

Projeto: **FS007525 – Integração SENAI APP**

Emissão: **16/03/2016**

Analista Especificador: **Maria Fernanda Notini de Castro**

1. Dados do Projeto

Cliente	Projeto N°	Projeto Descrição
SENAI-DN	FS007525	Integração SENAI APP

2. Síntese

Após levantamento realizado entre CNI e Fábrica de Software, foi definida a necessidade de integrar o SENAI APP com o TOTVS SGE via Web Service de forma que o Professor, a partir do SENAI APP, tenha condições de recuperar no TOTVS SGE as turmas/disciplinas que atuará no período e os alunos matriculados na turma/disciplina.

Este documento tem como objetivo descrever de forma detalhada a solução a ser implementada pela TOTVS.

3. Requisitos

3.1. Escopo

Req. 01 – Criação de webmétodo de consulta para retornar turmas/disciplinas do professor

Req. 02 – Criação de webmétodo de consulta para retornar os alunos matriculados na turma/disciplina

3.2. Restrições

Não estão contemplados nesta proposta:

- i. Funcionalidades não contempladas no escopo deste documento ou qualquer informação que esteja implícita na documentação.
- ii. Desenvolvimento de help on-line.
- iii. Criação e manutenção de usuários do sistema.
- iv. Elaboração ou alteração de qualquer relatório.
- v. Desenvolvimento de rotinas ou programas para carga de dados nos demais sistemas que proverão interfaces com o projeto desenvolvido.
- vi. Seleção, instalação e configuração de infraestrutura, banco de dados, hardware e ambiente operacional necessário à implementação do projeto desenvolvido, seja em ambiente de testes ou ambiente oficial.
- vii. Manutenções em outros Sistemas não descritos na Proposta.
- viii. Procedimentos de backup do sistema ou banco de dados.
- ix. Disponibilidade do projeto desenvolvido em outras línguas.
- x. Instalação do sistema em outras localidades.

3.3. Infraestrutura e Ambiente de Desenvolvimento

O presente documento considera os itens abaixo como premissas técnicas referentes à infraestrutura e ambiente da sua empresa. Todo o projeto de desenvolvimento será feito tomando como base as informações colocadas abaixo:

- Sistema Base: TOTVS Educacional
- Base de Dados: SQL
- Linguagem: WEB
- Versão ERP: 11.83
- Ambiente 3 camadas: Não

4. Modelo de Solução

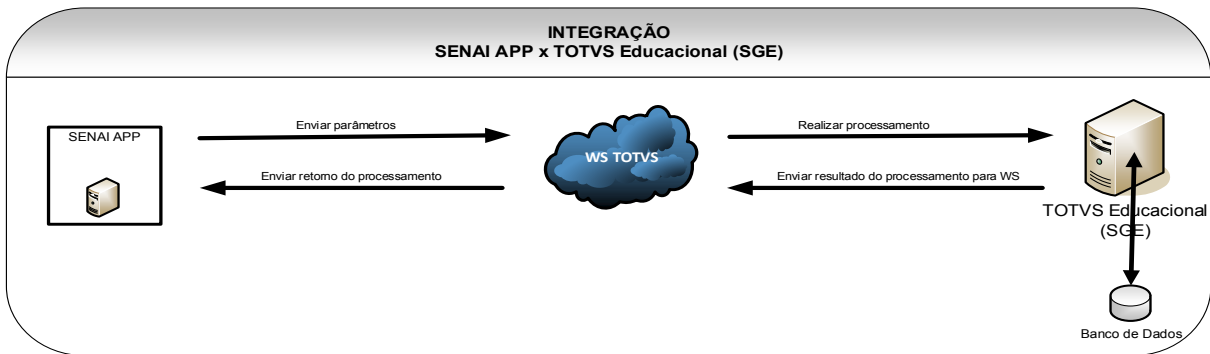
Foi identificada pelo cliente CNI a necessidade de integrar SENAI APP com o TOTVS Educacional (SGE).

Para atender esta demanda, será desenvolvida uma aplicação WebService de responsabilidade da TOTVS onde serão disponibilizados serviços de consulta a serem consumidos pelo SENAI APP.

