

# **UNIVALI / CTTMAR**

Data: 11/11/2003

Padrão proposto pelo TCC

Versão: 1.0.0

Categoria: RFC (*Request For Comment*)

Editor: José Morelli Neto

## **Web Service para Transferência de Dados Telemétricos**

## Sumário

<b>1. Escopo .....</b>	<b>1</b>
<b>2. Referências Normativas .....</b>	<b>1</b>
<b>3. Termos e Definições.....</b>	<b>2</b>
3.1. CLIENTE .....	2
3.2. INTERFACE .....	2
3.3. OPERAÇÃO .....	2
3.4. REQUISIÇÃO .....	2
3.5. RESPOSTA.....	2
3.6. SERVIÇO.....	2
3.7. SERVIDOR.....	2
<b>4. Convenções .....</b>	<b>3</b>
4.1. TERMOS ABREVIADOS.....	3
<b>5. Diagrama de Caso de Uso .....</b>	<b>4</b>
<b>6. Elementos Básicos do Serviço .....</b>	<b>4</b>
6.1. INTRODUÇÃO.....	4
6.2. PROTOCOLO DE <i>WEB SERVICE</i> UTILIZADO .....	4
6.3. AUTENTICAÇÃO.....	5
6.4. CRIPTOGRAFIA.....	5
<b>7. Operações.....</b>	<b>5</b>
7.1. OPERAÇÃO SET_OCORRENCIA .....	5
7.2. OPERAÇÃO GET_COLETA_INF .....	7
7.3. OPERAÇÃO GET_SENSOR_INF .....	8
7.4. OPERAÇÃO GET_REGISTRO_INF .....	9
<b>8. Exemplos de Mensagens SOAP .....</b>	<b>11</b>
8.1. MENSAGEM SOAP DE REQUISIÇÃO DA OPERAÇÃO GET_COLETA_INF.....	11
8.2. MENSAGEM SOAP DE RESPOSTA DA OPERAÇÃO GET_COLETA_INF .....	11

<b>8.3. MENSAGEM SOAP DE REQUISIÇÃO DA OPERAÇÃO GET_REGISTRO_INF.....</b>	<b>12</b>
<b>8.4. MENSAGEM SOAP DE RESPOSTA DA OPERAÇÃO GET_REGISTRO_INF .....</b>	<b>12</b>
<b>8.5. MENSAGEM SOAP DE REQUISIÇÃO DA OPERAÇÃO SET_REGISTRO .....</b>	<b>13</b>
<b>8.6. MENSAGEM SOAP DE RESPOSTA DA OPERAÇÃO SET_OCORRENCIA.....</b>	<b>14</b>
<b>9. Documento WSDL que descreve o padrão proposto .....</b>	<b>14</b>

## **i. Prefácio**

O *Web Service* para Transferência de Dados Telemétricos (WSTDT) é o resultado de estudos realizados para a elaboração de um Trabalho de Conclusão de Curso no ano de 2003. O próximo passo a ser realizado é publicar esse documento na forma de um RFC (*Request for Comment*).

## **ii. Organizações Participantes**

As seguintes organizações submeteram sugestões ou implementações para essa especificação.

?? UNIVALI - Universidade do Vale do Itajaí

## **iii. Contribuições para esse Documento**

Abaixo são listadas as pessoas que contribuíram para a elaboração dessa proposta.

<b>Nome</b>	<b>E-mail</b>	<b>Organização</b>
Fabio Coelho	qoelhex@cttmar.univali.br	UNIVALI
Fernando Simon	simon@cttmar.univali.br	UNIVALI
Rodrigo Becke Cabral	cabral@univali.br	UNIVALI

## **iv. Histórico de Revisão**

<b>Data</b>	<b>Versão</b>	<b>Autor</b>	<b>Descrição</b>
11/11/2003	1.0	José Morelli Neto	Versão inicial da proposta de padrão

## **v. Trabalhos Futuros**

Futuras versões dessa proposta de padronização são aguardadas considerando algumas expansões das habilidades especificadas nesse documento. Serão citados abaixo, possíveis trabalhos inclusos no futuro:

- a) Definir novas operações para possibilitar consultas diferenciadas;
- b) Utilizar estruturas de PKI (*Public Key Infrastructure*) ou XKMS (*XML Key Management Specification*) para assegurar as transações efetuadas entre as empresas de rastreamento e o cliente.

