:  
result, resultDesc, token e lastUpdate

## Cadastrando o serviço WSManger



## 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:

- Nome:** A text box containing "getAuthentication".
- Descrição:** A text box containing "Autentica Usuário".
- Ação:** An empty text box.
- Identificador universal:** A text box containing "http://webservice.comunika.com.br".
- Estilo de codificação:** An empty text box.
- Modo de retorno:** Two radio buttons: "Padrão" (selected) and "Senior".
- Timeout:** A text box containing "0" followed by "minuto(s)".
- Parâmetros:** A list box containing "login", "pass", and "return". To the right of the list box are three buttons: "Adicionar", "Alterar", and "Excluir".
- Buttons:** At the bottom of the dialog are three buttons: "Testar", "Ok", and "Cancelar".

## Parâmetros da porta getAuthentication

**Editor de Serviços**

Arquivo Editar Exibir Ajuda

**Provedores de serviços:**

- Provedores
  - interno
  - rubi\_localhost
  - passwdgen\_localhost
  - Sapiens
  - sapiens\_localhost
  - spg\_localhost
  - Comunika
  - WSManager
    - getAuthentication
      - login — Alfanumérico
      - pass — Alfanumérico
      - return — Tabela
        - result — Inteiro
        - resultDesc — Alfanumérico
        - lastUpdate — Alfanumérico
        - token — Alfanumérico
  - WSSend
  - GeraToken
  - g\_local
  - Sharepoint
  - teste
  - TokenGenerator
  - TomCat
  - w3schools

← Entrada  
→ Saída

São parâmetros adicionados na tabela "return"

# Entendendo a porta sendMessage do 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="sendMessaae">
```

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

Apenas definição das portas!

Schema:

```
<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 base="tns:responseGeneric">
    <xs:sequence />
  </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>
```

## Cadastrando o serviço WSSend

WSSend

Nome:  
WSSend

Ok Cancelar

## Cadastrando a porta sendMessage

The image shows a Windows-style dialog box titled "sendMessage". It contains several input fields and controls for configuring a message service. The "Nome:" field is filled with "sendMessage". The "Descrição:" field is empty. The "Ação:" field is empty, and the "Identificador universal:" field contains "http://webservice.comunika.com.br". The "Estilo de codificação:" field is empty. Under "Modo de retorno:", the "Padrão" radio button is selected, and the "Senior" radio button is unselected. The "Timeout:" field is set to "0" with the unit "minuto(s)". The "Parâmetros:" section contains a list box with the following items: "token", "sender", "target", "idMessage", "return", "sched", "activePeriodFirst", "activePeriodLast", "expireTime", and "message". To the right of the list box are three buttons: "Adicionar", "Alterar", and "Excluir". At the bottom of the dialog are three buttons: "Istar", "Ok", and "Cancelar".

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

Istar 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
        - token — Alfanumérico (Entrada)
        - sender — Alfanumérico (Entrada)
        - target — Alfanumérico (Entrada)
        - idMessage — Alfanumérico (Entrada)
        - return — Tabela (Saída)
          - result — Inteiro
          - resultDesc — Alfanumérico
          - token — Alfanumérico
        - sched — Alfanumérico (Entrada)
        - activePeriodFirst — Inteiro (Entrada)
        - activePeriodLast — Inteiro (Entrada)
        - expireTime — Inteiro (Entrada)
        - message — Alfanumérico (Entrada)
- GeraToken
- g\_local
- Sharepoint
- teste

← Entrada  
→ Saída

São parâmetros adicionados na tabela "return"