O SENAI APP irá integrar com TOTVS Educacional (SGE) tendo como base a utilização de WebServices, ou seja, serviços de rede que possibilitam a comunicação entre sistemas criados em diferentes arquiteturas através de uma linguagem de comunicação universal.

Desta forma, o SENAI APP deverá consumir WebServices disponibilizados pela TOTVS.

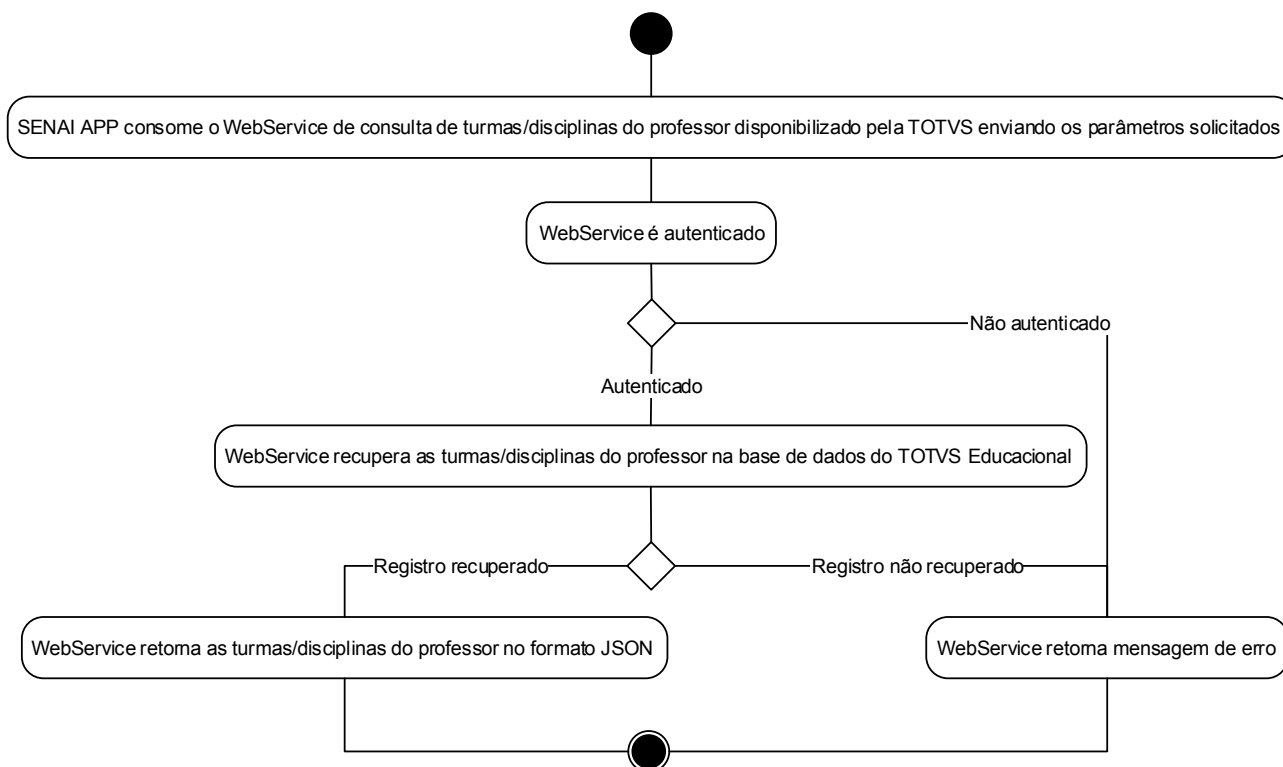
Para que o processo funcione de acordo com o modelo proposto, o SENAI deverá estar ciente deste cenário:



Observação: O SENAI deverá disponibilizar um servidor Web com a Biblioteca da RM instalada e o IIS (Internet Information Services) configurado, viabilizando a implementação da solução proposta. As informações do IIS serão de responsabilidade do cliente. A TOTVS não será responsável por manter os webservices On Line.

