



**API Web de Integração do  
SophiA Biblioteca Web e Philos**

## Sumário

1.	Introdução .....	4
2.	Requisitos e Instalação .....	4
3.	Estrutura da API Web .....	4
3.1.	Parâmetros do cabeçalho de requisição.....	4
3.2.	Exemplo do cabeçalho de requisição.....	4
4.	Recursos da API Web.....	5
4.1.	GET - /api/autenticacao .....	5
4.1.1.	Respostas HTTP .....	6
4.1.2.	Objetos de retorno .....	6
4.2.	GET - /api/versao .....	6
4.2.1.	Respostas HTTP .....	6
4.2.2.	Objetos de retorno .....	6
4.3.	GET - /api/usuario/{identificacao} .....	6
4.3.1.	Respostas HTTP .....	7
4.3.2.	Objetos de retorno .....	7
4.4.	POST - /api/usuario.....	9
4.4.1.	Respostas HTTP .....	11
4.4.2.	Objetos de retorno .....	12
4.5.	PUT - /api/usuario/{identificacao} .....	12
4.5.1.	Respostas HTTP .....	13
4.5.2.	Objetos de retorno .....	14
4.6.	DELETE - /api/usuario/{identificacao} .....	14
4.6.1.	Respostas HTTP .....	15
4.6.2.	Objetos de retorno .....	15
4.7.	GET - /api/usuario/{identificacao}/nadaconsta .....	15
4.7.1.	Respostas HTTP .....	16
4.7.2.	Objetos de retorno .....	16
4.8.	GET - /api/biblioteca .....	16
4.8.1.	Respostas HTTP .....	16

4.8.2. Objetos de retorno ..... 17

## 1. INTRODUÇÃO

Neste documento são apresentados os recursos da *API Web* de integração para os produtos *Philos* e *SophiA Biblioteca Web* na versão 1, bem como sua forma de uso.

## 2. REQUISITOS E INSTALAÇÃO

A *API Web* de integração não necessita de nenhuma instalação específica, pois se encontra disponível no módulo *Gerenciador* dos produtos *Philos* e *SophiA Biblioteca Web*. Desta forma, seu único requisito, é possuir o módulo *Gerenciador* citado acima.

## 3. ESTRUTURA DA API WEB

A *API Web* de integração é construída sobre a arquitetura *REST*, onde seus recursos são expostos por meio de um endereço (URI) e o método HTTP utilizado para requisição. Através destas duas informações, a *API* consegue localizar o recurso necessário e devolver a informação solicitada ao cliente.

Alguns recursos podem precisar de mais informações que o endereço do recurso e o método HTTP apropriado. Recursos que inserem ou alteram itens nos produtos *Philos* e *SophiA Biblioteca Web*, devem também enviar estes dados à *API Web*. Qualquer dado enviado a um recurso da *API Web* de integração pode ser enviado nos formatos JSON ou XML (fica a cargo do cliente, escolher o formato de dados preferido para seu uso). Os dados devolvidos para os sistemas integrados, também podem ser enviados nos formatos JSON ou XML (mais uma vez, a escolha fica a cargo do cliente). Neste documento, disponibilizamos exemplos de utilização com os dois formatos de dados.

Para que a comunicação com a *API Web* de integração seja estabelecida corretamente, o cabeçalho da requisição deve conter os seguintes parâmetros.

### 3.1. Parâmetros do cabeçalho de requisição

Parâmetro	Valor do parâmetro
User-Agent	Nome do sistema (ex: SophiAGestaoEscolar)
Host	Endereço do servidor que hospeda a API (ex: <a href="http://www.sophiabiblioteca.com.br">http://www.sophiabiblioteca.com.br</a> )
Content-Type	Formato do dado a ser enviado para à API (text/xml ou application/json)
Accept	Formato dos dados devolvidos pela API (text/xml ou application/json)
Token	Token de autenticação do sistema (gerado por um dos recursos da API)

### 3.2. Exemplo do cabeçalho de requisição

Caso o produto *SophiA Gestão Escolar*, deseje obter a lista de bibliotecas ativas do produto *SophiA Biblioteca Web*, hospedado em: <http://www.sophiabiblioteca.com.br>, deverá ser feita uma requisição para o recurso <http://www.sophiabiblioteca.com.br/api/biblioteca>, com método *HTTP GET*, e com um cabeçalho nos seguintes formatos (de acordo com o formato de dados escolhido):

#### XML

```
User-Agent: SophiAGestaoEscolar
Host: http://www.sophiabiblioteca.com.br
```

```
Content-Type: text/xml
Accept: text/xml
Token: bcdc03e458ea4ae5b30908bfa22a9264
```

## JSON

```
User-Agent: SophiAGestaoEscolar
Host: http://www.sophiabiblioteca.com.br
Content-Type: application/json
Accept: application/json
Token: bcdc03e458ea4ae5b30908bfa22a9264
```

## 4. RECURSOS DA API WEB

### 4.1. GET - /api/autenticacao

Este recurso tem a responsabilidade de realizar a autenticação na *API Web* e validar a sua comunicação com os sistemas integrados.

