|  |  |
| --- | --- |
| DSE – Departamento de Sistemas Educacionais | |
| Logo AccSys Abr15 | Logo MIT Abr15 |

**ESPECIFICAÇÃO TECNICA**

**Projecto ISUTC Fénix -Primavera**

**Versão 0.1**

Índice

[Operação 1 – Actualiza Cliente 2](#_Toc427162544)

[Descrição 2](#_Toc427162545)

[Funcionalidade do ActualizaCliente 2](#_Toc427162546)

[ActualizaCliente 3](#_Toc427162547)

[Classe Contacto 4](#_Toc427162548)

[Operação 2 – GravarFactura 5](#_Toc427162549)

[Descrição 5](#_Toc427162550)

[Funcionalidade do Método GravarFactura 5](#_Toc427162551)

[Gravar Factura 6](#_Toc427162552)

[Classe Artigo 6](#_Toc427162553)

[Operação 3 – ActualizaContrato 7](#_Toc427162554)

[Descrição 7](#_Toc427162555)

[Funcionalidade do Método ActualizaContrato 7](#_Toc427162556)

[Gravar Contracto 8](#_Toc427162557)

[Classe Artigo 8](#_Toc427162558)

[Operação 4 – ConsultaConta 9](#_Toc427162559)

[Descrição 9](#_Toc427162560)

[Funcionalidade do Método ConsultaConta 9](#_Toc427162561)

[Gravar Contracto 9](#_Toc427162562)

[Classe Pendentes 9](#_Toc427162563)

# Operação 1 – Actualiza Cliente

## Descrição

Esta operação será responsável por criar ou editar um determinado cliente (candidato, estudante ou instituto) através do método **actualizarCliente (<*inputs listados*>)**.

O Fénix será responsável por gerir a lógica do uso durante as operações de inscrição, matricula e actualização a nível dos campos obrigatórios. (candidatura, matricula).

### Funcionalidade do ActualizaCliente

1. **Cria ou Actualiza o cliente no Erp Primavera (public PrimaveraResultStruture actualizaCliente(<inputs listados>))**

Se o código do cliente não existir no erp primavera irá criar o cliente, obedecendo as regras de negócio do Primavera por defeito.

Se o código do cliente existir irá actualizar todos os dados do cliente.

**Dados Por Defeito**

***Moradas***

**Tipo Terceiro** este campo esta mapeado com o código da licenciatura

***Dados Fiscais***

**Nome Fiscal** é por defeito igual ao Nome

**Espaço Fiscal** é por defeito é Nacional

**Regime do IVA** por defeito é Normal

**Pessoa** por defeito é Singular

***Dados Comercias***

**Tipo Preço** por defeito é P.V.P. 1

**Desconto** 0,00%

**Condições de Pagamento** será actualizada consoante o Vendedor (ISU – 10, Próximo dia 10 ; ITC – Próximo dia 5, EMU – Próximo dia 10, Outros – Factura 30 dias)

**Modo Pagamento** é por defeito é REP24 – Recebimento Ponto 24

**Moeda** por defeito é MT - Metical

1. **Adicionar os respectivos contactos ao cliente (private PrimaveraResultStruture insert\_update\_Client\_Contacts(List<Contacts> lista\_contactos) )**
2. **Criar a respectiva referencia bancaria (private PrimaveraResultStruture criar\_Referencia\_Bancaria(string codigoCliente))**

No caso do campo nrNovoCliente ou o cliente ser novo será invocado este método que irá criar a referência bancaria do respectivo cliente e actualiza a referência em todos os documentos financeiros pendentes de pagamento (FA) a respectiva referencia.