Req. 01 – Criação de webmétodo de consulta para retornar turmas/disciplinas do professor

- Fluxograma do Processo



- Descrição do Processo

- SENAI APP consome o Webservice de consulta de turmas/disciplinas do professor disponibilizado pela TOTVS enviando os parâmetros solicitados
- WebService é autenticado
 - Se autenticado, o passo 3 é executado
 - Se não autenticado, WebService retorna mensagem de erro e o passo 4 é executado
- WebService recupera as turmas/disciplinas do professor na base de dados do TOTVS Educacional
 - Se recuperado registro, WebService retorna as turmas/disciplinas do professor no formato JSON
 - Se nenhum registro recuperado, WebService retorna mensagem de erro
- Processo é finalizado

- Detalhamento de Cálculos

Não se aplica.

- Detalhamento de Processos Específicos

O objetivo deste requisito é a criação de um Webservice de consulta que retornará no formato JSON as turmas/disciplinas ativas e vigentes do professor passado como parâmetro.

O Web Service será implementado utilizando arquitetura REST, possibilitando a troca de mensagens em formato JSON para uso em ambientes distribuídos e que atende a diversas plataformas com mensagens personalizadas.

Para autenticação do usuário no TOTVS Educacional, serão criados no WebConfig do Web Service três parâmetros:

CodUsuario	Usuário do TOTVS Educacional responsável pela autenticação
Senha	Senha do usuário do TOTVS Educacional responsável pela autenticação
Token	Token (fixo) utilizado para validação do token enviado pelo SENAI APP

Quando uma requisição ao método for realizada, o Web Service realizará duas autenticações:

- **Validação do Token:** será validado o Token enviado pelo SENAI APP como parâmetro com a informação existente no WebConfig.
- **Validação do Usuário (SGE):** será validado se o usuário e a senha do WebConfig é um ogin válido no TOTVS Educacional.

O Webservice receberá como parâmetro o Token e CPF do professor.

O Webservice irá recuperar o professor referente ao CPF passado como parâmetro. Para recuperar o professor referente ao CPF passado como parâmetro, o Webservice irá filtrar na tabela de pessoa (PPESSOA) a pessoa referente ao CPF recebido (PPESSOA.CPF = cpfProfessor) e realizar o relacionamento entre a tabela de professor (SPROFESSOR) e o código da pessoa (SPROFESSOR.CODPESSOA = PPESSOA.CODIGO).

Se não existir professor para o CPF informado, será retornada a mensagem de erro “Não existe no SGE professor cadastrado para o CPF informado”.

Uma vez recuperado o código do professor (SPROFESSOR.CODPROF), o Webservice irá filtrar na tabela de turma do professor (SPROFESSORTURMA) as turmas/disciplinas ativas e vigentes na data atual (SPROFESSORTURMA.IDTURMADISC) vinculadas ao professor (SPROFESSORTURMA.CODPROF = código do professor).

- A turma/disciplina será considerada ativa se estiver setada como “Ativa” (STURMADISC.ATIVA = S).
- A turma/disciplina será considerada vigente se a data atual estiver entre a data inicial (STURMADISC.DTINICIAL) e a data final (STURMADISC.DTFINAL) da turma/disciplina.

Se não existir turma/disciplina ativa e vigente para o professor, será retornada a mensagem de erro “Não existe no SGE turma/disciplina ativa e vigente para o professor.”.

getTurmasDisciplinasProfessor

Exemplo de assinatura do serviço:

```
public string getTurmasDisciplinasProfessor (string codUsuario, string hash, string cpfProfessor)
```

Parâmetro(s) de entrada:

Parâmetro	Tipo	Observação
token	String	Código Hash gerado pelo TOTVS
cpfProfessor	String	CPF do professor

Parâmetro(s) de saída:

Parâmetro	Tipo	Observação
descMatrizCurricular	String	Descrição da matriz curricular vinculada à turma/disciplina (SGRADE.DESCRICAO) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.IDHABILITACAOFILIAL -> SHABILITACAOFILIAL.IDHABILITACAOFILIAL SHABILITACAOFILIAL.CODGRADE -> SGRADE.CODCURSO SCURSO.CODCURSO -> SCURSO.NOME
codTurma	String	Código da turma (STURMADISC.CODTURMA) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.CODTURMA
nomeTurma	String	Nome da turma (STURMA.NOME) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.CODTURMA -> STURMA.CODTURMA STURMA.NOME
idTurmaDisc	String	Identificador da turma/disciplina (SPROFESSORTURMA.IDTURMADISC)
codDisc	String	Código da disciplina (STURMADISC.CODDISC) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.CODDISC
descDisc	String	Descrição da disciplina (SDISCIPLINA.NOME) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.CODDISC -> SDISCIPLINA.CODDISC SDISCIPLINA.NOME
dtInicioTurmaDisc	Date	Data inicial da turma/disciplina (STURMADISC.DTINICIAL) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC STURMADISC.DTINICIAL
dtFimTurmaDisc	Date	Data final da turma/disciplina (STURMADISC.DTFINAL) STURMAPROFESSOR.IDTURMADISC -> STURMADISC.IDTURMADISC

	STURMADISC.DTFINAL
--	--------------------

Se erro na autenticação, o Webservice irá retornar uma mensagem de erro e será finalizado.

Exemplo de mensagem com erro:

```
{
  "sucesso": "0",
  "mensagem": "Erro ao buscar Turmas do Professor 'Professor Senai 01':
  Usuário não autenticado.",
  "turmas": []
}
```

Se autenticado, o Webservice irá realizar a consulta na base de dados do TOTVS Educacional e retornar as turmas/disciplinas do professor passado como parâmetro. O retorno será no formato **JSON**.

Exemplo de mensagem com sucesso:

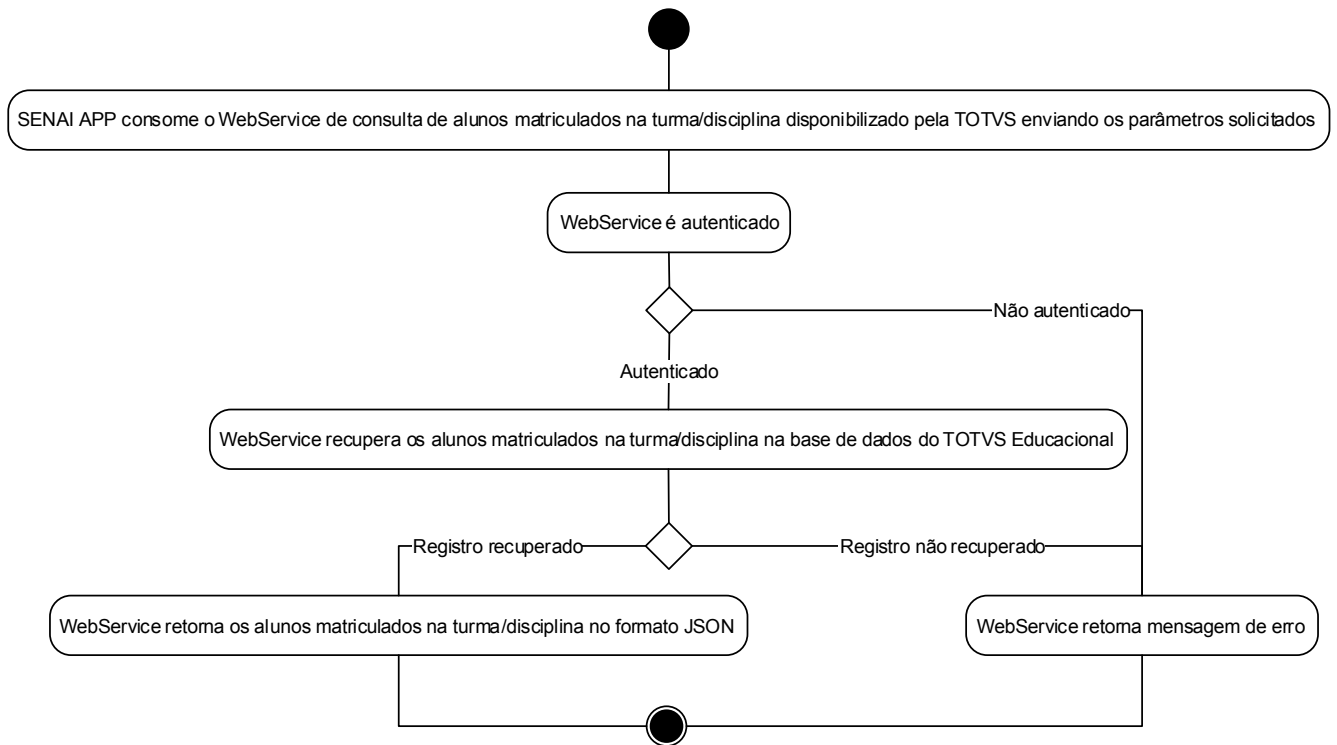
```
{
  "sucesso": "1",
  "mensagem": "Registros recuperados com sucesso!",
  "turmas":
  [
    {
      "descMatrizCurricular": "Técnico em Celulose e Papel",
      "codTurma": "TEC-P-000208/2016",
      "nomeTurma": "TCP - SISUTEC - SUB 1 4º",
      "idTurmaDisc": "37362",
      "codDisc": "TEC.036.0001",
      "descDisc": "Acabamento e Conversão",
      "dtInicioTurmaDisc": "2016-05-25 00:00:00.000",
      "dtFimTurmaDisc": "2016-07-05 00:00:00.000"
    },
    {
      "descMatrizCurricular": "Técnico em Celulose e Papel",
      "codTurma": "TEC-P-000209/2016",
      "nomeTurma": "TCP - PRONATEC 5",
      "idTurmaDisc": "37368",
      "codDisc": "TEC.036.0001",
      "descDisc": "Acabamento e Conversão",
      "dtInicioTurmaDisc": "2016-05-01 00:00:00.000",
      "dtFimTurmaDisc": "2016-06-05 00:00:00.000"
    },
    {
      "descMatrizCurricular": "Técnico em Celulose e Papel",
      "codTurma": "TEC-N-000202/2016",
      "nomeTurma": "Celulose e Papel PISA/STORA",
      "idTurmaDisc": "35960",
      "codDisc": "TEC.036.0001",
      "descDisc": "Acabamento e Conversão",
      "dtInicioTurmaDisc": "2016-02-11 00:00:00.000",
      "dtFimTurmaDisc": "2016-07-07 00:00:00.000"
    }
  ]
}
```