Como mostrado acima, as requisições para a *API Web* devem conter em seu cabeçalho o *Token* de autenticação. E cabe a este recurso gerar este *Token* e devolvê-lo ao sistema requisitante.

Os demais recursos do *API* de integração, esperam por este *Token*, e validam o mesmo para confirmar se o cliente realmente tem autorização para acessar determinado recurso. Só após confirmada esta autorização, o recurso é processado.

Para gerar um *Token*, o sistema integrado deve fazer uma requisição para o recurso de autenticação, passando como parâmetro o usuário e a senha de integração definidos em: *Configurações > Configuração Geral > Integração > Autenticação do Web Service de integração* dos produtos *Philos* ou *SophiA Biblioteca Web* (conforme Figura 1). Este recurso valida os dados informados, e se estiverem corretos, gera um novo *Token*, e devolve o mesmo para o sistema que fez a requisição. O *Token* gerado só pode ser utilizado pelo equipamento que requisitou o mesmo (o *Token* fica atrelado ao endereço IP do equipamento). Todo *Token* gerado é renovado a cada requisição feita a qualquer recurso da *API*. Um *Token* se torna inválido, após 20 minutos de sua criação ou última renovação.

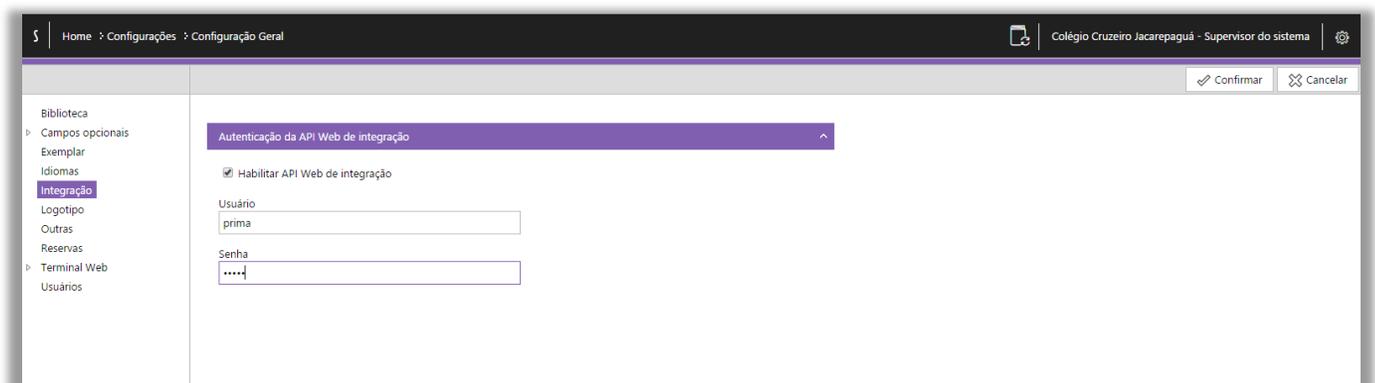


Figura 1. Configuração de autenticação da API Web

### 4.1.1. Respostas HTTP

#### HTTP 404 Not Found.

Requisição realizada sem informar o usuário e/ou senha.

#### HTTP 401 Unathourized.

Requisição realizada com usuário e senha inválidos.

#### HTTP 200 OK.

Autenticação realizada com sucesso, Token gerado e retornado.

### 4.1.2. Objetos de retorno

#### Token XML.

```
<string>6733e272c81244e7aed641fe0d9cd55e</string>
```

#### Token JSON.

```
"c5f0914b4f754ed59080c3d3d2e31174"
```

## 4.2. GET - /api/versao

Este recurso tem a responsabilidade de retornar aos sistemas integrados, em qual versão a *API Web* de integração se encontra. Este é o único recurso que não necessita de um *Token* de autenticação.

