

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE INFORMÁTICA



**Ciências  
ULisboa**

## **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 Resource 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 de contrato ou o cessar de contrato de um trabalhador. Esta interface irá permitir <sup>ao</sup> 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

Contexto?  
Contribuições?

Capítulo 1	Introdução .....	1
1.1	Motivação .....	1
1.2	Objetivos .....	2
1.3	Estrutura do documento .....	3
Capítulo 2	Trabalho Relacionado .....	5
Capítulo 3	Especificação de Requisitos .....	7
3.1	Estrutura da Especificação de Requisitos .....	7
3.2	Web Service .....	8
3.3	Comunicar Vínculo de Contrato do Trabalhador .....	9
3.3.1	Documentação .....	9
3.3.2	Redação .....	10
3.4	Cessar Vínculo do Trabalhador .....	12
3.4.1	Documentação .....	12
3.4.2	Redação .....	13
Capítulo 4	Implementação .....	16
4.1	Tabelas .....	16
4.2	Web Services .....	19
4.2.1	Request .....	19
4.2.2	Interface .....	20
4.2.3	Log .....	21
4.2.4	Response .....	21
4.2.5	Comunicação .....	21
Capítulo 5	Resultados .....	24
5.1	Tabelas .....	24
Capítulo 6	Conclusão .....	29
6.1	Trabalho Futuro .....	30



# **Lista de Figuras**

Figura 3.1: Fluxo dos web services da Segurança Social.....	8
Figura 3.2: Esboço Interface Vínculo Trabalhador .....	11
Figura 3.3: Esboço Interface Cessar Contrato.....	14
Figura 4.1: Estrutura Bases de Dados .....	17
Figura 4.2: Diagrama de Classes UML .....	19
Figura 5.1: Aceder às Tabelas .....	24
Figura 5.2: Tabela de Motivos de Cessação de Contrato .....	25
Figura 5.3: Tabela Motivos de Vínculo de Contrato.....	25
Figura 5.4: Menu Serviços Segurança Social.....	26
Figura 5.5: Interface Vínculo Trabalhador .....	26
Figura 5.6: Interface Erro na Validação .....	27
Figura 5.7: Interface Cessar Contrato.....	27



# **Lista de Tabelas**

# Capítulo 1

## 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 as atividades financeiras de uma empresa, as cadeias de fornecimentos, comércio, recursos humanos e outros.

? a  
empresa?

ARTSOFT é uma empresa de software, com mais de 30 anos de experiência na área, que desenvolve soluções de gestão empresarial. 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.

### 1.1 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 pode ser efetuado de duas diferentes maneiras: a empresa pode utilizar o seu acesso na

este processo

plataforma *online* da Segurança Social e aí comunicar essa mudança, ou pode-se deslocar diretamente a uma delegação da Segurança Social; ~~sendo que~~ para uma empresa este processo pode ser bastante ineficiente.

Um grande número de empresas utilizam aplicações ERP para suportar o seu negócio, ~~e~~ onde grande parte dessas aplicações disponibilizam funcionalidades de base de dados que guardam 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.

## 1.2 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. ~~Foram enumerados os seguintes objetivos de forma a obter uma solução de qualidade;~~ *os seguintes sub-objetivos devem ser alcançados:*

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 no desenvolvimento da 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~~ *se* 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, ~~é~~ 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 ~~é~~ 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.

### 1.3 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 ~~é~~ será abordada a concretização da implementação da solução do projeto, com uma apresentação dos resultados do desenvolvimento no capítulo 5. Concluindo no capítulo 6 ~~com~~ <sup>finalmente,</sup> as principais conclusões e trabalho futuro.

*Além do presente capítulo,*

 Contexto ?  
Contribuição ?



# **Capítulo 2**

## **Trabalho Relacionado**

TODO



# Capítulo 3

## Especificação de Requisitos

*focano-nos*

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.

### 3.1 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 e o esboço das interfaces relacionadas - 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.

