UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA



|  |
| --- |
| Desenvolvimento de uma Aplicação em Orientação a Objetos |

**Documento Provisório**

|  |
| --- |
| Pedro Miguel Ferreira Tavares Carrega |

Mestrado em Engenharia Informática

Designação da Especialização / Perfil, se aplicável

|  |
| --- |
| Dissertação orientada por:/Trabalho de Projeto orientado por:  Nome dos orientadores |

2022

**Agradecimentos**

Lorem ipsum dolor sit amet, …

*Dedicatória.*

**Resumo**

O ser humano é uma espécie sempre em desenvolvimento. Para acompanhar esse mesmo desenvolvimento existe uma constante evolução tecnológica, que se aplica a todas as facetas da nossa sociedade, quer seja social, política ou económica. Contudo esta evolução tem uma potencial consequência: o aumento do número e da complexidade de recursos que podem ser tratados. A necessidade de gestão de todos estes recursos levou à criação de aplicações de Enterprise Resouce Planning (ERP), sendo este tipo de aplicação a desenvolvida na empresa ARTSOFT. ERP ARTSOFT é uma aplicação de gestão empresarial que é estruturada em vários módulos, como Gestão Comercial, Contabilidade, Gestão Financeira e Recursos Humanos, permitindo adaptar a solução às necessidades do utilizador.

Este projeto de tese revolve à volta do módulo de Recursos Humanos, em particular o desenvolvimento de uma interface que integre os *web services* disponibilizados pela Segurança Social, para comunicar o vínculo ou cessar de contrato de um trabalhador. Esta interface irá permitir o utilizador pré-visualizar e editar os dados a serem comunicados pelo *web service*, propagando possíveis alterações efetuadas para a base de dados.

**Palavras-chave:** ERP ARTSOFT, OOP, Serviços Web, Segurança Social.

**Abstract**

The human species is in a state of permanent development. To match this development it exists a constant technological development to all facets of our society, whether it is social, political or economic. However this evolution comes with a potential cost: the increase in the number and complexity of the resources to be processed. That need of managing all those resources led to the creation of Enterprise Resource Planning (ERP) applications, being this the type of solution developed at the company ARTSOFT. ERP ARTSOFT is integrated management business application that is divided in multiple modules, like Commercial Management, Accounting, Financial Management and Human Resources, allowing the user to create a solution according to its needs.

This thesis project will revolve around the Human Resources modules, in particular the development of an interface that integrates web services made available by Segurança Social, to communicate the contract agreement or end of the contract agreement of a worker. This interface will allow the user preview and edit the contract info to be communicated to the web service, saving possible changes made in the database.

**Keywords:** ERP ARTSOFT, OOP, Segurança Social, Web Services

**Conteúdo**

Capítulo 1 Introdução 1

1.1 Objetivos 1

1.2 Estrutura do documento 2

Capítulo 2 Trabalho Relacionado 4

Capítulo 3 Especificação de Requisitos 6

3.1 Estrutura da Especificação de Requisitos 6

3.2 Web Service 7

3.3 Comunicar Vínculo de Contrato do Trabalhador 8

3.3.1 Documentação 8

3.3.2 Redação 9

3.4 Cessar Vínculo do Trabalhador 11

3.4.1 Documentação 11

3.4.2 Redação 11

Capítulo 4 Implementação 15

**Lista de Figuras**

**Lista de Tabelas**

# Introdução

O desenvolvimento tecnológico é o responsável por grande parte do avanço científico observável na nossa sociedade, sendo que este desenvolvimento é visível em todas as facetas da sociedade, quer seja social, político ou económico. De um ponto de vista empresarial este desenvolvimento trouxe várias vantagens oferecendo às empresas mais dados e informação sobre o mercado e o produto em que a mesma atua permitindo uma gestão e análise mais detalhada e precisa do seu negócio. Porém, estas vantagens trazem possíveis consequências às empresas: um aumento da quantidade e complexidade dos recursos a processar; levando a uma gestão e análise mais lenta e menos rigorosa do seu negócio. Isto levou à criação de aplicações *Enterprise Resource Planning* (ERP), um *software* que fosse capaz de gerir atividades financeiras de uma empresa, as cadeias de fornecimentos, comércio, recursos humanos e outros.