### 4.2.1. Respostas HTTP

#### HTTP 200 OK.

Requisição realizada com sucesso e versão retornada.

### 4.2.2. Objetos de retorno

#### Token XML.

```
<string>1.19.0.0</string>
```

#### Token JSON.

```
"1.19.0.0"
```

## 4.3. GET - /api/usuario/{identificacao}

Este recurso disponibiliza aos sistemas integrados, os dados cadastrais de um usuário cadastrado nos produtos *Philos* e *SophiA Biblioteca Web*. Para ter acesso a estes dados, os sistemas integrados devem fazer uma requisição GET ao endereço do recurso, passando o valor do código de identificação do usuário (código ou matrícula).

#### Exemplos.

```
http://www.sophiabiblioteca.com.br/api/usuario/123
```

```
http://www.sophiabiblioteca.com.br/api/usuario/abc
```

O código de identificação do usuário, é definido em *Configurações > Configuração Geral > Usuários > Código de Identificação > Empréstimo* (conforme a Figura 2) nos produtos *Philos* e *SophiA Biblioteca Web*. O campo marcado nesta configuração, é o campo do usuário considerado como identificação para a *API Web* de integração.

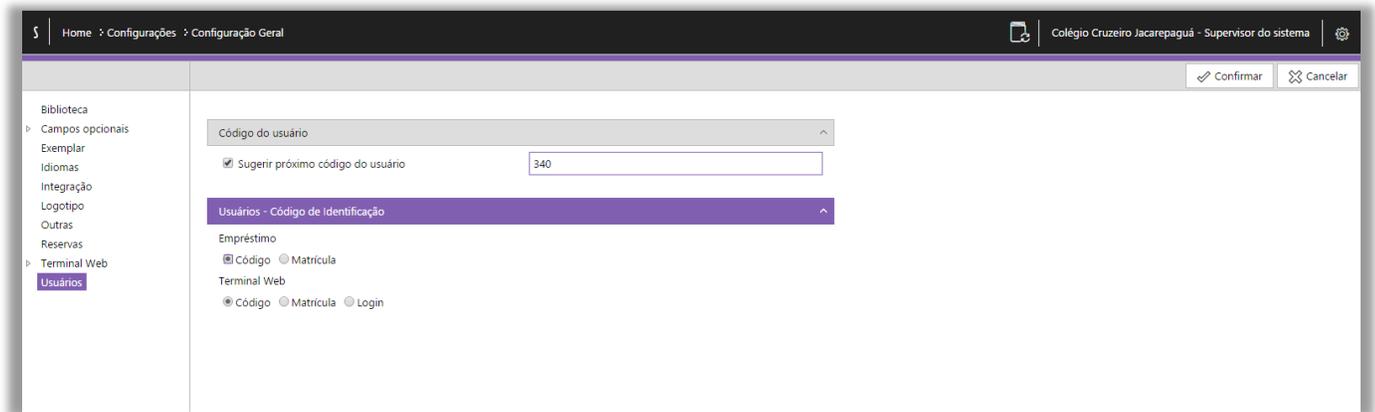


Figura 2. Configuração da identificação do usuário (campo chave)

### 4.3.1. Respostas HTTP

#### HTTP 401 Unauthorized.

Requisição sem Token ou com Token inválido.

#### HTTP 405 Method Not Allowed.

Requisição sem identificação do usuário.

#### HTTP 400 Bad Request.

Requisição com identificação em branco.

#### HTTP 409 Conflict.

Requisição realizada, porém, nenhum usuário foi encontrado.

Requisição realizada, porém, mais de um usuário foi encontrado.

#### HTTP 200 OK.

Requisição realizada, usuário encontrado e retornado.

#### HTTP 500 Internal Server Error.

Algum erro inesperado ocorreu durante a requisição.

### 4.3.2. Objetos de retorno

#### Valores possíveis de retorno da requisição.

0: Erro.

1: Usuário encontrado.

2: Identificação em branco.

3: Usuário não encontrado.

4: Foi localizado mais de um usuário com o código de identificação informado.