- Protótipo

Não se aplica.

Req. 02 – Criação de web método de consulta para retornar os alunos matriculados na turma/disciplina

- Fluxograma do Processo



- Descrição do Processo

- SENAI APP consome o Webservice de consulta de alunos matriculados na turma/disciplina disponibilizado pela TOTVS enviando os parâmetros solicitados
- WebService é autenticado
 - Se autenticado, o passo 3 é executado
 - Se não autenticado, WebService retorna mensagem de erro e o passo 4 é executado
- WebService recupera os alunos matriculados na turma/disciplina na base de dados do TOTVS Educacional
 - Se recuperado registro, WebService retorna os alunos matriculados na turma/disciplina no formato JSON
 - Se nenhum registro recuperado, WebService retorna mensagem de erro
- Processo é finalizado

- Detalhamento de Cálculos

Não se aplica.

- Detalhamento de Processos Específicos

O objetivo deste requisito é a criação de um Webservice de consulta que retornará no formato JSON os alunos matriculados na turma/disciplina passada como parâmetro.

O Web Service será implementado utilizando arquitetura REST, possibilitando a troca de mensagens em formato JSON para uso em ambientes distribuídos e que atende a diversas plataformas com mensagens personalizadas.

Para autenticação do usuário no TOTVS Educacional, serão criados no WebConfig do Web Service três parâmetros:

CodUsuario	Usuário do TOTVS Educacional responsável pela autenticação
Senha	Senha do usuário do TOTVS Educacional responsável pela autenticação
Token	Token (fixo) utilizado para validação do token enviado pelo SENAI APP

Quando uma requisição ao método for realizada, o Web Service realizará duas autenticações:

- **Validação do Token:** será validado o Token enviado pelo SENAI APP pelo parâmetro com a informação existente no WebConfig.
- **Validação do Usuário (SGE):** será validado se o usuário e a senha do WebConfig é login válido no TOTVS Educacional.

O Webservice receberá como parâmetro o Token e identificador da turma/disciplina.