ARTSOFT é uma empresa de *software*, com mais de 30 anos de experiência na área, que desenvolve soluções de gestão empresariais. A empresa produz atualmente somente um produto ERP ARTSOFT, uma aplicação de gestão constituída por vários módulos permitindo a criação de uma solução que melhor se adapte às necessidades do cliente. Estes módulos cobrem todas as necessidades que uma empresa possa ter como por exemplo análise contabilística ou gestão de recursos humanos.

No âmbito da disciplina de Projeto do Mestrado de Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa, a empresa ARTSOFT propôs um projeto, que se irá focar no módulo de Recursos Humanos, cujo propósito é a integração de vários serviços criados e disponibilizados pela Segurança Social.

## Motivação

Sempre que uma empresa efetua qualquer ação que envolve um contrato entre empresa e trabalhador, por exemplo o assinar ou terminar de um contrato com um trabalhador, essa ação tem de ser reportada à Segurança Social. Este processo de momento pode ser efetuado de duas diferentes maneiras: a empresa pode utilizar o seu acesso na plataforma *online* da Segurança Social e aí comunicar essa mudança ou pode se deslocar diretamente a uma delegação da Segurança Social; para uma empresa este processo pode ser bastante ineficiente.

Um grande número de empresas utilizam aplicações ERP para suportar o seu negócio, sendo que grande parte dessas aplicações disponibilizam funcionalidades de base de dados que guardem toda a informação relevante referente aos seus trabalhadores. Desta maneira, uma empresa que efetue uma nova contratação tem de criar um registo no ERP e depois preencher o registo de contrato na plataforma da Segurança Social com esses mesmos dados, o que, num contexto empresarial, causa um gasto de tempo desnecessariamente dispendioso.

Para suportar as empresas a Segurança Social anunciou dois *web services*: o Consultar Trabalhadores e o Registar Vínculo Trabalhador; ambos estes serviços encontram-se integrados na aplicação ERP ARTSOFT. No fim do ano 2021 a Segurança Social anunciou a criação de diversos novos *web services* com o objetivo de suportar ainda mais as empresas. Em particular anunciou os dois serviços a serem integrados, na aplicação ERP ARTSOFT, no âmbito deste projeto: o Comunicar do Vínculo do Contrato de Trabalhador, o sucessor do serviço Registar Vínculo Trabalhador que irá ser descontinuado, e o Cessar do Vínculo de Trabalhador. Todos estes serviços planeiam-se integrar gradualmente na aplicação ERP ARTSOFT de forma a conseguir responder a todas as possíveis necessidades do cliente. Objetivos

O objetivo deste projeto de tese será a integração de dois diferentes *web services* disponibilizados pela Segurança Social: o comunicar de vínculo de trabalhador e o cessar do vínculo de trabalhador. Em ordem … a enumeram-se os seguintes objetivos de forma a obter uma solução de qualidade:

1. Estudo da aplicação ERP ARTSOFT e dos *web services* a integrar: A aplicação ERP ARTSOFT é composta por vários módulos contudo a linguagem utilizada nesta aplicação é C++, uma linguagem orientada a objetos, ou seja existem módulos dependentes de outros. Dada esta possível partilha de serviços e objetos, este projeto não pode de alguma maneira comprometer o funcionamento de qualquer uma funcionalidade já existente na aplicação ERP ARTSOFT.
2. Identificação dos requisitos funcionais: A partir do estudo realizado à aplicação ERP ARTSOFT e dos web services a serem integrados, vão ser definidos os requisitos funcionais a serem implementados tendo em conta os objetivos do desenvolvimento.
3. Redação de especificações de requisitos: Tendo em conta os resultados do objetivo anterior, será redigida uma especificação de requisitos para cada *web service* a ser integrado na aplicação.
4. Implementação da solução: Nesta fase do projeto serão implementadas as funcionalidades identificadas e definidas na especificação de requisitos; com consideração para a arquitetura da aplicação e do módulo afeto a este desenvolvimento.
5. Realização de testes: Uma vez implementada a solução serão efetuados uma bateria de testes funcionais para garantir a qualidade da mesma e que todos os requisitos funcionais identificados são respeitados. De notar que embora o programador responsável efetue testes iniciais, os testes mencionados são realizados pela equipa de testes; por isso esta parte do desenvolvimento não vai ser analisado neste relatório.