## **Introdução**

A Telemetria é a técnica que mede quantidades, transmitindo os resultados para um centro de controle que é responsável pela interpretação, apresentação e/ou armazenamento dos valores medidos. Os meios de transmissão de dados utilizados podem ser muito variados, abrangendo desde linhas telefônicas convencionais até a transmissão por meio da Internet.

Recentemente tem-se observado a existência de inúmeras formas para o transporte de dados entre as empresas de rastreamento, responsáveis por receber os dados dos pontos de coleta remotos, e o cliente final, responsável por tratar os dados conforme suas necessidades, formando uma lacuna na padronização das transações de telemetria em nível de camada de aplicação.

Essa proposta visa preencher esta lacuna apresentando uma especificação de transação de dados telemétricos através da tecnologia de *Web Services*.

## Web Service para Transferência de Dados Telemétricos

### 1. Escopo

Esse documento apresenta como implementar serviços que possam atender as necessidades para a troca de informações em sistemas telemétricos, descrevendo os formatos de mensagens de requisição e resposta, as operações disponíveis, os tipos de dados utilizados pelas operações, entre outros. Essa proposta é baseada em tecnologias já existentes e não pretende definir novos componentes para a utilização. Entre as tecnologias envolvidas, pode-se citar o uso de *Web Services* implementado sob o protocolo SOAP, SSL prevendo a criptografia dos dados transferidos e o uso de autenticação para o serviço, de forma a não permitir o uso das operações por pessoas não autorizadas.

### 2. Referências Normativas

Os documentos normativos apresentados a seguir foram utilizados como referência nesse texto, constituindo parte dessa especificação. Para referências datadas, as correções subseqüentes ou as revisões a alguma destas publicações não se aplicam. Para referências não datadas, a última edição do documento normativo pode ser aplicada.

AUSTIN, Daniel (Ed.) et al. **Web Services Architecture Requirements**. 2002. Disponível em: <<http://www.w3.org/TR/wsa-reqs>>. Acesso em: 05 mai. 2003.

BIRON, Paul V. & MALHOTRA, ashok. **XML Schema Part 2: Datatypes**. 2001. Disponível em: <<http://www.w3.org/TR/xmlschema-2/>>. Acesso em: 06 jun. 2003.

BOX, Don et al. **Simple Object Access Protocol (SOAP) 1.1**. 2000. Disponível em: <<http://www.w3.org/TR/SOAP>>. Acesso em: 10 mai. 2003.

BRAY, Tim (Ed.); HOLLANDER, Dave (Ed.); LAYMAN, Andrew (Ed.). **Namespaces in XML**. 1999. Disponível em: <<http://www.w3.org/TR/1999/REC-xml-names-19990114>>. Acesso em: 06 jun. 2003.

BRAY, Tim (Ed.) et al. **Extensible Markup Language (XML) 1.0**. 2. ed., 2000. Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: 05 jun. 2003.

FREED, N & BORENSTEIN, N. (Ed.). **Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies**. 1996. Disponível em: <<http://www.ietf.org/rfc/rfc2045.txt>>.

THOMPSON, Henry S. et al. **XML Schema Part 1: Structures**. 2001. Disponível em: <<http://www.w3.org/TR/xmlschema-1/>>. Acesso em: 06 jun. 2003.

### 3. Termos e Definições

#### 3.1. Cliente

Componente de *software* que invoca uma **operação** de um **servidor**.

#### 3.2. Interface

Conjunto de **operações** nomeados que caracterizam o comportamento de uma entidade.

#### 3.3. Operação

Especificação de uma função ou consulta em que um objeto possa ser chamado para executar.

#### 3.4. Requisição

Invocação de uma **operação** pelo **cliente**.

#### 3.5. Resposta

Resultado de uma **operação**, retornado do **servidor** para o **cliente**.

#### 3.6. Serviço

Parte distinta da funcionalidade que é fornecida por uma entidade através de **interfaces**.

#### 3.7. Servidor

É uma implementação do **serviço** – um componente de *software* onde o **cliente** poderá invocar uma **operação**.