#### Retorno da requisição XML.

```
<RetornoApi>
  <Codigo>Código do retorno</Codigo>
  <Descricao>Descrição do retorno</Descricao>
  <MensagemDeErro>Mensagem de erro ocorrido durante a requisição</MensagemDeErro>
</RetornoApi>
```

**Retorno da requisição JSON.**

```
{
  "Codigo":"Código do retorno",
  "Descricao":"Descrição do retorno",
  "MensagemDeErro":"Mensagem de erro ocorrido durante a requisição"
}
```

**Usuário retornado XML.**

```
<Usuario>
  <BairroComercial>Bairro comercial do usuário</BairroComercial>
  <BairroResidencial>Bairro residencial do usuário</BairroResidencial>
  <Biblioteca>Código ou nome da biblioteca do usuário</Biblioteca>
  <CampoOpcional1>Campo opcional 1 do usuário</CampoOpcional1>
  <CampoOpcional2>Campo opcional 2 do usuário</CampoOpcional2>
  <CampoOpcional3>Campo opcional 3 do usuário</CampoOpcional3>
  <CampoOpcional4>Campo opcional 4 do usuário</CampoOpcional4>
  <CampoOpcional5>Campo opcional 5 do usuário</CampoOpcional5>
  <CepComercial>CEP comercial do usuário</CepComercial>
  <CepResidencial>CEP residencial do usuário</CepResidencial>
  <CidadeComercial>Cidade comercial do usuário</CidadeComercial>
  <CidadeResidencial>Cidade residencial do usuário</CidadeResidencial>
  <CodigoUsuario>Código do usuário</CodigoUsuario>
  <Curso>Curso do usuário</Curso>
  <DataBloqueio>Data de bloqueio do usuário</DataBloqueio>
  <DataNascimento>Data de nascimento do usuário</DataNascimento>
  <DataValidade>Data de validade do usuário</DataValidade>
  <EmailComercial>E-mail comercial do usuário</EmailComercial>
  <EmailResidencial>E-mail residencial do usuário</EmailResidencial>
  <Empresa>Empresa do usuário</Empresa>
  <EnderecoComercial>Endereço comercial do usuário</EnderecoComercial>
  <EnderecoResidencial>Endereço residencial do usuário</EnderecoResidencial>
  <Foto>Foto do usuário em formato base 64</Foto>
  <Inativo>Status do usuário (true = inativo, false = ativo)</Inativo>
  <Login>Login do usuário</Login>
  <Matricula>Matrícula do usuário</Matricula>
  <MotivoBloqueio>Motivo do bloqueio do usuário</MotivoBloqueio>
  <Nome>Nome do usuário</Nome>
  <NumeroId>Número do documento de identificação</NumeroId>
  <Periodo>Período do usuário</Periodo>
  <Senha>Senha do usuário</Senha>
  <Serie>Série do usuário</Serie>
  <Sexo>Sexo do usuário (M ou F)</Sexo>
  <TabelaOpcional1>Tabela opcional 1 do usuário</TabelaOpcional1>
  <TabelaOpcional2>Tabela opcional 2 do usuário</TabelaOpcional2>
  <TabelaOpcional3>Tabela opcional 3 do usuário</TabelaOpcional3>
  <TabelaOpcional4>Tabela opcional 4 do usuário</TabelaOpcional4>
  <TelefoneComercial>Telefone comercial do usuário</TelefoneComercial>
  <TelefoneResidencial>Telefone residencial do usuário</TelefoneResidencial>
  <Tipo>Tipo do usuário (ex: aluno, professor, etc)</Tipo>
  <TipoId>Tipo do documento de identificação (ex: rg, cpf, etc)</TipoId>
  <Turma>Turma do usuário</Turma>
</Usuario>
```