## Estrutura do documento

Este relatório vai ser composto por seis capítulos. Dada a introdução deste documento, aonde é realizada uma contextualização do problema e apresentados os objetivos do projeto, irão ser apresentados no capítulo 2 alguns conceitos básicos necessários para auxiliar a compreensão dos temas e conceitos discutidos neste relatório. No capítulo 3 serão apresentadas em detalhe as especificações de requisitos elaboradas para este projeto. De seguida no capítulo 4 será abordada a concretização da implementação da solução do projeto, com uma apresentação dos resultados do desenvolvimento no capítulo 5. Concluindo no capítulo 5 com as principais conclusões e possíveis formas de melhorar o processo de desenvolvimento normalmente realizado na empresa ARTSOFT.

# Trabalho Relacionado

TODO

*Web services* é um tema já muito explorado na área de engenharia de *software* devido à sua constante evolução, existindo vários artigos que propõem e exploram diferentes metodologias e técnicas de implementação.

O artigo de Lee *et al.* defende que as atuais metodologias utilizadas para o desenvolvimento de aplicações que integram *web services* no seu funcionamento não se revelam suficientes. As principais dificuldades destacadas são:

\begin{itemize}

\item A dificuldade em determinar todos os requisitos da aplicação dado que os requisitos não provêm de uma só fonte e

\item A implementação do consumo e comunicação do servi-ço pois diferentes sistemas utilizam diferentes \textit{interfaces} e métodos de interação.

\end{itemize}

A solução proposta é a integração na metodologia \textit{Agile} das melhores práticas no desenvolvimento de \textit{web services}. Contudo, os autores não apresentam resultados que suportem a eficácia da sua solução.

*Middlewares* foram criados com o propósito de permitir a comunicação entre diferentes sistemas independentemente da implementação ou \textit{hardware} que utilizem. Contudo, com o aumento da complexidade de \textit{middlewares}, o problema pque levou à sua criação voltou a surgir, criando uma necessidade de \textit{"middleware for middleware"}\cite{vinoskiMiddleware}. *Web services* tentam resolver este problema mas, ao focarem as implementações no uso de SOAP API, os programadores destas soluções estão a ficar aquém do potencial de *web services*. SOAP API não segue um padrão universal, ou seja, diferentes aplicações têm diferentes implementações da mesma. Este é o problema apresentado no artigo de Vinoski, S.\cite{vinoskiIntegration}, que expõe e explica em detalhe um dos problemas que \textit{web services} enfrentam. O autor desenvolve o tema apresentando várias soluções, em particular, uma solução em desenvolvimento por uma equipa de engenheiros da \textit{Sun Microsystems} com o nome \textit{Web Services Invocation Framework} (WSIF). Esta solução abstrai o protocolo de comunicação, permitindo aplicações seguirem só um padrão de comunicação, independente do protocolo utilizado pelo *web service*. Segundo o autor, esta solução resolve vários dos problemas atuais de *web services* mas é longe de ser perfeita sendo exclusiva a soluções que utilizem Java.

# Especificação de Requisitos

Neste capítulo vamos focar-nos em responder aos objetivos 2 e 3: determinar os requisitos funcionais e a redação da especificação de requisitos. Para isso vamos dividir este capítulo em quatro diferentes secções: primeiro vai ser apresentada e contextualizada a estrutura da especificação de requisitos; concluindo com o estudo da documentação relevante e redação da especificação de requisitos para cada desenvolvimento associado a este projeto: a comunicação do vínculo de contrato do trabalhador e o cessar do contrato de trabalhador.