O Webservice irá filtrar na tabela de matrícula na disciplina (SMATRICULA) os alunos matriculados na turma/disciplina passada como parâmetro (SMATRICULA.IDTURMADISC = idTurmaDisc).

- Serão considerados como matriculados na turma/disciplina apenas os alunos cuja situação de matrícula esteja com a flag 'É uma disciplina em curso' marcada (SSTATUS.DIEMCURSO = S onde SSTATUS.CODSTATUS = SMATRICULA.CODSTATUS).

Se não existir aluno matriculado na turma/disciplina, será retornada a mensagem de erro "Não existe no SGE aluno matriculado na turma/disciplina."

getAlunosTurmaDisciplina

Exemplo de assinatura do serviço:

```
public string getAlunosTurmaDisciplina (string token, int idTurmaDisc)
```

Parâmetro(s) de entrada:

Parâmetro	Tipo	Observação
token	String	Código Hash gerado pelo TOTVS
idTurmaDisc	Int	Identificador da turma/disciplina

Parâmetro(s) de saída:

Parâmetro	Tipo	Observação
raAluno	Int	RA do aluno

		<p>(SHABILITACAOALUNOCOMPL.CODMATRICULASCOPI ou SHABILITACAOALUNO.RA + STURMA.IDFT)</p> <p>SMATRICULA.RA -> SHABILITACAOALUNOCOMPL.RA</p> <p>SHABILITACAOALUNOCOMPL.CODMATRICULASCOPI</p> <p>Se o campo acima não estiver preenchido:</p> <p>SMATRICULA.RA -> SHABILITACAOALUNO.RA + STURMA.IDFT</p>
nomeAluno	String	<p>Nome do aluno (PPESSOA.NOME)</p> <p>SMATRICULA.RA -> SALUNO.RA</p> <p>SALUNO.CODPESSOA -> PPESSOA.CODIGO</p> <p>PPESSOA.NOME</p>
dtNascAluno	Data	<p>Data de nascimento do aluno (PPESSOA.DTNASCIMENTO)</p> <p>SMATRICULA.RA -> SALUNO.RA</p> <p>SALUNO.CODPESSOA -> PPESSOA.CODIGO</p> <p>PPESSOA.DTNASCIMENTO</p>
sexoAluno		<p>Sexo (PPESSOA.SEXO)</p> <p>SMATRICULA.RA -> SALUNO.RA</p> <p>SALUNO.CODPESSOA -> PPESSOA.CODIGO</p> <p>PPESSOA.SEXO</p>
corRacaAluno		<p>Cor / Raça (PCORRACA.DESCRICAO)</p> <p>SMATRICULA.RA -> SALUNO.RA</p> <p>SALUNO.CODPESSOA -> PPESSOA.CODIGO</p> <p>PPESSOA.CORRACA -> PCORRACA.CODCLIENTE</p> <p>PCORRACA.DESCRICAO</p>

<p>tipoDeficiencia</p>	<p>Portador de Necessidades Especiais Tipo:</p> <p>SMATRICULA.RA -> SALUNO.RA</p> <p>SALUNO.RA -> SALUNOMECA.RA</p> <p>SALUNOMECA.SURDEZ (Surdez)</p> <p>SALUNOMECA.DEFICIENTEFALA (Fala)</p> <p>SALUNOMECA.DEFICIENTEAUDITIVO (Auditiva)</p> <p>SALUNOMECA.SURDOCEGUEIRA (Surdocegueira)</p> <p>SALUNOMECA.SUPERDOTADO (Alta habilidade/Superdotação)</p> <p>SALUNOMECA.AUTISMO (Transtorno do Espectro Autista)</p> <p>SALUNOMECA.DEFMULTIPLA (Deficiência múltipla)</p> <p>SALUNOMECA.DEFICIENTEMENTAL (Intelectual)</p> <p>SALUNOMECA.PSICOSSOCIAL (Psicossocial)</p> <p>SALUNOMECA.PSICOSEINFANTIL (Transtorno desintegrativo da infância)</p> <p>SALUNOMECA.RETT (Síndrome de Rett)</p> <p>SALUNOMECA.ASPERGER (Síndrome de Asperger)</p> <p>SALUNOMECA.BAIXAVISAO (Baixa visão)</p> <p>SALUNOMECA.CEGUEIRA (Cegueira)</p> <p>SALUNOMECA.PARAPLEGIA (Paraplegia)</p> <p>SALUNOMECA.TETRAPLEGIA (Tetraplegia)</p> <p>SALUNOMECA.HEMIPLÉGIA (Hemiplegia)</p> <p>SALUNOMECA.PARALCEREBRAL (Paralisia cerebral)</p> <p>SALUNOMECA.AMPUTACAO (Amputação)</p> <p>Será retornado a classificação da Necessidade Especial, ou seja, os valores em parênteses acima.</p> <p>Caso mais de um campo acima seja verdadeiro, será retornado a concatenação das informações.</p> <p>Exemplo: Fala - Visual</p>
<p>emailAluno</p>	<p>Email do aluno (PPESSOA.EMAIL)</p> <p>SMATRICULA.RA -> SALUNO.RA</p>