**Usuário retornado JSON.**

```
{
  "BairroComercial":"Bairro comercial do usuário",
  "BairroResidencial":"Bairro residencial do usuário",
```

```

"Biblioteca": "Código ou nome da biblioteca do usuário",
"CampoOpcional1": "Campo opcional 1 do usuário",
"CampoOpcional2": "Campo opcional 2 do usuário",
"CampoOpcional3": "Campo opcional 3 do usuário",
"CampoOpcional4": "Campo opcional 4 do usuário",
"CampoOpcional5": "Campo opcional 5 do usuário",
"CepComercial": "CEP comercial do usuário",
"CepResidencial": "CEP residencial do usuário",
"CidadeComercial": "Cidade comercial do usuário",
"CidadeResidencial": "Cidade residencial do usuário",
"CodigoUsuario": "Código do usuário",
"Curso": "Curso do usuário",
"DataBloqueio": "Data de bloqueio do usuário",
"DataNascimento": "Data de nascimento do usuário",
"DataValidade": "Data de validade do usuário",
"EmailComercial": "E-mail comercial do usuário",
"EmailResidencial": "E-mail residencial do usuário",
"Empresa": "Empresa do usuário",
"EnderecoComercial": "Endereço comercial do usuário",
"EnderecoResidencial": "Endereço residencial do usuário",
"Foto": "Foto do usuário em formato base 64",
"Inativo": "Status do usuário (true = inativo, false = ativo)",
>Login": "Login do usuário",
"Matricula": "Matrícula do usuário",
"MotivoBloqueio": "Motivo do bloqueio do usuário",
"Nome": "Nome do usuário",
"NumeroId": "Número do documento de identificação",
"Periodo": "Período do usuário",
"Senha": "Senha do usuário",
"Serie": "Série do usuário",
"Sexo": "Sexo do usuário (M ou F)",
"TabelaOpcional1": "Tabela opcional 1 do usuário",
"TabelaOpcional2": "Tabela opcional 2 do usuário",
"TabelaOpcional3": "Tabela opcional 3 do usuário",
"TabelaOpcional4": "Tabela opcional 4 do usuário",
"TelefoneComercial": "Telefone comercial do usuário",
"TelefoneResidencial": "Telefone residencial do usuário",
"Tipo": "Tipo do usuário (ex: aluno, professor, etc)",
"TipoId": "Tipo do documento de identificação (ex: rg, cpf, etc)",
"Turma": "Turma do usuário"
}

```

#### 4.4. POST - /api/usuario

Este recurso disponibiliza aos sistemas integrados, incluir um novo usuário nos produtos *Philos* e *SophiA Biblioteca Web*. Para realizar inclusão do usuário, além de efetuar a requisição POST ao endereço do recurso, o sistema integrado deve enviar um XML ou JSON contendo os dados do usuário (o XML ou JSON enviados, deve seguir o mesmo formato apresentado no recurso *GET - /api/usuario*).

Para inclusão de usuário, é obrigatório que as propriedades *Nome*, *Tipo* e *Biblioteca* sejam enviados à *API Web*. Além destas, as propriedades definidas como códigos de identificação do usuário (para empréstimo e para acesso ao terminal), também são obrigatórias. As propriedades definidas como identificadores, podem ser encontradas nas sessões *Configurações > Configuração Geral > Usuários > Código de Identificação > Empréstimo e Terminal Web* (conforme a Figura 2).

Todas as outras propriedades, são opcionais e podem ser omitidas do XML/JSON de envio, porém no caso do XML, a ordem destes itens não pode ser alterada.

Algumas propriedades, possuem regras específicas para garantir o funcionamento do recurso de inclusão de usuário. São eles:

- As propriedades *DataNascimento*, *DataBloqueio* e *DataValidade*, se informadas, devem seguir o formato: dd/mm/aaaa (ex: 31/12/2000). Além do formato, o valor das propriedades deve estar entre 01/01/1900 e 31/12/9999.
- A propriedade *CodigoUsuario*, se informada deve conter um valor numérico. Caso *CodigoUsuario*, seja utilizada como código de identificação do usuário, o recurso não incluirá o usuário se o valor enviado já for utilizado por outro usuário.
- Caso a propriedade *Matricula* seja utilizada como código de identificação do usuário, o recurso não incluirá o usuário se o valor enviado já for utilizado por outro usuário.
- Caso a propriedade *Login* seja utilizada como código de identificação do usuário, o recurso não incluirá o usuário se o valor enviado já for utilizado por outro usuário.
- A propriedade *Inativo*, se informada deve conter o valor "True" ou "False".
- A propriedade *Sexo*, se informada deve conter o valor "M" ou "F".
- O valor da propriedade *Biblioteca* deve corresponder ao código ou nome de uma biblioteca ativa nos produtos *Philos* e *SophiA Biblioteca Web*. Caso contrário, o recurso não incluirá o usuário.
- Todas as propriedades são tratadas como texto, e devem respeitar suas quantidades máximas de caracteres:

<b>Propriedade</b>	<b>Máximo de caracteres</b>
BairroComercial	100
BairroResidencial	100
Biblioteca	60
CampoOpcional1	252
CampoOpcional2	252
CampoOpcional3	252
CampoOpcional4	252
CampoOpcional5	252
CepComercial	9
CepResidencial	9
CidadeComercial	252
CidadeResidencial	252
CodigoUsuario	15
Curso	252
DataBloqueio	10
DataNascimento	10