## Estrutura da Especificação de Requisitos

A nossa especificação de requisitos vai seguir o seguinte formato:

1. Introdução
2. Sumário
3. Requisitos
4. Informação Adicional
5. Documentos de referência e glossário
6. Testes

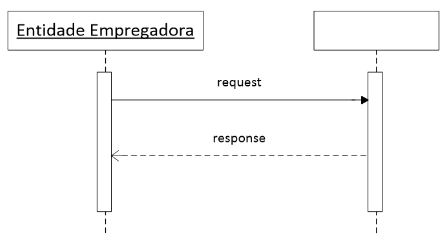
Na introdução são apresentadas as necessidades do cliente de forma a contextualizar o problema, são indicados os módulos que vão ser afetados pelo desenvolvimento e por fim é apresentado o coordenador e programador do projeto, os membros da equipa de teste, o custo estimado e uma estimativa do tempo que irá demorar a desenvolver a solução e a sua margem de erro. No sumário são resumidos em termos técnicos as necessidades do desenvolvimento e as funcionalidades chave a implementar. Na secção dos requisitos são descritos em detalhe, individualmente, os requisitos do desenvolvimento. A estes requisitos é também atribuído um grau de importância - por exemplo, se um requisito criar a necessidade de implementação de uma nova interface, isso é explicado em detalhe sendo apresentado um esboço da interface a ser desenvolvida e uma descrição das funcionalidades de todos os botões e campos interativos. No contexto da aplicação ERP ARTSOFT, esta secção contém mais um campo referente a possíveis esboços de relatórios novos a integrar na aplicação. No segmento da informação adicional, como o nome indica, são indicados quaisquer pressupostos e informação adicional que seja relevante para o desenvolvimento. De seguida são apresentados todos os documentos de referência e o glossário, concluindo o documento com a lista de testes aos quais o desenvolvimento vai ser submetido.

## Web Service

Ambos os serviços a serem integrados no âmbito deste projeto são bastante semelhantes, sendo iguais no método de comunicação entre o cliente e servidor; por isso vamos primeiro fazer uma análise dos pontos comuns dos *web services* transacionado depois para uma análise individual de cada serviço.

Os serviços utilizam *Hypertext Transfer Protocol Secure* (HTTPS) para comunicação entre a cliente e o servidor, utilizando o formato SOAP XML para os seus pedidos sendo este um formato bastante comum dada a sua versatilidade. Para autenticar o utilizador é utilizado um cabeçalho *HTTP Basic Auth*, aonde é concatenado o nome do utilizador com a *password* codificada em Base64; uma vez autenticado é possível efetuar chamadas ao serviço. O corpo do pedido tem toda a informação relevante comunicar ao serviço. Depois de submetido o pedido, a aplicação pode receber quatro diferentes respostas, cada uma com a sua própria assinatura *WSDL*; o valor ‘0’ que indica que ocorreu um erro interno no servidor; o valor ‘1’ indicando que a operação foi realizada com sucesso; o valor ‘2’ representa que houve falta de parâmetros obrigatórios; a falha de validação dos dados enviados é indicado pelo valor ‘3’; ambos os resultado ‘2’ e ‘3’ vêm acompanhados por uma mensagem de erro descritiva para indicar ao utilizador a causa do erro. Contudo, de notar que, só é indicado um erro de cada vez logo para uma experiencia de utilizador mais suave e eficiente, o ideal será a aplicação validar os campos enviados antes de efetuar o pedido à Segurança Social.

O fluxo dos serviços é bastante simples sendo que a comunicação resume-se somente a um só pedido e uma vez processado no servidor, o mesmo devolve uma resposta completando o serviço; esta descrição de fluxo encontra-se representado na imagem abaixo:



## Comunicar Vínculo de Contrato do Trabalhador

### Documentação

O primeiro serviço a ser analisado é o de comunicar o vínculo de contrato do trabalhador à Segurança Social. O corpo do pedido contém toda a informação a ser comunicada, sendo necessária uma análise detalhada do mesmo:

* NISS do trabalhador
* Data de nascimento
* Data de início de contrato
* Data de fim de contrato
* Prestação de contrato
* Código de profissão
* Local de trabalho
* Modalidade de contrato
* Motivo de contrato
* Remuneração base mensal ilíquida
* Diuturnidades
* Número de horas de trabalho
* Número de dias de trabalho
* Percentagem de trabalho
* NISS do trabalhador a substituir