## 3.2 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* ~~transacionando~~ *passando* 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*, onde é concatenado o nome do utilizador com sua palavra-passe codificando ambos em Base64. 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 resultados ‘2’ e ‘3’ vêm acompanhados por uma mensagem de erro descriptiva para indicar ao utilizador a causa do erro. Contudo, de notar que só é indicado um erro de cada vez, logo, para uma experiência 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:

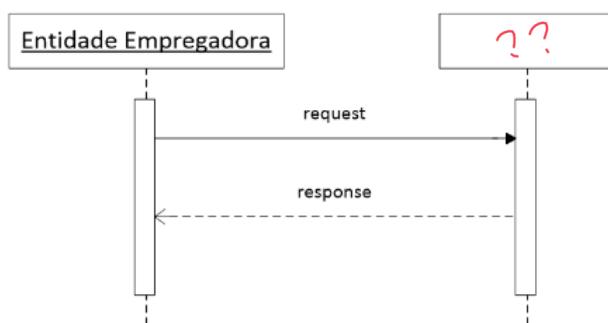


Figura 3.1: Fluxo dos *web services* da Segurança Social

### 3.3 Comunicar Vínculo de Contrato do Trabalhador

#### 3.3.1 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 serem comunicados~~ ao serviço: 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 têm 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 aspecto 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, onde os valores aceites correspondem a valores tabelados pela Classificação Portuguesa das

Profissões; o local de trabalho é um número até quatro dígitos, único <sup>para</sup> 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.

### 3.3.2 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~~ <sup>Tendo-o</sup> substituído por um serviço mais completo e robusto. Uma limitação da implementação <sup>atual</sup> 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 <sup>ele tem de sair do</sup> 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 experiência de utilizador, efetuar a validação dos campos a serem comunicados só permitindo o utilizador efetuar o pedido ao serviço caso todos os campos <sup>se</sup> encontrem ~~válios~~ 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 decidido implementar a escrita de um ficheiro log que contenha toda a informação comunicada no serviço, <sup>este</sup> 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:

- Recolha dos dados a enviar e apresentação dos mesmos no novo ecrã;
- 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;

*O fluxo do processo e dados têm de ser mais claro!*

- c) Depois do ecrã de pré-visualização é apresentada um pedido de confirmação de envio;
- d) Comunicar o vínculo de contrato utilizando o *web service* disponibilizado pela Segurança Social;
- e) Apresentar o resultado da operação.

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

- a) Pré-visualização dos campos a enviar – Crítica;
- b) Validação dos campos – Alta;
- c) Escrita de um ficheiro log – Baixa;
- d) Comunicação com o *web service* – Crítica;
- e) 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:

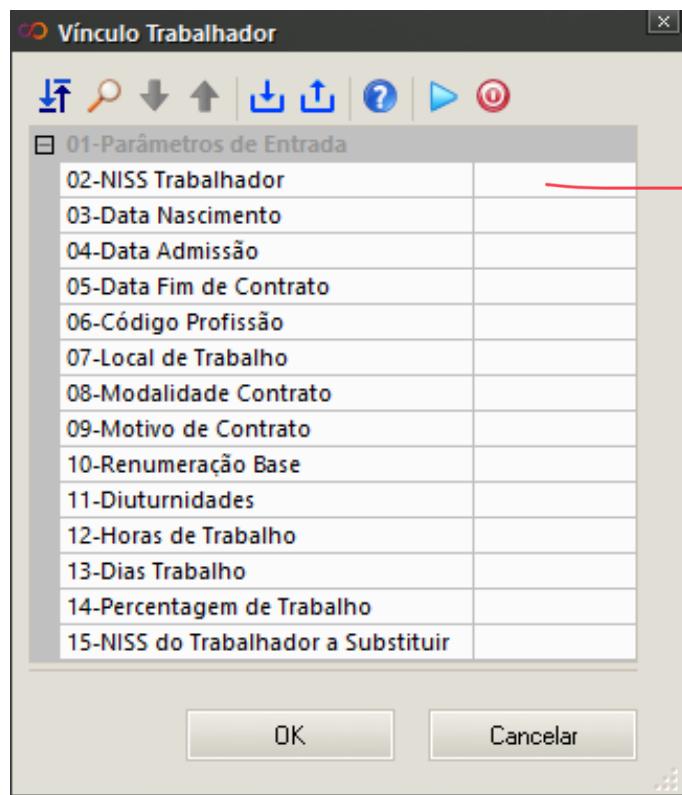


Figura 3.2: Esboço Interface Vínculo Trabalhador

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 quaisquer alterações efetuadas na base de dados.~~ *?? Né se percebe ...*

*Após o  
A resposta  
receberde é  
apresente*

## 3.4 Cessar Vínculo do Trabalhador

### 3.4.1 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 aqui 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.