DataValidade	10
EmailComercial	252
EmailResidencial	252
Empresa	100
EnderecoComercial	252
EnderecoResidencial	252
Foto	Sem limites
Inativo	5
Login	100
Matricula	20
MotivoBloqueio	30
Nome	252
NumeroId	20
Periodo	100
Senha	30
Serie	100
Sexo	1
TabelaOpcional1	252
TabelaOpcional2	252
TabelaOpcional3	252
TabelaOpcional4	252
TelefoneComercial	252
TelefoneResidencial	252
Tipo	50
TipoId	252
Turma	100

- Caso alguma propriedade seja enviada com uma quantidade de caracteres acima do permitido, a *API Web* irá desconsiderar os caracteres excedentes. Exemplo: caso a propriedade *CepResidencial* seja enviada com 15 caracteres, a *API Web* irá considerar apenas os primeiros 9 caracteres da propriedade.

#### 4.4.1. Respostas HTTP

##### **HTTP 401 Unauthorized.**

Requisição sem Token ou com Token inválido.

##### **HTTP 400 BadRequest.**

Campos obrigatórios não informados.

Valor inválido para propriedades de datas, numéricas e com valores pré-definidos.

##### **HTTP 409 Conflict.**

Requisição realizada, porém código de identificação informado já está em uso.  
 Requisição realizada, porém biblioteca informada é inválida.

**HTTP 500 Internal Server Error.**

XML/JSON não foi enviado ou é inválido.  
 Algum erro inesperado ocorreu durante a requisição.

**HTTP 201 Created.**

Requisição realizada e usuário criado com sucesso.

#### 4.4.2. Objetos de retorno

**Valores possíveis de retorno da requisição.**

0: Erro.  
 1: Usuário incluído.  
 2: Nome ou tipo do usuário em branco.  
 3: Código do usuário em branco.  
 4: Matrícula do usuário em branco.  
 5: Valor inválido ou fora do formato esperado.  
 6: Código do usuário duplicado.  
 7: Matrícula do usuário duplicada.  
 8: Login do usuário duplicado.  
 9: Login do usuário em branco.  
 10: Biblioteca em branco.  
 11: Biblioteca inválida.

**Retorno da requisição XML.**

```
<RetornoApi>
  <Codigo>Código do retorno</Codigo>
  <Descricao>Descrição do retorno</Descricao>
  <MensagemDeErro>Mensagem de erro ocorrido durante a requisição</MensagemDeErro>
</RetornoApi>
```

**Retorno da requisição JSON.**

```
{
  "Codigo": "Código do retorno",
  "Descricao": "Descrição do retorno",
  "MensagemDeErro": "Mensagem de erro ocorrido durante a requisição"
}
```

#### 4.5. PUT - /api/usuario/{identificacao}

Este recurso disponibiliza aos sistemas integrados, alterar o cadastro de um usuário nos produtos *Philos* e *SophiA Biblioteca Web*. Para alterar os dados do usuário, o sistema integrado deve fazer uma requisição PUT ao endereço do recurso, passando o valor do código de identificação do usuário (código ou matrícula).

**Exemplos.**

```
http://www.sophiabiblioteca.com.br/api/usuario/123
http://www.sophiabiblioteca.com.br/api/usuario/abc
```

Além destas informações, o sistema integrado deve enviar um XML ou JSON contendo as propriedades a serem alteradas do usuário. Não há a necessidade de enviar todas, basta enviar apenas as propriedades alteradas. Caso deseje apagar o valor de uma propriedade, basta enviar a mesma vazia. As propriedades não enviadas mantêm seus valores atuais.

Para todos as propriedades enviadas, vale ressaltar:

- A ordem das propriedades do XML de envio deve respeitar a ordem apresentada no recurso *GET - /api/usuario* (a ordem não interfere no uso do JSON).
- As propriedades *DataNascimento*, *DataBloqueio*, *DataValidade*, *Sexo* e *Biblioteca* se informadas, devem seguir as mesmas validações descritas no recurso *POST - /api/usuario*.
- As propriedades *CodigoUsuario*, *Matricula* e *Login*, se informados e forem usados como código de identificação do usuário, devem seguir as mesmas validações descritas no recurso *POST - /api/usuario*.
- Todas as propriedades também devem ser tratadas como texto e seguir os limites de caracteres apresentados no recurso *POST - /api/usuario*. Em caso de propriedades que ultrapassam o limite de caracteres; os caracteres excedentes também serão desconsiderados (da mesma forma que o recurso de inclusão).
- Para ativar um usuário, basta enviar "False" na propriedade *Inativo*.
- Para inativar um usuário, basta enviar "True" na propriedade *Inativo*. Caso o usuário em questão esteja ativo nos produtos *Philos* e *SophiA Biblioteca Web*, o recurso só prosseguirá com a alteração se o usuário não possuir nenhuma das pendências abaixo:
  - Circulações em aberto.
  - Serviços e taxas em aberto.
  - Reservas aguardando liberação.
- Para bloquear um usuário, basta enviar uma data válida na propriedade *DataBloqueio*.
- Para desbloquear um usuário, basta enviar a propriedade *DataBloqueio* vazia.