Destes elementos vários são obrigatórios em todos os casos de uso: NISS do trabalhador, data de nascimento, data de início de contrato, código de profissão, local de trabalho, modalidade de contrato e remuneração base. Em alguns casos de uso alguns dos outros elementos passam a ser de comunicação obrigatória, isto depende dos valores comunicados em certos elementos. Em adição é necessário ter em conta o formato de cada elemento a ser comunicado: o número de identificação da Segurança Social tem de ser um número com onze dígitos e tem de ser um número válido e registado na Segurança Social; as datas de nascimento, início de contrato e fim de contrato tem de obedecer ao formato AAAA-MM-DD, além disso a data de nascimento do trabalhador tem de corresponder à data registada na Segurança Social, a data de fim de contrato tem de ser igual ou posterior à data de início de contrato sendo que a mesma não pode ser doze meses anterior à data atual nem sete dias depois da data de utilização do serviço; a prestação de contrato é um campo opcional representado por um só caracter ‘P’ ou ‘T’, representando “Presencial” e “Teletrabalho” respetivamente, este campo é único no aspeto que é o único em que a Segurança Social assume o valor “P – Presencial” no caso que não seja comunicado; o código de profissão é representado por um *array* de seis caracteres, aonde os valores aceites correspondem a valores tabelados pela Classificação Portuguesa das Profissões; o local de trabalho é um número até quatro dígitos, único a cada estabelecimento, fornecido pela Segurança Social; a modalidade e motivo de contrato do trabalhador são dois campos que aceitam uma *string* até quatro caracteres, os possíveis valores encontram-se indicados em duas diferentes tabelas presentes na documentação do *web service* fornecidas pela Segurança Social; a remuneração base mensal e diuturnidades são representadas por doze dígitos, aceitando duas casas decimais; o número de horas de trabalho e o número de dias de trabalho são representados por um seis dígitos com duas casas decimais; a percentagem de trabalho aceita até cinco dígitos com duas casas decimais; e por fim, o número de identificação da Segurança Social do trabalhador a ser substituído é representado por um número de onze dígitos que corresponda a um número de identificação da Segurança Social válido e presente no sistema da Segurança Social.

### Redação

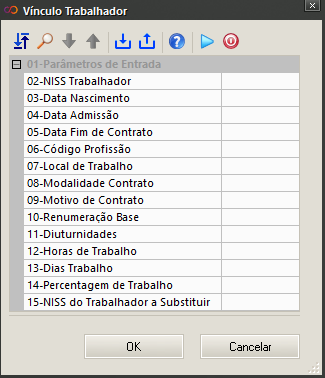
Como indicado na secção 1.1, no período em que este desenvolvimento foi iniciado a Segurança Social anunciou o descontinuar do serviço Registar Vínculo sendo substituído por um serviço mais completo e robusto. Uma limitação da implementação atual da funcionalidade é que só permite o utilizador pré-visualizar a informação a ser enviada; caso o utilizador queira alterar ou corrigir algum dos dados ele tem de sair do ecrã existente e ir alterar diretamente na ficha do trabalhador. A adição da capacidade de edição dos dados a serem comunicados levanta a possibilidade da comunicação de informação inválida e embora estes erros sejam identificados pelo serviço, foi decidido, com o objetivo de promover uma melhor experiencia de utilizador, efetuar a validação dos campos a serem comunicados só permitindo o utilizador efetuar o pedido ao serviço caso todos os campos encontrem-se válidos. Depois de efetuado o pedido é necessário apresentar a resposta do mesmo ao utilizador; em caso de sucesso, dado que os dados enviados podem ser diferentes dos presentes na ficha do trabalhador, é necessário atualizar a base de dados com essa nova informação. Por fim, para auxiliar a deteção e tratamento de erros depois de a funcionalidade ser lançada para o mercado, foi decido implementar a escrita de um ficheiro log que contenha toda a informação comunicada no serviço, este ficheiro será escrito antes de a aplicação comunicar com o serviço.

