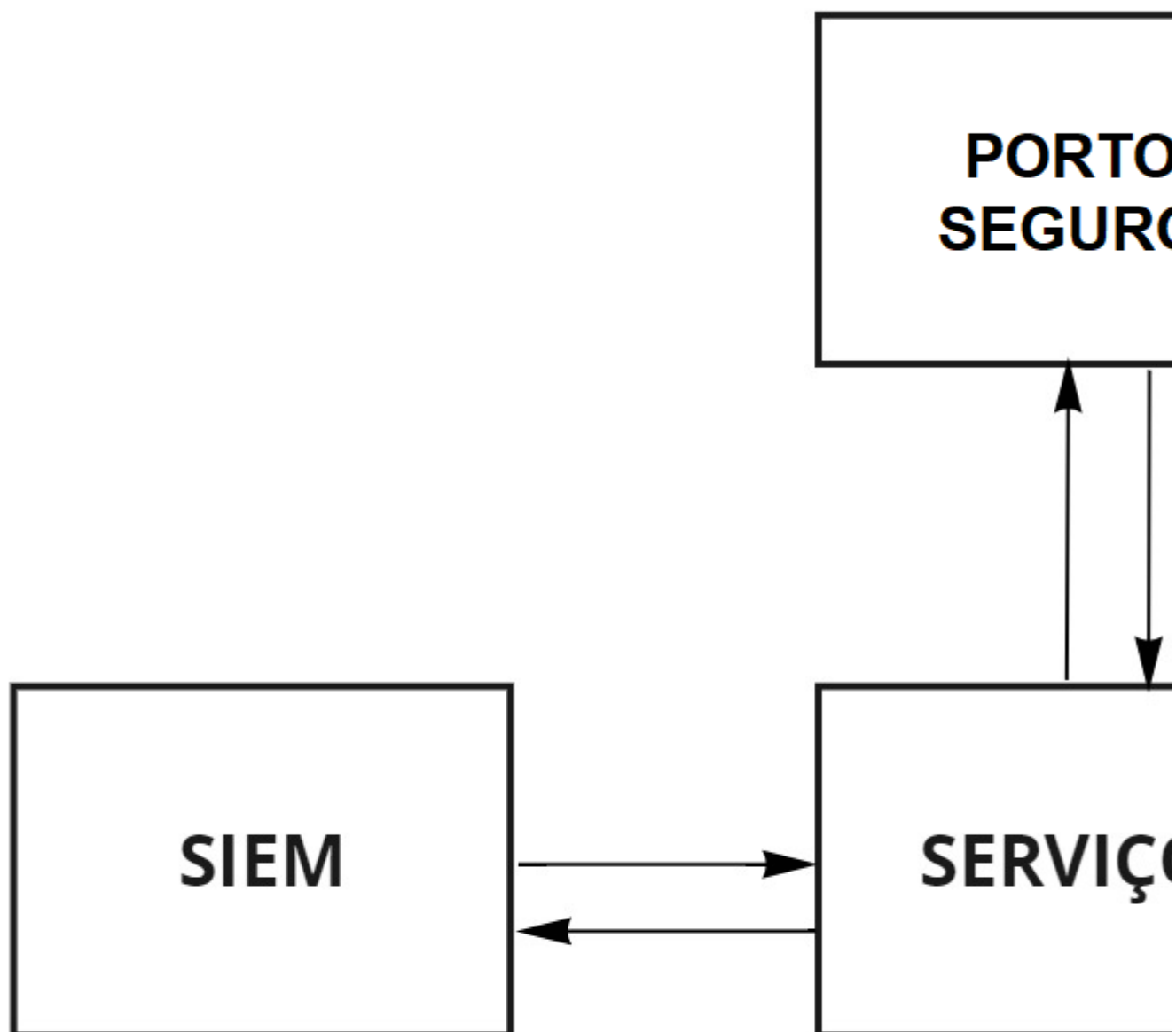


# PROC001 - Averbação via API Porto Seguro (Webservice tipo REST)

***Fluxograma***



Os sistemas SIEM e GA, se comunicaram apenas com o Serviço, que irá receber a informação e encaminhar para a Porto Seguro, e ao receber o retorno da Porto Seguro, o Serviço irá verificar quem realizou a solicitação, e encaminhará para o SIEM ou GA, a comunicação pode ser realizada toda em Json.