#### 4.5.1. Respostas HTTP

##### **HTTP 401 Unauthorized.**

Requisição sem Token ou com Token inválido.

##### **HTTP 405 Method Not Allowed.**

Requisição sem identificação do usuário.

##### **HTTP 400 BadRequest.**

Requisição com identificação em branco.

Campos obrigatórios informados em branco.

Valor inválido para propriedades de datas, numéricas e com valores pré-definidos.

##### **HTTP 409 Conflict.**

Requisição realizada, porém, nenhum usuário foi encontrado.

Requisição realizada, porém, mais de um usuário foi encontrado.

Requisição realizada, porém, código de identificação informado já está em uso.

Requisição realizada, porém, biblioteca informada é inválida.

Requisição realizada, porém, usuário possui pendências e não pode ser inativado.

##### **HTTP 500 Internal Server Error.**

XML/JSON não foi enviado ou é inválido.

Algum erro inesperado ocorreu durante a requisição.

**HTTP 200 OK.**

Requisição realizada e usuário atualizado com sucesso.

### 4.5.2. Objetos de retorno

**Valores possíveis de retorno da requisição.**

0: Erro.  
 1: Usuário alterado.  
 2: Código de identificação em branco.  
 3: Usuário não encontrado.  
 4: Usuário não pode ficar inativo pois possui pendências.  
 5: Nome ou tipo de usuário em branco.  
 6: Biblioteca em branco.  
 7: Biblioteca inválida.  
 8: Código do usuário em branco.  
 9: Matrícula do usuário em branco.  
 10: Login do usuário em branco.  
 11: Valor inválido ou fora do formato esperado.  
 12: Código do usuário duplicado.  
 13: Matrícula do usuário duplicada.  
 14: Login do usuário duplicado.  
 15: Foi localizado mais de um usuário com o código de identificação informado.

**Retorno da requisição XML.**

```
<RetornoApi>
  <Codigo>Código do retorno</Codigo>
  <Descricao>Descrição do retorno</Descricao>
  <MensagemDeErro>Mensagem de erro ocorrido durante a requisição</MensagemDeErro>
</RetornoApi>
```

**Retorno da requisição JSON.**

```
{
  "Codigo": "Código do retorno",
  "Descricao": "Descrição do retorno",
  "MensagemDeErro": "Mensagem de erro ocorrido durante a requisição"
}
```

## 4.6. DELETE - /api/usuario/{identificacao}

Este recurso permite aos sistemas integrados, remover os usuários dos produtos *Philos* e *SophiA Biblioteca Web*. Para remover um usuário, os sistemas integrados devem fazer uma requisição DELETE ao endereço do recurso, passando o valor do código de identificação do usuário (código ou matrícula).

**Exemplos.**

```
http://www.sophiabiblioteca.com.br/api/usuario/123
http://www.sophiabiblioteca.com.br/api/usuario/abc
```

Os usuários só podem ser removidos se não possuírem nenhuma das pendências abaixo:

- Circulações em aberto.
- Serviços e taxas em aberto.

### 4.6.1. Respostas HTTP

#### HTTP 401 Unauthorized.

Requisição sem Token ou com Token inválido.

#### HTTP 405 Method Not Allowed.

Requisição sem identificação do usuário.

#### HTTP 400 Bad Request.

Requisição com identificação em branco.

#### HTTP 409 Conflict.

Requisição realizada, porém, nenhum usuário foi encontrado.

Requisição realizada, porém, mais de um usuário foi encontrado.

Requisição realizada, porém, usuário possui pendências e não pode ser removido.

#### HTTP 500 Internal Server Error.

Algum erro inesperado ocorreu durante a requisição.

#### HTTP 200 OK.

Requisição realizada e usuário removido com sucesso.

### 4.6.2. Objetos de retorno

#### Valores possíveis de retorno da requisição.

0: Erro.