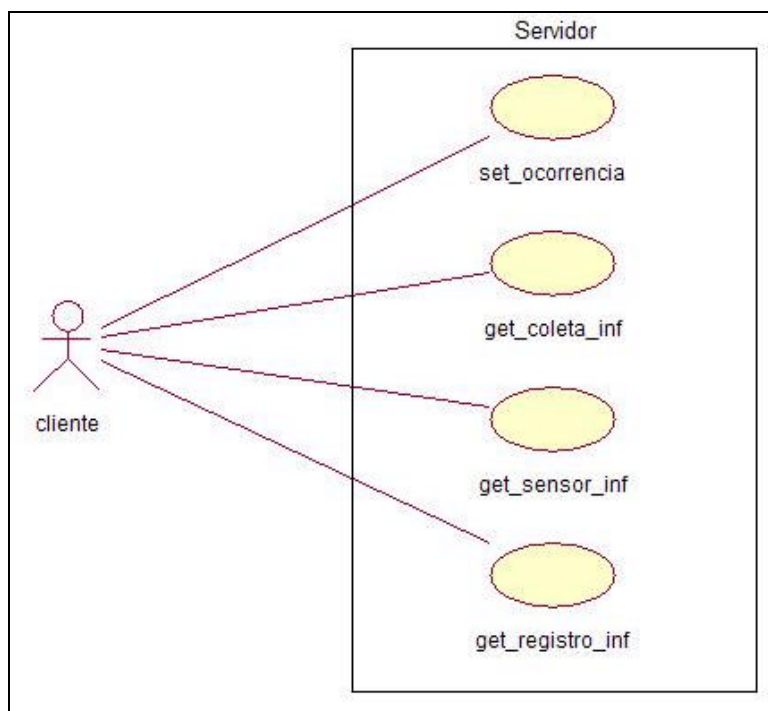
## 4. Convenções

### 4.1. Termos Abreviados

HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IP	<i>Internet Protocol</i>
MIME	<i>Multi-purpose Internet Mail Extension</i>
PEAR	<i>PHP Extension and Application Repository</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
PKI	<i>Public Key Infrastructure</i>
SOAP	<i>Simple Object Access Protocol</i>
SSL	<i>Secure Socket Layer</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifiers</i>
URL	<i>Uniform Resource Locator</i>
XKMS	<i>XML Key Management Specification</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>XML Schema Document</i>
W3C	<i>World Wide Web Consortium</i>
WSDL	<i>Web Services Description Language</i>

## 5. Diagrama de Caso de Uso

A Figura a seguir apresenta o diagrama de Caso de Uso. O Cliente é uma aplicação que fará uso de chamadas via *Web Service* às operações disponíveis remotamente no servidor. Cada operação possui parâmetros que devem ser repassados na requisição ou que recebem como resposta às requisições efetuadas.



## 6. Elementos Básicos do Serviço

### 6.1. Introdução

Esse capítulo descreve os aspectos gerais para utilização do *Web Service* proposto, independente de operações em particular, ou que são comuns a várias operações ou *interface*.

### 6.2. Protocolo de *Web Service* utilizado

O protocolo de *Web Service* utilizado para a implementação do padrão proposto deve ser o SOAP (*Simple Object Access Protocol*). Esse protocolo permite a integração com o WSDL de forma mais fácil e rápida.

### 6.3. Autenticação

Todas as operações disponíveis por esse *web service* devem ser acessadas apenas por clientes que possuem permissão. A permissão é concedida no momento de execução da operação, onde o cliente, ao enviar uma mensagem SOAP de requisição, deverá adicionar no cabeçalho da mensagem as informações pertinentes a execução. A Figura 1 apresenta um exemplo de cabeçalho SOAP com as informações para a autenticação.

```
<SOAP-ENV:Header>
  <AtD:autentica SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next "
  SOAP-ENV:mustUnderstand="1">
    <cli_cod xsi:type="xsd:string">1313</cli_cod>
    <cli_senha xsi:type="xsd:string">difícil</cli_senha>
  </AtD:autentica>
</SOAP-ENV:Header>
```

Figura 1: Exemplo de cabeçalho SOAP autenticado.

### 6.4. Criptografia

É recomendado o uso de meios de criptografia para garantir a segurança nos dados transacionados. Para a implementação desse padrão, o uso de HTTPS (*HyperText Transfer Protocol Secure*) garante o transporte de forma confiável, pois utiliza a camada de segurança SSL (*Secure Socket Layer*) sobre o protocolo HTTP (*HyperText Transfer Protocol*).

## 7. Operações

As operações, descritas a seguir, são implementadas no servidor e disponíveis aos clientes. Cada operação é apresentada em conjunto com os tipos de dados utilizados tanto no chamado a elas, quanto no retorno após processado pelo servidor.