1. **Fim do Actualiza Cliente**

Devolve um resultado uma classe do tipo ***PrimaveraResultStruture*** nos seguintes cenários

- Erro, o respectivo erro

- Criação do cliente ou actualização do código do cliente, o código da referencia

- Actualização de dados, sucesso completo

### ActualizaCliente

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| tipoPlataforma | Tipo Plataforma Instalada no Servidor | Integer | 0- Executive  **1- Profissional** |
| codEmpresa | Código da Empresa Primavera | String(10) |  |
| codUtilizador | Código do Utilizador Primavera | String (10) |  |
| password | Password do Utilizador Primavera | String (10) |  |
| cliente (\*) | Código do Cliente | String(12) | Alfanuméricos |
| novoNrCliente | Novo código do cliente | String(12) | Alfanuméricos |
| nome (\*) | Nome completo do cliente | String (50) | no numbers |
| nomeFiscal | Nome fiscal do cliente | String (50) | no numbers |
| Desconto | Desconto do cliente | Double | Positive Numbers |
| Fac\_Morada (\*) | Morada do cliente na Factura | String (50) |  |
| Fac\_Localidade | Localidade do cliente na Factura | String (20) |  |
| Fac\_NrTelefone | Número do Telefone do cliente na Factura | String (20) | (82) 1234567 |
| Nuit | NUIT do cliente | String (20) |  |
| Vendedor (\*) | Instituto do cliente | String(3) | ISU |
| Observação | Observações sobre o cliente | String (250) |  |
| CDU\_Bolsa | Se o cliente tem bolsa | Boolean |  |
| CDU\_GeraMulta | Se o cliente vai gerar multas | Boolean |  |
| CDU\_TipoIngresso | O tipo de Ingresso do cliente | String (100) |  |
| CDU\_CodLic | O código da licenciatura do cliente | String(50) |  |
| CDU\_Turma | O código da turma do cliente | String(50) |  |
| lista\_Contactos |  | List<Contacto> |  |

### Classe Contacto

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| Codigo | Código do cliente | String(12) | Alfanuméricos |
| primeiroNome | Primerio nome do cliente | String(12) |  |
| ultimoNome | Ultimo nome do cliente | String (12) |  |
| nomeCompleto | Nome completo do cliente | String(12) |  |
| tipoContacto | Tipo de Contacto do cliente | String(12) | Aluno |
| emailPrincipal | Email ISUTC do cliente | String(100) |  |
| emailAlternativo | Email Alternativo do cliente | String(100) |  |
| nrTelefone | Número de telefone do cliente | String(20) |  |
| nrTelefoneAlternativo | Número de telefone Alternativo do cliente | String(20) |  |
| Morada | Morada do cliente | String(50) |  |
| Localidade | Localidade do cliente | String(20) |  |
|  |  |  |  |

**Legenda:**

* **SA –** Secretaria Académica;
* **(\*) –** Campos obrigatórios
* **Cliente –** Estudante, Candidato a Estudante e Instituições provedoras das bolsas

# Operação 2 – GravarFactura

## Descrição

Esta operação será responsável por gravar facturas para um determinado cliente (candidato, estudante ou instituto) através do método **gravarFactura (<*inputs listados*>)**.

O Fénix será responsável por gerir a lógica do uso durante as operações na inscrição, matricula nível dos artigos e as suas respectivas quantidades sem os preços que serão definidos no ERP para cada tipo de terceiro(curso).

### Funcionalidade do Método GravarFactura

1. **Cria o Cabeçalho do Documento de Venda -** public PrimaveraResultStructure PreencheCabecalho\_DocumentoVenda( DateTime dataDoc, DateTime dataGravacao, string tipoEntidade, string codigoEntidade, string tipoDoc, ref string serie, ref string CDU\_semestre, ref string CDU\_RefBancaria)

Preenche o cabeçalho do documento factura, o webService fazer o pedido para este método com o seguintes valores:

* **dataDoc** será a data do sistema
* **dataGravacao** será a data do sistema
* **tipoEntindade** será igual á “C”
* **tipoDoc** para o caso da Factura – **FA**
* **série** será calculada no ERP a série por defeito para o respectivo documento de venda
* **CDU\_Semestre** será vazio
* **CDU\_RefBancaria** este campo irá ser preenchido pela referência do estudante

O resto dos campos necessários para preencher ao documento de venda factura será invocada partir do método **preencheDadosRelacionados()**

Do motor ERP que irá preencher os dados fiscais e comerciais da entidade.