1: Usuário excluído.

2: Usuário não pode ser excluído, pois possui circulações em aberto.

3: Usuário não pode ser excluído, pois possui débitos na biblioteca.

4: Usuário não encontrado.

5: Código de identificação em branco.

6: Foi localizado mais de um usuário com o código de identificação informado.

#### Retorno da requisição XML.

```
<RetornoApi>
  <Codigo>Código do retorno</Codigo>
  <Descricao>Descrição do retorno</Descricao>
  <MensagemDeErro>Mensagem de erro ocorrido durante a requisição</MensagemDeErro>
</RetornoApi>
```

#### Retorno da requisição JSON.

```
{
  "Codigo": "Código do retorno",
  "Descricao": "Descrição do retorno",
  "MensagemDeErro": "Mensagem de erro ocorrido durante a requisição"
}
```

## 4.7. GET - /api/usuario/{identificacao}/nadaconsta

Este recurso permite aos sistemas integrados, verificar se um usuário em específico possui pendências (circulações em aberto e serviços/taxas em aberto) nos produtos *Philos* e *SophiA Biblioteca Web*. Para fazer esta verificação, os sistemas integrados devem fazer uma requisição GET ao endereço do recurso, passando o valor do código de identificação do usuário (código ou matrícula).

#### Exemplos.

```
http://www.sophiabiblioteca.com.br/api/usuario/123/nadaconsta
http://www.sophiabiblioteca.com.br/api/usuario/abc/nadaconsta
```

### 4.7.1. Respostas HTTP

#### HTTP 401 Unauthorized.

Requisição sem Token ou com Token inválido.

#### HTTP 405 Method Not Allowed.

Requisição sem identificação do usuário.

#### HTTP 400 Bad Request.

Requisição com identificação em branco.

#### HTTP 409 Conflict.

Requisição realizada, porém, nenhum usuário foi encontrado.

Requisição realizada, porém, mais de um usuário foi encontrado.

Requisição realizada e usuário possui pendências.

#### HTTP 200 OK.

Requisição realizada e usuário não tem pendências.

### 4.7.2. Objetos de retorno

#### Valores possíveis de retorno da requisição.

- 0: Erro.
- 1: Usuário não deve nada à biblioteca.
- 2: Usuário está devedor.
- 3: Código de identificação em branco.
- 4: Usuário não encontrado.
- 5: Foi localizado mais de um usuário com o código de identificação informado.

#### Retorno da requisição XML.

```
<RetornoApi>
  <Codigo>Código do retorno</Codigo>
  <Descricao>Descrição do retorno</Descricao>
  <MensagemDeErro>Mensagem de erro ocorrido durante a requisição</MensagemDeErro>
</RetornoApi>
```

#### Retorno da requisição JSON.

```
{
  "Codigo": "Código do retorno",
  "Descricao": "Descrição do retorno",
  "MensagemDeErro": "Mensagem de erro ocorrido durante a requisição"
}
```

## 4.8. GET - /api/biblioteca

Este recurso permite aos sistemas integrados, obter o código e o nome das bibliotecas ativas nos produtos *Philos* e *SophiA Biblioteca Web*. Para obter a lista de bibliotecas ativas, o sistema integrado deve fazer uma requisição GET ao endereço do recurso.

### 4.8.1. Respostas HTTP

#### HTTP 401 Unauthorized.

Requisição sem Token ou com Token inválido.

**HTTP 200 OK.**

Requisição realizada e bibliotecas retornadas.

#### **4.8.2. Objetos de retorno**

##### **Lista de bibliotecas XML.**

```
<ArrayOfBiblioteca>
  <Biblioteca>
    <Codigo>Código da biblioteca</Codigo>
    <Nome>Nome da biblioteca</Nome>
  </Biblioteca>
  <Biblioteca>
    <Codigo>Código da biblioteca</Codigo>
    <Nome>Nome da biblioteca</Nome>
  </Biblioteca>
  <Biblioteca>
    <Codigo>Código da biblioteca</Codigo>
    <Nome>Nome da biblioteca</Nome>
  </Biblioteca>
</ArrayOfBiblioteca>
```

##### **Lista de bibliotecas JSON.**

```
[
  {"Codigo":"Código da biblioteca","Nome":"Nome da biblioteca"},
  {"Codigo":"Código da biblioteca","Nome":"Nome da biblioteca"},
  {"Codigo":"Código da biblioteca","Nome":"Nome da biblioteca"}
]
```