### 7.1. Operação `set_ocorrencia`

Essa operação é responsável por adicionar uma ocorrência à base de dados do servidor. Como apresentado na Figura 2, essa operação possui como entrada (ou passagem de parâmetros) a estrutura de dados `dados_ocorrencia` e retorna `op_cod` após o processamento.

```

<message name="set_ocorrenciaRequest">
  <part name="inputStruct" type="ns5:dados_ocorrencia"/>
</message>
<message name="set_ocorrenciaResponse">
  <part name="op_cod" type="xsd:string"/>
</message>

<operation name="set_ocorrencia">
  <input message="tns:set_ocorrenciaRequest"/>
  <output message="tns:set_ocorrenciaResponse"/>
</operation>

```

Figura 2: Descrição da operação set\_ocorrencia.

A Figura 3 ilustra as estruturas de dados que compõem dados\_ocorrencia. Como pode ser observado, o atributo arr\_sen é do tipo array\_sensor, que por sua vez é um *array* de dad\_sensor. Isso se faz necessário, pois ao registrar uma ocorrência, leituras medidas de vários sensores diferentes podem estar sendo repassados para a operação. A Tabela 1 apresenta os atributos da estrutura dados\_ocorrencia e a Tabela 2 apresenta os atributos de dad\_sensor.

```

<complexType name="dad_sensor">
  <all>
    <element name="lei_med" type="xsd:float"/>
    <element name="sen_cod" type="xsd:int"/>
  </all>
</complexType>

<complexType name="array_sensor">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:dad_sensor[]" />
    </restriction>
  </complexContent>
</complexType>

<complexType name="dados_ocorrencia">
<complexType name="dados_ocorrencia">
  <all>
    <element name="col_cod" type="xsd:int"/>
    <element name="reg_lat" type="xsd:float"/>
    <element name="reg_lon" type="xsd:float"/>
    <element name="oco_dat" type="xsd:datetime"/>
    <element name="arr_sen" type="ns5:array_sensor"/>
  </all>
</complexType>

```

Figura 3: Tipos de dados de entrada da operação set\_ocorrencia.

Tabela 1: Atributos utilizados pela estrutura dados\_ocorrencia.

Nome do Atributo	Tipo/Tamanho	Descrição
col_cod	Inteiro	Código do ponto de Coleta.
reg_lat	Real/13	Latitude no momento da coleta.
reg_lon	Real/13	Longitude no momento da coleta.
oco_dat	<i>DateTime</i>	Data e hora no momento da coleta.
arr_sen	Array de array_sensor	array_sensor é um <i>array</i> de dad_sensor, que carrega informações de coleta dos sensores.

Tabela 2: Atributos utilizados pela estrutura dad\_sensor.

Nome do Atributo	Tipo/Tamanho	Descrição
sen_cod	Inteiro	Código do Sensor.
lei_med	Real/13	Leitura medida no sensor.

## 7.2. Operação get\_coleta\_inf

Essa operação solicita ao servidor todos os pontos de coleta de um determinado cliente cadastrado no banco de dados. Como apresentado na Figura 4, essa operação possui como entrada o código do cliente (cli\_cod) e, após o processamento, é retornada a estrutura de dados registro\_coleta.

```

<message name="get_coleta_infRequest">
  <part name="cli_cod" type="xsd:int"/>
</message>
<message name="get_coleta_infResponse">
  <part name="outputStruct" type="ns5:registro_coleta"/>
</message>

<operation name="get_coleta_inf">
  <input message="tns:get_coleta_infRequest"/>
  <output message="tns:get_coleta_infResponse"/>
</operation>

```

Figura 4: Descrição da operação get\_coleta\_inf.

A Figura 5 ilustra as estruturas de dados que compõem registro\_coleta. Como pode ser observado, o tipo registro\_coleta é um *array* de campos\_coleta. A Tabela 3 apresenta os atributos utilizados pela estrutura campos\_coleta.

```

<complexType name="campos_coleta">
  <all>
    <element name="col_cod" type="xsd:int"/>
    <element name="col_nom" type="xsd:string"/>
    <element name="col_des" type="xsd:string"/>
  </all>
</complexType>

<complexType name="registro_coleta">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_coleta[]"/>
    </restriction>
  </complexContent>
</complexType>

```

Figura 5: Tipos de dados de entrada da operação `get_coleta_inf`.

Tabela 3: Atributos utilizados pela estrutura `campos_coleta`.