## ***Regras de Negócio***

**1.** O serviço irá tratar as informações, envidadas do Siem (Usuário, Senha e Arquivo XML do CT-e a ser Averbado);

Exemplo do Arquivo Json, de Login a ser enviado do SIEM, para ser tratado no Serviço:

```
{
  "mod": "login",
  "user": "USUARIO_CNPJ",
  "pass": "SENHA"
}
```

**1.1** Em seguida o Serviço, trata as informações de Login recebidas do Siem, e envia para o Webservice Porto Seguro.

#### **1.1.1** Endereço da API Rest

**Seguro:** <https://apis.averbeporto.com.br/php/conn.php>  
(TLS 1.0, 1.1, 1.2 e 1.3 \*2) - HTTP/3

- **1 - NÃO utilizar www. ou wws. ou nenhum outro subdomínio para acesso à API.**
- 2 - Defina um user-agent como “Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko/20100101 Firefox/12.0” para acessar apis, isso impedirá que o CF bloqueie seu programa com o erro 403/1010
- 3 - Chamar o endereço da API no navegador irá gerar uma mensagem de erro, pela absoluta falta de parâmetros na comunicação.

**1.2** O login, deve ser realizado com o método **POST** (application/x-www-form-urlencoded), exemplo abaixo de envio do arquivo Json.

```
{
  "mod": "login",
  "comp": 5,
  "user": "USUARIO_CNPJ",
  "pass": "SENHA_Credencial_do_Web_Service_API",
  ["dump": [1,2]] // Opcional.
}
```

*Opcional:*

- - *dump: Pode ser utilizado em qualquer requisição à API para auxiliar na depuração.*
- *Exibe um “dump” de como estão chegando as variáveis enviadas ao sistema pelo usuário.*

- *dump=1: Adiciona a tag “dump” no json de resposta, após todo o processamento normal da requisição.*
- *dump=2: Exibe o Json de resposta apenas com a tag “dump”, antes de qualquer processamento, e aborta o processamento normal.*

**1.2.1** Exemplo de Retorno do Arquivo Json de falha no login, usuário ou senha inválidos ou enviados de maneira incorreta.

```
{
  "success": 1,
  "logout": 1
}
```

**1.2.2** Exemplo de Retorno do Arquivo Json de login, realizado com sucesso.

```
{
  "success": 1,
  "C": {
    "id": "00",
    "userName": "USUARIO",
    "name": "Usuario",
    "email": "usuario@dominio.com",
    "portal_groups_id": "00",
    "type": "U"
  },
  "S": [...]
}
```

Com o Retorno do Login, realizado com sucesso, o usuário poderá realizar a averbação do CT-e.

**1.2.3** Retorno de Login que deverá ser enviado do Serviço para o Siem, quando for realizado com sucesso e quando houver erro no Login.

```
{
  "mod": "loginRet",
  "user": "USUARIO_CNPJ",
  "status": "1", //LOGIN REALIZADO COM SUCESSO
  "cookie": "COOKIE_RETORNADO_SERVICO"
```

```
}

{
  "mod": "loginRet",
  "user": "USUARIO_CNPJ",
  "status": "0" //LOGIN NÃO REALIZADO
}
```

## 2. Upload Arquivo XML do CT-e.

**2.1** Exemplo do Arquivo Json, de Upload a ser enviado do SIEM, para ser tratado no Serviço.

```
{
  "mod": "Upload",
  "CNPJ": "CNPJ_CLIENTE",
  "n_cte": "NUMERO_CTE",
  "cte": "Arquivo.xml",
  "chave": "CHAVE_CTE",
  "cookie": "COOKIE_SALVO_BD"
}
```

**2.2** Exemplo do Arquivo Json, de Upload a ser enviado do Serviço, para o Webservice da Porto Seguro.

```
{
  "comp": 5,
  "mod": "Upload",
  "path": "guarda/php/",
  "recipient": "",
  ["v": N] //N = Versão da API. Parâmetro opcional.
}
```

- 1 - Os COOKIES residem no cabeçalho da resposta HTTP e não no corpo como o JSON.
- 2 - Enviar o arquivo junto aos parâmetros, mesmo como um parâmetro codificado em base64, não irá funcionar. O arquivo deve ser enviado da mesma maneira que um formulário HTML o faria, em multipart.
- 3 - Quando gerada a Credencial de API no módulo Cadastro do Usuário, o cookie de sessão passa a ter validade de 1 ano, ao contrário das 24 horas padrão.

### 2.2.1 Parâmetro recipient (*Opcional, utilizar somente se assim indicado*)

Parâmetro que indica o tipo do(s) arquivo(s) sendo enviado(s). Necessário apenas nos casos indicados pela Seguradora.