Dado o contexto, o problema e objetivos descritos foram determinadas as seguintes funcionalidades chave:

1. Recolha dos dados a enviar e apresentação dos mesmos no novo ecrã;
2. O ecrã de pré-visualização dos dados deve deixar o utilizador alterar o valor dos campos existentes, guardando depois essas alterações na Base de Dados em caso de sucesso;
3. Depois do ecrã de pré-visualização é apresentada um pedido de confirmação de envio;
4. Comunicar o vínculo de contrato utilizando o *web service* disponibilizado pela Segurança Social;
5. Apresentar o resultado da operação.

E os seguintes requisitos funcionais com os seus graus de importância:

1. Pré-visualização dos campos a enviar – Crítica;
2. Validação dos campos – Alta;
3. Escrita de um ficheiro log – Baixa;
4. Comunicação com o *web service* – Crítica;
5. Propagar campos alterados para a base de dados – Média.

Para a pré-visualização dos campos, e os seus valores, a serem comunicados foi decidido utilizar uma nova interface em que o foco é a apresentação e edição de dados, na imagem seguinte encontra-se um esboço da interface a implementar:

Como é possível observar, a interface irá apresentar ao utilizador todos os campos que podem ser comunicados pelo serviço. Alguns destes campos irão ter restrições no *input* que aceita dadas as restrições descritas na documentação; alguns campos também se encontrarão bloqueados dadas as opções selecionadas pelo utilizador. Depois o utilizador pré-visualizar e, possivelmente, editar os dados o utilizador clica no botão “OK”; de seguida será apresentada uma mensagem de confirmação, onde depois do utilizador confirmar, é efetuado o pedido ao *web service*. Ao receber a resposta apresenta-se a mesma e em caso de sucesso guarda-se quais queres alterações efetuadas na base de dados.

## Cessar Vínculo do Trabalhador

### Documentação

O segundo serviço a ser integrado é o cessar do vínculo de contrato do trabalhador e tal como o serviço previamente analisado, o corpo do pedido contém toda a informação a ser comunicada:

* NISS do trabalhador
* Data de fim de contrato
* Motivo de contrato
* Comunicação de desemprego
* Fundamentação

Com exceção dos campos: motivo de contrato e fundamentação, todos os campos são obrigatórios. Quanto ao relacionamento entre campos, o campo fundamentação passa a ser valor obrigatório com certos valores dos campos motivo de contrato e comunicação de desemprego. Tal como no serviço da comunicação do vínculo de contrato, também é importante ter em conta o formato de cada campo: o número de identificação de Segurança Social tem de ter 11 dígitos e ser um número registado no sistema; a data fim de contrato obedece ao formato AAAA-MM-DD e tem de ser igual ou inferior, até 60 meses, à data de utilização do serviço; o motivo de fim de contrato é representado por um array de até 4 caracteres, sendo que os valores válidos encontram-se tabelados na documentação fornecida; o campo comunicação de desemprego é binário, só aceitando os valores ‘0’ e ‘1’ significando “Não” e “Sim”, respetivamente; por último o campo fundamentação é um conjunto de até 500 caracteres.

### Redação

Com o propósito de reforçar a informação sobre as empresas e os seus trabalhadores presente na sua base dados, a Segurança Social decidiu lançar gradualmente um conjunto de *web services* que permita as empresas mais facilmente comunicar estes dados à Segurança Social sendo o serviço Cessar de Vínculo de Contrato o primeiro destes novos serviços. Como mencionado previamente, este serviço é bastante semelhante ao outro serviço deste projeto, por isso foi decido implementar o mesmo ecrã para pré-visualização dos dados a serem comunicados, validando os mesmos para que o utilizador não faça um pedido com informação inválida. Depois de enviado o pedido à Segurança Social, também será apresentada uma mensagem com a resposta do serviço. Será igualmente efetuada a criação de um ficheiro *log* para um mais fácil tratamento de possíveis futuros erros.