		SALUNO.CODPESSOA -> PPESSOA.CODIGO PPESSOA.EMAIL
cepAluno		CEP do aluno (PPESSOA.CEP) SMATRICULA.RA -> SALUNO.RA SALUNO.CODPESSOA -> PPESSOA.CODIGO PPESSOA.CEP
celAluno		Telefone celular do aluno (PPESSOA.TELEFONE2) SMATRICULA.RA -> SALUNO.RA SALUNO.CODPESSOA -> PPESSOA.CODIGO PPESSOA.TELEFONE2
cpfAluno		CPF do aluno (PPESSOA.CPF) SMATRICULA.RA -> SALUNO.RA SALUNO.CODPESSOA -> PPESSOA.CODIGO PPESSOA.CPF
rgAluno		Carteira de identidade do aluno (PPESSOA.CARTIDENTIDADE) SMATRICULA.RA -> SALUNO.RA SALUNO.CODPESSOA -> PPESSOA.CODIGO PPESSOA.CARTIDENTIDADE

Se erro na autenticação, o Webservice irá retornar uma mensagem de erro e será finalizado.

Exemplo de mensagem com erro:

```
{
  "sucesso": "0",
  "mensagem": "Erro ao buscar alunos da Turma 'TEC-P-000208/2016' do Professor 'Professor Senai 01': Usuário não autenticado.",
  "alunosTurma": []
}
```

Se autenticado, o Webservice irá realizar a consulta na base de dados do TOTVS Educacional e retornar os alunos matriculados da turma/disciplina passada como parâmetro. O retorno será no formato **JSON**.

Exemplo de mensagem com sucesso:

```
{
  "sucesso": "1",
```

```
"mensagem": "Registros recuperados com sucesso!",
"alunosTurma": [
  {
    "raAluno": "13740",
    "nomeAluno": "Aluno 13740",
    "dtNascAluno": "1996-10-02 00:00:00.000",
    "sexoAluno": "F",
    "corRacaAluno": "Parda",
    "tipoDeficiencia": "",
    "emailAluno": "aluno13740@gmail.com",
    "cepAluno": "85960-123",
    "celAluno": "(45) 9999-8888",
    "cpfAluno": "08884398573",
    "rgAluno": "mg15102821"
  },
  {
    "raAluno": "28105",
    "nomeAluno": "Aluno 28105",
    "dtNascAluno": "1999-07-31 00:00:00.000",
    "sexoAluno": "M",
    "corRacaAluno": "Branca",
    "tipoDeficiencia": "Auditiva",
    "emailAluno": "aluno28105@hotmail.com",
    "cepAluno": "85960-321",
    "celAluno": "(45) 9999-7777",
    "cpfAluno": "09928517936",
    "rgAluno": "mg15102821"
  }
]
}
```

- Protótipo

Não se aplica.

5. Histórico de Atualização do Documento

5.1. Histórico de Versões

Autor	Versão	Data rev.	Descrição das alterações	Requer. aprov.	Seção	Pag.
Maria Fernanda Notini de Castro	1.0	16/03/2016	Documento inicial	Sim		
Davi Leles Braga Pereira Lacerda	2.0	20/04/2016	Inclusão de novas deficiências	Sim		12

5.2. Aprovação

Nome	Data	Assinatura