# Operação 1 – Actualizar Cliente

## Descrição

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

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

### Funcionalidades do método *actualizarCliente(<inputsListados>)*

1. **Cria ou Actualiza o cliente no ERP Primavera *(publicPrimaveraResultStrutureactualizaCliente(<inputsListados>))***

Se o código do cliente não existir no ERP primavera, será criado o cliente, obedecendo as regras de negócio do Primavera por defeito.

Se o código do cliente existir, serão actualizados todos os dados do cliente.

**Dados Por Defeito**

* ***Moradas***
  + **Tipo Terceiro -** este campo esta mapeado com o código da licenciatura
* ***Dados Fiscais***
  + **Nome Fiscal -** por defeito é igual ao Nome
  + **Espaço Fiscal** - por defeito é Nacional
  + **Regime do IVA -** por édefeito Normal
  + **Pessoa –** por defeito é Singular
* ***Dados Comercias***
  + **Tipo Preço -** por defeito é P.V.P. 1
  + **Desconto -** 0,00%
  + **Condição de Pagamento -** será actualizada consoante o Vendedor (ISU – Próximo dia 10; ITC – Próximo dia 5, EMU – Próximo dia 10, Outros – Factura 30 dias)
  + **Modo Pagamento -** por defeito é REP24 – Recebimento Ponto 24
  + **Moeda -** por defeito é MT - Metical

1. **Adicionar os respectivos contactos ao cliente *(privatePrimaveraResultStrutureinsert\_update\_Client\_Contacts(List<Contacts>lista\_contactos))***
2. **Criar a respectiva referência bancária *(privatePrimaveraResultStruturecriar\_Referencia\_Bancaria(stringcodigoCliente))***

No caso de o cliente ser novo (campo *“nrNovoCliente”*), será invocado este método que irá criar a referência bancária deste, e actualizar a referência em todos os documentos financeiros pendentes do pagamento (FA) a respectiva referência.

1. **Fim do método *Actualizar Cliente***

Devolve como resultado uma classe do tipo ***PrimaveraResultStruture()*** nos seguintes cenários:

* **Erro** – que representa o respectivo erro;
* **Criação do cliente ou actualização do código do cliente** – que cria ou actualiza o código de referência;
* **Actualização de dados** – que representa sucesso completo.

### actualizarCliente(<inputs(campos listados abaixo)>)

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| tipoPlataforma | Tipo de 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 |
| facMorada (\*) | Morada do cliente na Factura | String(50) |  |
| facLocalidade | Localidade do cliente na Factura | String(20) |  |
| facNrTelefone | Número do Telefone do cliente na Factura | String(20) | (82) 1234567 |
| nuit | NUIT do cliente | String(20) |  |
| vendedor (\*) | Instituto do cliente | String(3) | ISU |
| observacao | 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) |  |
| listaContactos | Contactos do cliente | List<Contacto> |  |

### Classe Contacto

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do campo | Descrição | Tipo de Dados/ Tamanho | Valores Permitidos |
| codigo | Código do cliente | String(12) | Alfanuméricos |
| primeiroNome | Primeiro nome do cliente | String(12) |  |
| ultimoNome | Último 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 – Gravar Factura

## Descrição

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

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

### Funcionalidades do Método *gravarFactura(<inputsListados>)*

1. **Cria o Cabeçalho do Documento de Venda –**

***publicPrimaveraResultStructurePreencheCabecalho\_DocumentoVenda(DateTimedataDoc, DateTimedataGravacao, stringtipoEntidade, stringcodigoEntidade, stringtipoDoc,refstringserie, refstringCDU\_semestre, refstringCDU\_RefBancaria)***

Preenche o cabeçalho do documento factura. O *webService* faz o pedido para este método com os seguintes valores:

* **dataDoc** será igual a data do sistema
* **dataGravacao** será igual a data do sistema
* **tipoEntindade** será igual á “C”
* **tipoDoc** para o caso da Factura – **FA**
* **serie** 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**

***(publicPrimaveraResultStructureAdiciona\_Linhas\_DocumentoVenda(stringartigo,double quantidade,doubledesconto))***

Após a criação do documento de venda, o método irá fazer um *loop* sobre a lista de artigos recebidos no método *geraFactura()*, e adicionar-se-á 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 utilizador no ERP.

1. **Grava o Documento de Venda *(publicPrimaveraResultStructureGravar\_DocumentoVenda())***

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

1. **Fim do método *GravarFactura***

Devolve como resultado uma classe do tipo ***PrimaveraResultStruture*** nos seguintes cenários:

* **Erro** – que representa 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 de 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 |
| listaArtigos | Lista de Artigos a Serem Facturados | List<Artigo> |  |
| tipoDoc(\*) | Tipo de documento (FA) | String (5) |  |

### *ClasseArtigo*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 – Actualizar Contrato

## Descrição

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

O Fénix será responsável por gerir a lógica do uso durante as operações de matrícula, 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(<inputsListados>)*

1. **Cria o Cabeçalho do Documento de Venda**

***(publicPrimaveraResultStructurePreencheCabecalho\_DocumentoVenda(DateTimedataDoc, DateTimedataGravacao, stringtipoEntidade, stringcodigoEntidade, stringtipoDoc,refstringserie, refstringCDU\_semestre, refstringCDU\_RefBancaria))***

Preenche o cabeçalho do documento factura. O webService faz o pedido para este método com os 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**
* **serie** 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 a partir do método ***preencheDadosRelacionados()*** do motor ERP que irá preencher os dados fiscais e comerciais da entidade.

1. **Adiciona Linha ao Documento de Venda**

***(publicPrimaveraResultStructureAdiciona\_Linhas\_DocumentoVenda(stringartigo,double quantidade,doubledesconto))***

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

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

1. **Grava o Documento de Venda *(publicPrimaveraResultStructureGravar\_DocumentoVenda())***

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

1. **Fim do método *actualizaContrato(<inputsListados>)***

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

* **Erro** – que representa 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 *(publicPrimaveraResultStructureGrava\_DocumentoAvenca())***

Este método irá gravar a 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 de 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) |  |

### *ClasseArtigo*

|  |  |  |  |
| --- | --- | --- | --- |
| 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 – Consultar Conta

## Descrição

Esta operação será responsável por consultar os pendentes do cliente que se encontram fora do prazo. *publicList<Pendentes>****consultaConta(stringcodigo)****.*

### Funcionalidade do Método *consultaConta(stringcodigo)*

1. **Devolver a lista de Pendentes *publicList<Pendentes>daListaPendentes(stringtipoEntidade, stringcodigoEntidade)***

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 de 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 |

### *ClassePendente*

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Número do Documento | String(50) |  |
| serie | Série do Documento | String(10) |  |
| valorTotal | Valor Total do Documento | Double |  |
| valorPendente | Valor Pendente do Documento | Double |  |
| moeda | Moeda | String(10) |  |
| dataCriacao | 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