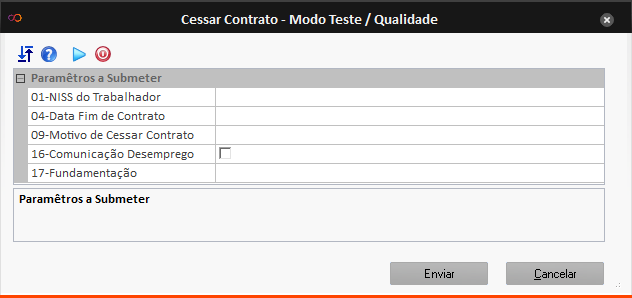
Com o seguinte contexto, problema e objetivos em mente foram determinadas as seguintes funcionalidades chave:

1. Recolha dos dados a enviar e apresentação dos mesmos no novo ecrã;
2. O ecrã de pré-visualização dos dados deve deixar o utilizador alterar o valor dos campos existentes, guardando depois essas alterações na Base de Dados em caso de sucesso;
3. Depois do ecrã de pré-visualização é apresentada um pedido de confirmação de envio;
4. Comunicar o vínculo de contrato utilizando o *web service* disponibilizado pela Segurança Social;
5. Apresentar o resultado da operação.

E os seguintes requisitos funcionais com os seus graus de importância:

1. Pré-visualização dos campos a enviar – Crítica;
2. Validação dos campos – Alta;
3. Escrita de um ficheiro log – Baixa;
4. Comunicação com o *web service* – Crítica;
5. Propagar campos alterados para a base de dados – Média.

Como mencionado previamente, a interface utilizada será a mesma do serviço Comunicação de Vínculo do Trabalhador, sendo possível visualizar o esboço da mesma na seguinte imagem:

Para a pré-visualização dos campos, e os seus valores, a serem comunicados foi decidido utilizar uma nova interface em que o foco é a apresentação e edição de dados, na imagem seguinte encontra-se um esboço da interface a implementar:

Todas as liberdades e limitações descritas para o serviço Comunicação de Vínculo do Trabalhador se irão aplicar a esta interface: a interface irá apresentar e permitir a edição de todos os campos passíveis de comunicação pelo serviço; o *input* será limitado dado os valores esperados para cada campo; o envio do pedido para o serviço só será possível depois de todos os campos passarem as validações definidas pelo serviço. Depois do utilizador clicar no botão ”OK”, ele irá ser apresentado uma mensagem para confirmar o seu desejo de enviar o pedido ao serviço, onde uma vez enviado o pedido o utilizador será apresentado com o resultado do mesmo. Por fim, quais queres alterações que o utilizador tenha efetuado na interface serão propagadas para a base de dados da aplicação ARTSOFT.

# Implementação

Como foi referido no capítulo anterior, ambos estes desenvolvimentos são bastante semelhantes; com isso em mente ambos os desenvolvimentos vão ser discutidos em paralelo, com algumas partes de discussão individual nos aspetos únicos em cada desenvolvimento.

Para começar o nosso desenvolvimento vamos primeiro identificar quais dos campos a serem comunicados não são guardados na base de dados de ERP ARTSOFT e desses mesmos campos quais fazem sentido guardar na base de dados para possível utilização futura. Relativamente ao 1º serviço, da informação pedida pelo serviço, somente dois campos não são atualmente guardados na base de dados: o motivo de contrato e o NISS do trabalhador a substituir; destes dois campos o NISS do trabalhador a substituir não é informação que vá ser utilizada noutras funcionalidades da aplicação, significando que não justifica a criação de uma entrada na base de dados para guardar esta informação. O motivo de contrato já um valor que vamos querer guardar na base de dados dado a sua potencial futura utilização. O serviço da comunicação da cessação do vínculo de contrato tem três campos que não são guardados na base de dados: o motivo de cessar contrato, comunicação para desemprego e fundamentação; destes três campos dois deles são exclusivos ao serviço, não tendo outra utilização fora do *web service*. O motivo de cessar contrato é um campo que vamos querer passar a guardar na base de dados devido à sua potencial futura utilização noutras funcionalidades.