Nome do Atributo	Tipo/Tamanho	Descrição
<code>col_cod</code>	Inteiro	Código do ponto de coleta.
<code>col_nom</code>	Alfanumérico/50	Nome do ponto de coleta.
<code>col_des</code>	Alfanumérico/100	Descrição do ponto de coleta.

### 7.3. Operação `get_sensor_inf`

Essa operação tem por finalidade solicitar ao servidor todos os sensores existentes em um determinado ponto de coleta. Como apresentado na Figura 6, essa operação possui como entrada o código do ponto de coleta (`col_cod`) e, após o processamento, é retornada a estrutura de dados `registro_sensor`.

```

<message name="get_sensor_infRequest">
  <part name="col_cod" type="xsd:int"/>
</message>
<message name="get_sensor_infResponse">
  <part name="outputStruct" type="ns5:registro_sensor"/>
</message>

<operation name="get_sensor_inf">
  <input message="tns:get_sensor_infRequest"/>
  <output message="tns:get_sensor_infResponse"/>
</operation>

```

Figura 6: Descrição da operação `get_sensor_inf`.

A Figura 7 apresenta as estruturas de dados que compõem `registro_sensor`. Semelhante a estrutura de dados da operação anterior, o tipo `registro_sensor`

também é um *array* de `campos_sensor`. A Tabela 4 apresenta os atributos utilizados pela estrutura `campos_sensor`.

```
<complexType name="campos_sensor">
  <all>
    <element name="sen_cod" type="xsd:int"/>
    <element name="sen_nom" type="xsd:string"/>
    <element name="sen_des" type="xsd:string"/>
    <element name="sen_uni" type="xsd:string"/>
  </all>
</complexType>

<complexType name="registro_sensor">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_sensor[ ]"/>
    </restriction>
  </complexContent>
</complexType>
```

Figura 7: Tipos de dados de entrada da operação `get_sensor_inf`.

Tabela 4: Atributos utilizados pela estrutura `campos_sensor`.

Nome do Atributo	Tipo/Tamanho	Descrição
<code>sen_cod</code>	Inteiro	Código do Sensor.
<code>sen_nom</code>	Alfanumérico/50	Nome do Sensor.
<code>sen_des</code>	Alfanumérico/50	Descrição do sensor.
<code>sen_uni</code>	Alfanumérico/50	Unidade de medida que o sensor está coletando.

#### 7.4. Operação `get_registro_inf`

Essa operação solicita ao servidor todas as ocorrências registradas nos últimos dez (10) dias de um determinado ponto de coleta. O parâmetro de entrada dessa função é igual ao da operação `get_sensor_inf`, ou seja, o código do ponto de coleta (`col_cod`). Como apresentado na Figura 8, essa operação possui como retorno a estrutura de dados `reg_registro`.

```

<message name="get_registro_infRequest">
  <part name="col_cod" type="xsd:int"/>
</message>
<message name="get_registro_infResponse">
  <part name="outputStruct" type="ns5:reg_registro"/>
</message>

<operation name="get_registro_inf">
  <input message="tns:get_registro_infRequest"/>
  <output message="tns:get_registro_infResponse"/>
</operation>

```

Figura 8: Descrição da operação `get_registro_inf`.

A Figura 9 apresenta as estruturas de dados que compõem `reg_registro`. A estrutura de dados `reg_registro` é um *array* de `campos_registro`. A Tabela 4 apresenta os atributos utilizados pela estrutura `campos_registro`.

```

<complexType name="campos_registro">
  <all>
    <element name="reg_cod" type="xsd:int"/>
    <element name="reg_dat" type="xsd:string"/>
    <element name="reg_lat" type="xsd:float"/>
    <element name="reg_lon" type="xsd:float"/>
    <element name="lei_med" type="xsd:float"/>
    <element name="sen_cod" type="xsd:int"/>
  </all>
</complexType>

<complexType name="reg_registro">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_registro[]"/>
    </restriction>
  </complexContent>
</complexType>

```

Figura 9: Tipos de dados de entrada da operação `get_registro_inf`.

Tabela 5: Atributos utilizados pela estrutura `campos_sensor`.

Nome do Atributo	Tipo/Tamanho	Descrição
<code>reg_cod</code>	Inteiro	Código do Registro.
<code>reg_dat</code>	DataHora	Data e hora de coleta do registro.
<code>reg_lat</code>	Real/13	Latitude no momento da coleta.
<code>reg_lon</code>	Real/13	Longitude no momento da coleta.
<code>sen_cod</code>	Inteiro	Código do Sensor.
<code>lei_med</code>	Real/13	Leitura medida no sensor.