1. **Adiciona Linha ao Documento de Venda (**public PrimaveraResultStructure Adiciona\_Linhas\_DocumentoVenda(string artigo,double quantidade, double desconto )**)**

Após a criação do documento de venda o método irá fazer um loop sobre a lista de artigos recebido no método geraFactura, e adicionar-se a linha do documento de venda.

O motor ERP irá preencher automaticamente o valor do preço do artigo definido no inicio do ano académico pelo o utilizador no erp.

1. **Grava o Documento de Venda (**public PrimaveraResultStructure Gravar\_DocumentoVenda()**)**

O motor Erp irá actualizar no Erp Primavera o documento de venda a partir do método .Actualiza(documentoVenda, str\_avisos).

1. **Fim do GravarFactura**

Devolve um resultado uma classe do tipo ***PrimaveraResultStruture*** nos seguintes cenários

- Erro, o respectivo erro

- Sucesso Completo, string.Format("{0} {1}/{2}", documentoVenda.get\_Tipodoc(), documentoVenda.get\_NumDoc(), documentoVenda.get\_Serie())

- Sucesso Com Avisos, string.Format("{0} {1}/{2}", documentoVenda.get\_Tipodoc(), documentoVenda.get\_NumDoc(), documentoVenda.get\_Serie()) + str\_avisos

### Gravar Factura

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| tipoPlataforma | Tipo Plataforma Instalada no Servidor | Integer | 0- Executive  **1- Profissional** |
| codEmpresa | Código da Empresa Primavera | String(10) |  |
| codUtilizador | Código do Utilizador Primavera | String (10) |  |
| Password | Password do Utilizador Primavera | String (10) |  |
| nomeUtilizador | Nome do Utilizador Fénix | String (100) | Alfanuméricos |
| cliente (\*) | Código do Cliente | String(12) | Alfanuméricos |
| lista\_Artigos | Lista de Artigos a Serem Facturados | List<Artigo> |  |
| tipoDoc (\*) | Tipo de documento (FA) | String (5) |  |

### Classe Artigo

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| Artigo (\*) | Código do Artigo | String(12) | Alfanuméricos |
| Quantidade | Quantidade dos artigos | Double |  |
| Desconto | Desconto sobre o artigo | Double |  |

**Legenda:**

* **SA –** Secretaria Académica;
* **(\*) –** Campos obrigatórios
* **Cliente –** Estudante, Candidato a Estudante e Instituições provedoras das bolsas

# Operação 3 – ActualizaContrato

## Descrição

Esta operação será responsável por gravar contratos para um determinado cliente (estudante) através do método **actualizaContrato (<*inputs listados*>)**.

O Fénix será responsável por gerir a lógica do uso durante as operações na matrícula e renovação nível dos artigos e as suas respectivas quantidades sem os preços que serão definidos no ERP para cada tipo de terceiro (curso).

### Funcionalidade do Método ActualizaContrato

1. **Cria o Cabeçalho do Documento de Venda -** public PrimaveraResultStructure PreencheCabecalho\_DocumentoVenda( DateTime dataDoc, DateTime dataGravacao, string tipoEntidade, string codigoEntidade, string tipoDoc, ref string serie, ref string CDU\_semestre, ref string CDU\_RefBancaria)

Preenche o cabeçalho do documento factura, o webService fazer o pedido para este método com o seguintes valores:

* **dataDoc** será a data do sistema
* **dataGravacao** será a data do sistema
* **tipoEntindade** será igual á “C”
* **tipoDoc** para o caso da Factura – **CTR**
* **série** será calculada no ERP a série por defeito para o respectivo documento de venda
* **CDU\_Semestre** será vazio
* **CDU\_RefBancaria** este campo irá ser preenchido pela referência do estudante

O resto dos campos necessários para preencher ao documento de venda factura será invocada partir do método **preencheDadosRelacionados()**

Do motor ERP que irá preencher os dados fiscais e comerciais da entidade.

1. **Adiciona Linha ao Documento de Venda (**public PrimaveraResultStructure Adiciona\_Linhas\_DocumentoVenda(string artigo,double quantidade, double desconto )**)**

Após a criação do documento de venda o método irá fazer um loop sobre a lista de artigos recebido no método geraFactura, e adicionar-se a linha do documento de venda.