## Tabelas

Vai então ser necessário adicionar dois campos à ficha do trabalhador e a criação de duas tabelas, uma que guarde o motivo de vínculo de contrato e outra para o motivo de cessar contrato, ambos estes campos na ficha do trabalhador irão guardar a chave primária da sua respetiva tabela no formato do tipo *uchar* de forma a poder guardar um valor numérico ocupando o mínimo de espaço possível; surgindo aqui a primeira decisão de implementação do nosso projeto: como implementar as novas tabelas. O conteúdo de cada tabela é tabelado, ou seja é definido pela Segurança Social na documentação técnica dos serviços. Isto leva-nos a duas possibilidades de implementação: implementar a tabela no código ou criar uma tabela na base de dados que guarde estes valores. O primeiro método tem várias vantagens: a tabela ser *const* levaria a uma melhor performance dado que é removido o *overhead* de acesso à base de dados e também impedia qualquer alteração em *runtime*. Contudo, esta implementação implica uma grande limitação que caso a Segurança Social decida alterar o conteúdo desta tabela é necessário lançar uma atualização de produto. A segunda implementação iria permitir a alteração do conteúdo da tabela, removendo a necessidade de lançar uma atualização de produto caso a Segurança Social atualiza-se o conteúdo da tabela, porém isto pode levar a utilizadores inexperientes alterar o conteúdo de forma a levar ao não funcionamento do serviço. Foi determinado que os custos do primeiro método de implementação não era sustentável por isso vai ser criada uma tabela na base de dados para guardar os motivos de vínculo de contrato e outra para os motivos de cessar contrato. As nossas tabelas terão a seguinte estrutura:

Vamos começar este projeto por implementar as duas tabelas referidas, isto vai envolver não só a criação da classe que representa cada tabela como também os métodos e IDs para que a aplicação consiga ter acesso às tabelas aonde seja necessário. Começando pelas classes que representam as nossas tabelas foi implementado “CTabGerMotContrato” e “CTabGerMotCessar”, ambas estas classes herdam a classe “CTabelasStd” que é a classe base de todas as tabelas. A classe “CTabelasStd” já tem implementada todos os métodos que precisamos para comunicar com a base de dados sendo que utiliza um ID que está associado à tabela para distinguir entre diferentes tabelas. Dada a estrutura das tabelas só precisamos de implementar métodos de *get* e *set* que depois chamam os métodos já implementados de leitura e escrita na base de dados de “CTabelasStd”; para completar as nossas tabelas será necessário definir os seus métodos de importar e exportar. Os métodos de importe e exporte de tabelas são globais, ambas recebendo somente um apontador de memória para a tabela a ser importada ou exportada e o caminho de aonde se pretende importar ou exportar o ficheiro. Utilizando o ID associado à tabela e um *switch* é identificado o método de importe ou exporte a utilizar. Uma vez criadas as classes é necessário criar a interface, e o acesso à mesma, que vai permitir o utilizador modificar as tabelas. Vamos alterar o menu do ecrã principal para ter mais duas entradas, uma para cada tabela, estes campos terão um ID associado que irá permitir o programa identificar qual tabela o utilizador pretende consultar. Estes novos ecrãs serão gerados pela *framework* MFC que implementa a classe “CDialog”, classe que por sua vez é herdada pela classe “CDlgCheckMultiselC”. Para criar este objeto vamos utilizar a classe “CEditTabelasBaseGer” que recebe um objeto base de dados, como por exemplo um das nossas novas tabelas, e o ID associado a essa mesma tabela; com isto ele cria uma ligação ao *handler* da tabela e um conjunto de parâmetros que define a tabela como, por exemplo, os dados a apresentar e um objeto de *callback* que neste caso será o próprio objeto CEditTabelasBaseGer. Isto implica a necessidade de implementar um método que recolha todas as entradas da tabela sobre um formato definido pela classe que irá criar o ecrã que o utilizador irá visualizar: CDlgCheckMultiselC. Quanto às regras que limitam esta interface foi decido impedir o utilizador de manipular qualquer entrada da tabela, permitindo somente alterar informação utilizando as funcionalidades de importar e exportar da interface dado que os ficheiros de importe são criados pela empresa ARTSOFT, garantindo que os mesmos se encontram corretos. Estes são os métodos que implementamos previamente e são chamados pelo *callback*, ou seja, pela classe CEditTabelasBaseGer. Para completar esta parte do desenvolvimento falta adicionar um campo no ecrã da ficha do colaborador para permitir o utilizador alterar os novos campos que foram adicionaram adicionados na ficha do colaborador. Primeiro temos de alterar o ecrã atual, adicionando duas novas *combos*, uma para cada campo.

## Web Services