## 8. Exemplos de Mensagens SOAP

### 8.1. Mensagem SOAP de Requisição da operação get\_coleta\_inf

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/ "
xmlns:xsd="http://www.w3.org/2001/XMLSchema "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:AtD="urn:Manuseio_Cabecalho " >
  <SOAP-ENV:Header>
    <AtD:autentica SOAP-ENV:actor=http://schemas.xmlsoap.org/soap/actor/next
      SOAP-ENV:mustUnderstand="1" >
      <cli_cod xsi:type="xsd:string">1313</cli_cod>
      <cli_senha xsi:type="xsd:string">dificil</cli_senha>
    </AtD:autentica>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
    <NS1:get_coleta_inf xmlns:NS1="http://neto.univali.br/rastro/servidor" >
      <cli_cod xsi:type="xsd:int">1313</cli_cod>
    </NS1:get_coleta_inf>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 8.2. Mensagem SOAP de Resposta da operação get\_coleta\_inf

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns4="urn:Manuseio_Cabecalho "
xmlns:ns5="http://neto.univali.br/rastro/xsd "
xmlns:ns6="http://neto.univali.br/rastro/servidor "
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <SOAP-ENV:Header>
    <ns4:autentica xsi:type="xsd:string">Autenticacao OK</ns4:autentica>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns6:get_coleta_infResponse >
      <outputStruct xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="ns5:campos_coleta[3]"
        SOAP-ENC:offset="[0]" >
        <item xsi:type="ns5:campos_coleta">
          <col_cod xsi:type="xsd:int">1</col_cod>
          <col_nom xsi:type="xsd:string">Titanic</col_nom>
          <col_des xsi:type="xsd:string">Emb. Titanic</col_des>
        </item>
        <item xsi:type="ns5:campos_coleta">
          <col_cod xsi:type="xsd:int">2</col_cod>
          <col_nom xsi:type="xsd:string">Bateau Mouche</col_nom>
          <col_des xsi:type="xsd:string">Pesqueira BM</col_des>
        </item>
        <item xsi:type="ns5:campos_coleta">
          <col_cod xsi:type="xsd:int">3</col_cod>
          <col_nom xsi:type="xsd:string">Catatatu</col_nom>
          <col_des xsi:type="xsd:string">Emb. Catatau</col_des>
        </item>
      </outputStruct>
    </ns6:get_coleta_infResponse >
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 8.3. Mensagem SOAP de Requisição da operação get\_registro\_inf

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:AtD="urn:Manuseio_Cabecalho" ><SOAP-ENV:Header>
  <AtD:autentica SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" SOAP-
ENV:mustUnderstand="1">
    <cli_cod xsi:type="xsd:string">1313</cli_cod>
    <cli_senha xsi:type="xsd:string">dificil</cli_senha>
  </AtD:autentica></SOAP-ENV:Header>
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <NS1:get_registro_inf xmlns:NS1="http://neto.univali.br/rastro/servidor">
    <col_cod xsi:type="xsd:int">1</col_cod>
  </NS1:get_registro_inf>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 8.4. Mensagem SOAP de Resposta da operação get\_registro\_inf

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns4="urn:Manuseio_Cabecalho"
xmlns:ns5="http://neto.univali.br/rastro/xsd"
xmlns:ns6="http://neto.univali.br/rastro/servidor"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <SOAP-ENV:Header>
    <ns4:autentica xsi:type="xsd:string">Autenticacao OK</ns4:autentica>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns6:get_registro_infResponse>
      <outputStruct xsi:type="SOAP-ENC:Array" SOAP-
ENC:arrayType="ns5:campos_registro[4]"
SOAP-ENC:offset="0">
        <item xsi:type="ns5:campos_registro">
          <reg_cod xsi:type="xsd:int">1</reg_cod>
          <reg_dat xsi:type="xsd:string">2003-11-09 15:23:07</reg_dat>
          <reg_lat xsi:type="xsd:float">-98.87</reg_lat>
          <reg_lon xsi:type="xsd:float">-82.33</reg_lon>
          <lei_med xsi:type="xsd:float">13.22</lei_med>
          <sen_cod xsi:type="xsd:int">1</sen_cod>
        </item>
        <item xsi:type="ns5:campos_registro">
          <reg_cod xsi:type="xsd:int">1</reg_cod>
          <reg_dat xsi:type="xsd:string">2003-11-09 15:23:07</reg_dat>
          <reg_lat xsi:type="xsd:float">-98.87</reg_lat>
          <reg_lon xsi:type="xsd:float">-82.33</reg_lon>
          <lei_med xsi:type="xsd:float">35</lei_med>
          <sen_cod xsi:type="xsd:int">2</sen_cod>
        </item>
        <item xsi:type="ns5:campos_registro">
          <reg_cod xsi:type="xsd:int">2</reg_cod>
          <reg_dat xsi:type="xsd:string">2003-11-09 15:23:44</reg_dat>
          <reg_lat xsi:type="xsd:float">2</reg_lat>
          <reg_lon xsi:type="xsd:float">2</reg_lon>
          <lei_med xsi:type="xsd:float">2</lei_med>
          <sen_cod xsi:type="xsd:int">1</sen_cod>
        </item>
      </outputStruct>
    </ns6:get_registro_infResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