### 3.4.2 Redação

Com o propósito de reforçar a informação sobre as empresas e os seus trabalhadores presentes na sua base de dados, a Segurança Social decidiu lançar gradualmente um conjunto de *web services* que permita às empresas mais facilmente comunicar estes dados à Segurança Social, sendo o serviço Cessar o Vínculo de Contrato o primeiro destes novos serviços. Como mencionado previamente, este serviço é bastante semelhante ao ~~outro~~ anteriormente mencionado serviço deste projeto, por isso foi decidido 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:

- a) Recolha dos dados a enviar e apresentação dos mesmos no novo ecrã;
- b) 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;
- c) ~~Depois do ecrã de pré-visualização~~ é apresentada um pedido de confirmação de envio;
- d) Comunicar o vínculo de contrato utilizando o *web service* disponibilizado pela Segurança Social;
- e) Apresentar o resultado da operação.

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

- a) Pré-visualização dos campos a enviar – Crítica;
- b) Validação dos campos – Alta;
- c) Escrita de um ficheiro log – Baixa;
- d) Comunicação com o *web service* – Crítica;
- e) 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:

Parametros a Submeter	
01-NISS do Trabalhador	
04-Data Fim de Contrato	
09-Motivo de Cessar Contrato	
16-Comunicação Desemprego	<input checked="" type="checkbox"/>
17-Fundamentação	

Figura 3.3: Esboço Interface Cessar Contrato

Todas as liberdades e limitações descritas para o serviço Comunicação de Vínculo do Trabalhador ~~aplicam-se~~ se irão aplicar a esta interface: a interface ~~não~~ apresentar e permitir a edição de todos os campos passíveis de comunicação pelo serviço; o ~~input~~ será limitado ~~para~~ 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 ~~a~~ uma mensagem para confirmar o seu desejo de enviar o pedido ao serviço. ~~onde~~ ~~uma vez enviado o pedido~~, o utilizador ~~será~~ apresentado com ~~do~~ o resultado do mesmo. Por fim, quaisquer alterações que o utilizador tenha efetuado na interface serão propagadas para a base de dados da aplicação ARTSOFT.

*e informe*



*Fale-me necessário descrever o contexto  
de projectos / interfaces etc, onde o trabalho  
se encontra...*

## Capítulo 4

*se encontra...*

### Implementação

*Dado se os serviços a implementar têm semelhanças,  
abordemos aqui de modo uniforme tudo o que é  
comum, intercalando*

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.

*Das informações envolvidas nos serviços em questão, há algumas que não existem na base de dados de ARTSOFT. No entanto, entre elas, há algumas que já existem.*

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 é 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. (por exemplo, que?)

#### 4.1 Tabelas

*E* 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 ~~não~~ guardam 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

*criar int??*

*ler de um  
ficheiro  
local se  
ceric  
altere os  
(pode ser  
vezel de  
certezet)*

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 atualizasse o conteúdo da tabela, porém isto pode levar a utilizadores inexperientes alterar o conteúdo de forma a levá-lo ao não funcionamento do serviço. Foi determinado que os custos do primeiro método de implementação não era sustentável por isso ~~foi~~ ~~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:

CTabGerMotContrato	CTabGerMotCessar
Primary Key: id int	Primary Key: id int
descrição varchar	descrição varchar

Figura 4.1: Estrutura Bases de Dados

*A implementação das*

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 onde 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 implementados 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 onde 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 ao 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

*Tudo isto já devia ter sido referido/descrito !!*

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~~s~~ um formato definido pela classe que irá criar o ecrã que o utilizador irá visualizar: CDlgCheckMultiselC. Quanto às regras que limitam esta interface foi decidido 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. Uma vez adicionadas as *combos* é necessário adicionar o controlador do objeto à classe responsável pela janela, sendo essa “CEmpregIdent”; ao iniciar a classe é necessário ligar o controlador ao *combo* utilizando o ID associado ao mesmo. De seguida temos de preencher a combo com o conteúdo da nossa tabela, percorrendo a nossa tabela e por cada campo da mesma utilizar o método “AddStringAndData” do controlador que permite mostrar ao utilizador só a descrição do motivo mas associando a *primary key* para depois guardar na ficha do utilizador sem ter de recorrer outra vez à tabela.