**Vazio** = Automático (Recomendado na grande maioria dos casos)

**E** = Embarcador Emitente (Em desuso)

**F** = Fornecedor (Em desuso)

**T** = Transportador (Em desuso)

**D** = Duplo Ramo (As NF-e são averbadas 2 vezes, uma como T e outra como E)

### 2.3 Arquivo Json de retorno, referente ao Upload

```
{
  "success": 1,
  "S": [
    "P": 1, // Processado (xml guardado com sucesso)
    "D": 0, // Duplicado (xml pré-existente)
    "R": 0, // Rejeitado (xml não parece ser do tipo certo)
    "N": 0 // Negado (Não é xml ou zip)
  ]
  "prot": "1234567890123", // Protocolos do XML guardados P
  "prot": ["1234567890123"], // Protocolos dos XMLs (ZIP) guardados P
  "error": [ // Mensagens de erro para os resultados R ou N
    "code": 00,
    "msg": "Mensagem de erro"
  ]
}
```

O *"success": 1*, se refere ao sucesso na comunicação, não necessariamente a efetivação da requisição.

Quando o retorno for:

*"success": 1*

*"P": 1, // Processado (Arquivo Xml guardado com sucesso)*

Com requisição efetivada, o próximo passo é a realização da Consulta.

**2.4** Ao receber o Retorno do upload no Web service, deverá ser enviado ao Siem, um arquivo Json informando se a operação foi realizado com sucesso ou se ocorreu um erro para ser tratado, e visualizado pelo usuário.

#### RETORNO DE REQUISIÇÃO EFETIVADA

```
{  
  "mod": "uploadRet",  
  "error": [  
    "code": 00,  
    "msg": "",  
    "satus": 1 //Informar "1", quando a Requisição de upload for efetivada.  
  ]  
}
```

#### RETORNO DE ERRO

```
{  
  "mod": "uploadRet",  
  "error": [ // Mensagens de erro para os resultados R ou N  
    "code": 00,  
    "msg": "Mensagem de erro",  
    "satus": 0 //Informar "0", quando houver erro no Upload.  
  ]  
}
```

**3.** Com a efetivação da requisição, o serviço realizará automaticamente o próximo passo que é a Consulta, temos duas opções de Consulta disponíveis, abaixo está descrito as duas formas:

### 3.1 Primeiro Tipo de Consulta

**3.1.1** Endereço da API: <https://apis.averbeporto.com.br/php/conn.php>

**3.1.2** Para consulta, fazer o post do cookie de sessão (portal[ses] recebido no login), e dos parâmetros:

*comp=5&mod=Protocolo&path=atwe/php/&*

***chave[]=123456789012345678901234567890123456789012345678901234&***

***chave[]=223456789012345678901234567890123456789012345678901234***

*Neste exemplo estão sendo pesquisadas 2 chaves. Para realizar a pesquisa inversa, basta passar os protocolos nos parâmetros*

***protocolo[]=12345678901234567890123456789012345678901234567890*** e omitir os parâmetros ***chave[]***.

**3.1.3** Arquivo Json de Retorno da Consulta realizada no Web service da Porto Seguro.

Nesta consulta será retornado, a "chave" referente a chave de acesso do CT-e consultado, e o "protocolo" que é referente ao número de protocolo de Averbação.

*"success": 0 -> Erro na consulta.*

*"success": 1 -> Consulta realizada com Sucesso.*

```
{
  "success": 1,
  "S": [{
    "chave": "1234567890123456789012345678901234", //Chave de Acesso CT-e
    "protocolo": "123456789012345678901234567890" //Protocolo de Averbação
  }, {
    "chave": "2234567890123456789012345678901234", //Chave de Acesso CT-e
    "protocolo": "223456789012345678901234567890" //Protocolo de Averbação
  }]
}
```

#### 3.1.4 Arquivo Json com o Retorno para ser enviado ao Siem.

```
{
  "mod": "averbaRet",
  "cnpj": "CNPJ_CLIENTE",
  "numero": "NUMERO_CTE",
  "chave": "CHAVE_CTE",
  "protocolo": "PROTOCOLO_AVERBAÇÃO"
}
```

### 3.2 Segundo Tipo de Consulta

#### 3.2.1 Página de consulta (standalone) - Gera URL GET:

<https://www.averbeporto.com.br/atwe/protocolo.html>

#### 3.2.2 Exemplo de acesso via GET:

```
https://apis.averbeporto.com.br/atwe/php/Protocolo.php?out=json&download=0&chave[ ]=123456789012345678901234567890123456789012345678901234
```

#### 3.2.3 Arquivo Json de Retorno da Consulta realizada na URL da Porto Seguro.

```
{
  "success": 1,
  "S": [{
    "chave": "35220528779662000120579990000004681178902280",
    "protocolo": "0588604232877966200012057999000000468098"
  }]
}
```



### 3.2.4 Arquivo Json com o Retorno para ser enviado ao Siem.

```
{
  "mod": "averbaRet",
  "chave": "CHAVE_CTE",
  "protocolo": "PROTOCOLO_AVERBAÇÃO"
}
```

Fica a critério do desenvolvedor, escolher dentre as duas opções de Consultar Número de Averbação, usando como parâmetro, qual melhor se aplica para o Serviço.