<item xsi:type="ns5:campos_registro">
  <reg_cod xsi:type="xsd:int">2</reg_cod>
  <reg_dat xsi:type="xsd:string">2003-11-09 15:23:44</reg_dat>
  <reg_lat xsi:type="xsd:float">2</reg_lat>
  <reg_lon xsi:type="xsd:float">2</reg_lon>
  <lei_med xsi:type="xsd:float">2</lei_med>
  <sen_cod xsi:type="xsd:int">2</sen_cod>
</item>
</outputStruct>
</ns6:get_registro_infResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 8.5. Mensagem SOAP de Requisição da operação set\_registro

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="http://neto.univali.br/rastro/xsd"
xmlns:AtD="urn:Manuseio_Cabecalho">
  <SOAP-ENV:Header>
    <AtD:autentica SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" SOAP-
ENV:mustUnderstand="1">
      <cli_cod xsi:type="xsd:string">1313</cli_cod>
      <cli_senha xsi:type="xsd:string">dificil</cli_senha>
    </AtD:autentica>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:set_ocorrencia xmlns:m="http://neto.univali.br/rastro/servidor" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <inputStruct xsi:type="m0:dados_ocorrencia">
        <col_cod xsi:type="xsd:int">2</col_cod>
        <reg_lat xsi:type="xsd:float">24.453</reg_lat>
        <reg_lon xsi:type="xsd:float">52.343</reg_lon>
        <oco_dat xsi:type="xsd:datetime">03-12-05 17:36:12</oco_dat>
        <arr_sen xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="dad_sensor[1]">
          <m0:item0 xsi:type="m0:dad_sensor">
            <lei_med xsi:type="xsd:float">27.43</lei_med>
            <sen_cod xsi:type="xsd:int">1</sen_cod>
          </m0:item0>
          <m0:item1 xsi:type="m0:dad_sensor">
            <lei_med xsi:type="xsd:float">43.27</lei_med>
            <sen_cod xsi:type="xsd:int">2</sen_cod>
          </m0:item1>
        </arr_sen>
      </inputStruct>
    </m:set_ocorrencia>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 8.6. Mensagem SOAP de Resposta da operação set\_ocorrencia