## 4.2 Web Services

Integração de modos O passo seguinte no desenvolvimento será implementar a interface ~~aonde~~ o utilizador irá pré-visualizar os dados a serem comunicados pelo serviço e ~~a~~ envio ~~do~~ do pedido para o *web service* da Segurança Social; essa implementação irá seguir os seguintes diagramas de classe:

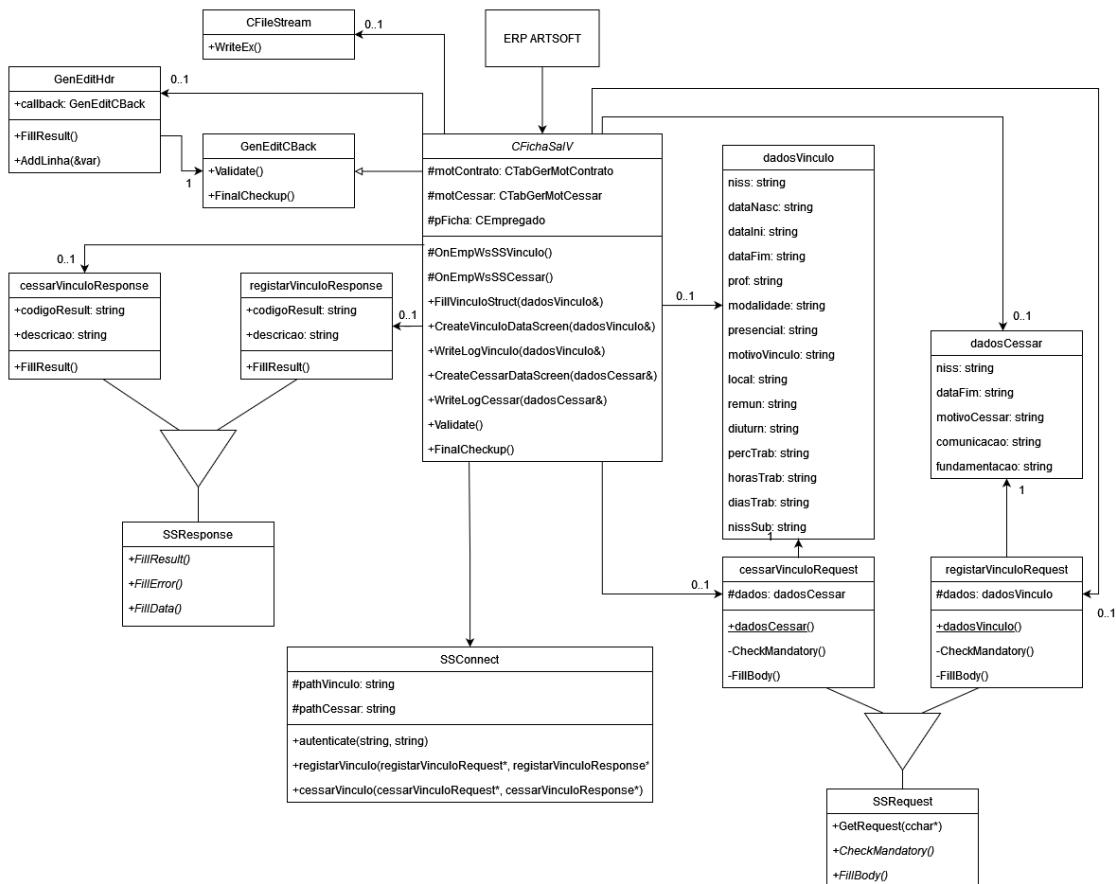


Figura 4.2: Diagrama de Classes UML

### 4.2.1 Request

Primeiro criamos vamos criar a classe que representa o pedido ao serviço, esta classe herda “SSRequest” que implementa a classe abstrata “WSRequest”. Vamos aproveitar esta classe para definir uma estrutura de dados que irá guardar toda a informação possível de ser comunicada. De seguida precisamos de implementar os dois métodos herdados da classe abstrata: “CheckMandatory” que irá simplesmente devolver 0 dado que todas as verificações serão tratadas pela interface; e “FillBody” que devolve o corpo da nossa mensagem que irá ser construído aos pedaços. Inicialmente a mensagem irá conter todos

os campos obrigatórios, adicionando um a um os campos opcionais caso os mesmos se encontrem preenchidos na estrutura de dados. Esta estrutura vai ser inicialmente preenchida recolhendo informação das tabelas e depois utilizada pela interface a apresentar ao utilizador.