O motor ERP irá preencher automaticamente o valor do preço do artigo definido no inicio do ano académico pelo o utilizador no erp.

1. **Grava o Documento de Venda (**public PrimaveraResultStructure Gravar\_DocumentoVenda()**)**

O motor Erp irá actualizar no Erp Primavera o documento de venda a partir do método .Actualiza(documentoVenda, str\_avisos).

Devolve um resultado uma classe do tipo ***PrimaveraResultStruture*** nos seguintes cenários

- Erro, o respectivo erro

- Sucesso Completo, string.Format("{0} {1}/{2}", documentoVenda.get\_Tipodoc(), documentoVenda.get\_NumDoc(), documentoVenda.get\_Serie())

- Sucesso Com Avisos, string.Format("{0} {1}/{2}", documentoVenda.get\_Tipodoc(), documentoVenda.get\_NumDoc(), documentoVenda.get\_Serie()) + str\_avisos

1. **Grava a Avença com base no Documento de Venda (**public PrimaveraResultStructure Grava\_DocumentoAvenca() **)**

Este método irá gravar avença com base no documento de contrato e na informação do semestre para o início e fim da avença.

### Gravar Contracto

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| tipoPlataforma | Tipo Plataforma Instalada no Servidor | Integer | 0- Executive  **1- Profissional** |
| codEmpresa | Código da Empresa Primavera | String(10) |  |
| codUtilizador | Código do Utilizador Primavera | String (10) |  |
| Password | Password do Utilizador Primavera | String (10) |  |
| nomeUtilizador | Nome do Utilizador Fénix | String (100) | Alfanuméricos |
| cliente (\*) | Código do Cliente | String(12) | Alfanuméricos |
| lista\_Artigos | Lista de Artigos a Serem Facturados | List <Artigo> |  |
| tipoDoc | Tipo de documento (CTR) | String (5) |  |

### Classe Artigo

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| Artigo (\*) | Código do Artigo | String(12) | Alfanuméricos |
| Quantidade | Quantidade dos artigos | Double |  |
| Desconto | Desconto sobre o artigo | Double |  |

**Legenda:**

* **SA –** Secretaria Académica;
* **(\*) –** Campos obrigatórios
* **Cliente –** Estudante, Candidato a Estudante e Instituições provedoras das bolsas

# Operação 4 – ConsultaConta

## Descrição

Esta operação será responsável por consultar os pendentes do cliente que se encontram fora do prazo. public List <Pendentes> **consultaConta (string codigo)**.

### Funcionalidade do Método ConsultaConta

1. **Da lista de Pendentes -** public List <Pendentes> daListaPendentes ( string tipoEntidade, string codigoEntidade)

Devolve a lista de pendentes do cliente que já se encontram fora do prazo de validade.

Esses pendentes podem ser Facturas ou Adiantamentos.

### Gravar Contracto

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| tipoPlataforma | Tipo Plataforma Instalada no Servidor | Integer | 0- Executive  **1- Profissional** |
| codEmpresa | Código da Empresa Primavera | String(10) |  |
| codUtilizador | Código do Utilizador Primavera | String (10) |  |
| Password | Password do Utilizador Primavera | String (10) |  |
| nomeUtilizador | Nome do Utilizador Fénix | String (100) | Alfanuméricos |
| cliente (\*) | Código do Cliente | String(12) | Alfanuméricos |

### Classe Pendentes

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| cliente | Código do cliente | String(12) | Alfanuméricos |
| tipoDocumento | Tipo de Documento | String(5) | FA, ADC |
| numDoc | Numero do Documento | String(50) |  |
| série | Serie do Documento | String(10) |  |
| valorTotal | Valor Total do Documento | Double |  |
| valorPendente | Valor Pendente do Documento | Double |  |
| moeda | Moeda | String(10) |  |
| dataCriação | Data de Criação do Documento | DateTime |  |
| dataVencimento | Data de Vencimento do Documento | DateTime |  |

**Legenda:**

* **SA –** Secretaria Académica;
* **(\*) –** Campos obrigatórios
* **Cliente –** Estudante, Candidato a Estudante e Instituições provedoras das bolsas