## Questões Técnicas

### 1. Exemplo em PHP (cURL):

```
<?php
/**
 * Open an url on https using curl and return content
 * @param string url The url to open
 * @param string refer Referer (optional)
 * @param mixed usecookie If true, cookie.txt will be used as default, or the usecookie value.
 * @return string
 */
function open_https_url($url,$refer = "", $usecookie = false) {
    if ($usecookie) {
        if (file_exists($usecookie)) {
            if (!is_writable($usecookie)) {
                return "Can't write to $usecookie cookie file, change file permission to 777 or remove read only for windows.";
            }
        } else {
            $usecookie = ($usecookie === true)? "cookie.txt" : $usecookie;
            if (!touch($usecookie)) {
                return "Can't write to $usecookie cookie file, change file permission to 777 or remove read only for windows.";
            }
        }
    }
    $ch = curl_init();
```

```

curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_USERAGENT, "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.0)");
if ($usecookie) {
curl_setopt($ch, CURLOPT_COOKIEJAR, $usecookie);
curl_setopt($ch, CURLOPT_COOKIEFILE, $usecookie);
}
if ($refer != "") {
curl_setopt($ch, CURLOPT_REFERER, $refer );
}
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
return $ch;
}
/**
* Limpeza ao terminar de executar
* Requer $ws
*/
function ws_shutdown(){
global $ws;
if (file_exists($ws['cookie'])) {
unlink($ws['cookie']);
}
}
register_shutdown_function('ws_shutdown');
/**
* Ajax Request
* Requer $ws Global Config (comp/path/cookie)
* aPost (array) json params
* sModule (string) mod (i.e. login/Upload/Retrieve)
* $sConn (string) URI to connect
*/
function websysRequest($aPost, $sModule = 'login', $sConn =
'https://apis.averbeporto.com.br/php/conn.php') {
global $ws;
if (!isset($aPost['comp'])) { $aPost['comp'] = $ws['comp']; }
if (!isset($aPost['path'])) { $aPost['path'] = $ws['path']; } elseif ($aPost['path'] == '') {
unset($aPost['path']); }

```

```

$aPost['mod'] = $sModule;
$ch = open_https_url($sConn, '', $ws['cookie']);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $aPost);
$res = curl_exec($ch);
curl_close($ch);
return $res;
}
?>

```

## 2. Exemplo de Uso:

```

<?php
require_once('func.php');
// Exemplo Config $ws
$ws = array(
    'comp' => 5,
    'path' => 'eguarda/php/',
    'conn' => 'https://apis.averbeporto.com.br/php/conn.php',
    'cookie' => tempnam(sys_get_temp_dir(), 'ws_'),
    'logged' => ''
);
/**
 * Envia arquivo
 *
 * @param {String} Conteudo do arquivo
 * @param {Array} Usuario e senha. Ex.: array('user'=>'USUARIO', 'pass'=>'SENHA', 'path'=>'')
 * @param {String} (optional) Remetente (em caso de email)
 * @return {Array} Retorna resposta do webservice
 */
function sendFile($sFileContent, $aUser, $sRecipient = ''){
    global $ws;
    $file = tmpfile();
    fwrite($file, $sFileContent);
    rewind($file);
    $meta = stream_get_meta_data($file);
    $mime = mime_content_type($meta['uri']);
    $post = array(
        'file' => (version_compare(PHP_VERSION, '5.5') >= 0)? new CURLFile($meta['uri'], $mime) :
        '@'.$meta['uri'].';type='.$mime
    );
}

```

```

);
if ($sRecipient) {
$post['recipient'] = $sRecipient;
}
// Login
if ($ws['logged'] != $aUser['user']) {
$res = json_decode(websysRequest($aUser), true);
if (isset($res['logout']) && $res['logout']) {
//ws_log(' MAIL2EG: ['. $aUser['user'].']: '.posix_getpid().': Falha do login. ');
}
} else {
$res['success'] = $res['C'] = true;
}
// Upload
if ($res['success'] && isset($res['C'])) {
$ws['logged'] = $aUser['user'];
$res = json_decode(websysRequest($post, 'Upload'), true);
}
fclose($file);
return $res;
}
$aUser = array(
' user' => ' USUARIO' ,
' pass' => ' SENHA' ,
' path' => ''
);
$sFileContent = file_get_contents(' ARQUIVO.xml');
$res = sendFile($sFileContent, $aUser);
print_r($res);
?>

```