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:Manuseio_Cabecalho"
  xmlns:ns5="http://neto.univali.br/rastro/xsd"
  xmlns:ns6="http://neto.univali.br/rastro/servidor"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <SOAP-ENV:Header>
    <ns4:autentica xsi:type="xsd:string">Autenticacao OK</ns4:autentica>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns6:set_ocorrenciaResponse>
      <op_cod xsi:type="xsd:int">13</op_cod>
    </ns6:set_ocorrenciaResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## 9. Documento WSDL que descreve o padrão proposto

```
<?xml version="1.0"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="urn:WS_telemetria"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:ns5="http://localhost/rastro/xsd" targetNamespace="urn:WS_telemetria"
  name="WS_telemetria">
  <wsdl:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://localhost/rastro/xsd">
      <complexType name="dad_sensor">
        <all>
          <element name="lei_med" type="xsd:float"/>
          <element name="sen_cod" type="xsd:int"/>
        </all>
      </complexType>
      <complexType name="array_sensor">
        <complexContent>
          <restriction base="SOAP-ENC:Array">
            <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:dad_sensor[]"/>
          </restriction>
        </complexContent>
      </complexType>
      <complexType name="dados_ocorrencia">
        <all>
          <element name="col_cod" type="xsd:int"/>
          <element name="reg_lat" type="xsd:float"/>
          <element name="reg_lon" type="xsd:float"/>
          <element name="oco_dat" type="xsd:datetime"/>
          <element name="arr_sen" type="ns5:array_sensor"/>
        </all>
      </complexType>
      <complexType name="campos_coleta">
        <all>
          <element name="col_cod" type="xsd:int"/>
          <element name="col_nom" type="xsd:string"/>
          <element name="col_des" type="xsd:string"/>
        </all>
      </complexType>
```

```

<complexType name="registro_coleta">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_coleta[]" />
    </restriction>
  </complexContent>
</complexType>
<complexType name="campos_registro">
  <all>
    <element name="reg_cod" type="xsd:int" />
    <element name="reg_dat" type="xsd:string" />
    <element name="reg_lat" type="xsd:float" />
    <element name="reg_lon" type="xsd:float" />
    <element name="lei_med" type="xsd:float" />
    <element name="sen_cod" type="xsd:int" />
  </all>
</complexType>
<complexType name="reg_registro">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_registro[]" />
    </restriction>
  </complexContent>
</complexType>
<complexType name="campos_sensor">
  <all>
    <element name="sen_cod" type="xsd:int" />
    <element name="sen_nom" type="xsd:string" />
    <element name="sen_des" type="xsd:string" />
    <element name="sen_uni" type="xsd:string" />
  </all>
</complexType>
<complexType name="registro_sensor">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="ns5:campos_sensor[]" />
    </restriction>
  </complexContent>
</complexType>
</schema>
</wsdl:types>
<message name="set_ocorrenciaRequest">
  <part name="inputStruct" type="ns5:dados_ocorrencia" />
</message>
<message name="set_ocorrenciaResponse">
  <part name="op_cod" type="xsd:string" />
</message>
<message name="get_coleta_infRequest">
  <part name="cli_cod" type="xsd:int" />
</message>
<message name="get_coleta_infResponse">
  <part name="outputStruct" type="ns5:registro_coleta" />
</message>
<message name="get_registro_infRequest">
  <part name="col_cod" type="xsd:int" />
</message>
<message name="get_registro_infResponse">
  <part name="outputStruct" type="ns5:reg_registro" />
</message>
<message name="get_sensor_infRequest">
  <part name="col_cod" type="xsd:int" />
</message>
<message name="get_sensor_infResponse">
  <part name="outputStruct" type="ns5:registro_sensor" />
</message>
<portType name="WS_telemetriaPort">
  <operation name="set_ocorrencia">
    <input message="tns:set_ocorrenciaRequest" />
    <output message="tns:set_ocorrenciaResponse" />
  </operation>
  <operation name="get_coleta_inf">
    <input message="tns:get_coleta_infRequest" />
    <output message="tns:get_coleta_infResponse" />
  </operation>

```

```

<operation name="get_registro_inf">
  <input message="tns:get_registro_infRequest"/>
  <output message="tns:get_registro_infResponse"/>
</operation>
<operation name="get_sensor_inf">
  <input message="tns:get_sensor_infRequest"/>
  <output message="tns:get_sensor_infResponse"/>
</operation>
</portType>
<binding name="WS_telemetriaBinding" type="tns:WS_telemetriaPort">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="set_ocorrencia">
    <soap:operation soapAction="http://localhost/rastro/servidor#metodos#set_ocorrencia"/>
    <input>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </input>
    <output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </output>
  </operation>
  <operation name="get_coleta_inf">
    <soap:operation soapAction="http://localhost/rastro/servidor#metodos#get_coleta_inf"/>
    <input>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </input>
    <output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </output>
  </operation>
  <operation name="get_registro_inf">
    <soap:operation
soapAction="http://localhost/rastro/servidor#metodos#get_registro_inf"/>
    <input>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </input>
    <output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </output>
  </operation>
  <operation name="get_sensor_inf">
    <soap:operation soapAction="http://localhost/rastro/servidor#metodos#get_sensor_inf"/>
    <input>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </input>
    <output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost/rastro/servidor"/>
    </output>
  </operation>
</binding>
<service name="WS_telemetriaService">
  <documentation/>
  <port name="WS_telemetriaPort" binding="tns:WS_telemetriaBinding">
    <soap:address location="http://localhost/rastro/wssserver.php"/>
  </port>
</service>
</wsdl:definitions>

```