#### 4.2.2 Interface

Tal como as janelas para visualizar as tabelas da base de dados, a interface para a pré-visualização e edição dos dados a comunicar vai ser criada a partir da *framework* MFC com a classe “CDialog”. Vamos então precisar primeiro de definir os parâmetros da interface, contudo para criar a interface do esboço precisamos de utilizar um conjunto diferente de classes e métodos. Para os parâmetros vamos utilizar a classe “GenEditHdr”, que comparativamente com a outra classe de parâmetros previamente utilizada, é desenhada com a edição de dados em mente permitindo definir um conjunto de regras para cada campo: como o *input* que o campo aceita; se é possível o utilizador editar o campo; ou se depois de editado o campo aciona uma chamada de *callback*. Para adicionar campos à interface utilizamos o método “AddLinha” que tem duas assinaturas diferentes dependendo do tipo de campo que estamos a tentar adicionar. A principal diferença entre as assinaturas é se o método recebe uma referência de memória para uma variável; caso não receba a interface interpreta como um nó de árvore que o utilizador pode abrir ou fechar, senão a interface interpreta como um campo que tem dados para apresentar ao utilizador. O facto de utilizar uma referência implica que qualquer alteração que o utilizador efetue na interface vai automaticamente refletir na variável fazendo as variáveis da estrutura de dados que criamos perfeita para utilizar como parâmetro do método “AddLinha”. Este método também permite definir um ID para cada campo, isto será particularmente útil para quando formos implementar as validações para não só distinguir qual é o serviço que estamos a pré-visualizar, como também para ter acesso aos valores de outros campos da interface a partir do “FindElemByID”; por isso vamos definir um *Enum* que representa o ID de cada campo a ser apresentado pela interface. Por fim vamos querer definir o *callback* da interface isto vai permitir implementar validações quando o utilizador tira o foco de certos campos e quando o utilizador tentar submeter os seus dados. Primeiro vamos ter de alterar a classe “CFichaSalV” para herdar a classe de *callback* de “GenEditHdr”, depois definir o próprio objeto “CFichaSalV” como o *callback* do objeto “GenEditHdr”, e por fim implementar o comportamento dos métodos de validação da interface. Estes dois métodos são herdados da classe “GenEditCBack”: “FinalCheckup” é chamada quando o utilizador tenta submeter os dados, em caso de erro é devolvido o ID de um dos campos que contém um erro; o outro “Validate” é chamado quando campos que contêm a *flag* “KillFocus\_CB” tem o seu valor alterado. Uma vez definido o objeto “GenEditHdr” vamos utilizar o mesmo como parâmetro do método

global “GenericEditTool” que processa estes parâmetros para um formato que a classe “CDialog” consiga utilizar.

Uma vez pré-visualizados e editados os dados, vamos iniciar uma *transaction* contendo as alterações efetuadas pelo utilizador; aproveitando o facto da classe “GenEditHdr” conter um método que verifica se o campo foi alterado permitindo evitar alteração de valores na base de dados desnecessários. Vamos também aproveitar para converter os dados que apresentamos ao utilizador para o formato requisitado pela Segurança Social.

#### 4.2.3 Log

Dada a confirmação do utilizador de enviar os dados presentes na interface, iremos escrever um ficheiro *log* que permita analisar os dados a serem comunicados pelo serviço à Segurança Social. Este ficheiro será criado numa pasta cujo propósito é guardar ficheiros relacionados com a Segurança Social, com um nome que indique o serviço que o criou e com um *timestamp* da hora e dia que foi criado.

A escrita deste ficheiro será efetuada com a classe “CFileStream”, uma classe que encapsula as funções de escrita base do C++ oferecendo ao programador uma biblioteca mais fácil de utilizar e de efetuar correção de erros. Este ficheiro vai ser escrito linha a linha, contendo cada uma variável da estrutura de dados, escrevendo todos os campos até os que se encontram vazios. Isto irá permitir ao programador identificar e corrigir potenciais erros mais rapidamente.

#### 4.2.4 Response

Para completar a implementação necessária para podermos comunicar com os serviços da Segurança Social falta implementar as classes que recebem a resposta do serviço. Herdando a classe “SSResponse”, além de guardar a resposta do serviço, esta classe é responsável por indicar o caminho do *web service* e por implementar a função “FillResult” que recebe uma resposta em formato XML e processa a mesma. A resposta XML é convertida para uma árvore de nós, cada nó guardando um valor e um nome; por exemplo, pegando no nó raiz e podemos pesquisar o nó filho que contém o código da resposta correspondendo ao resultado do serviço.

#### 4.2.5 Comunicação

Com todas as preparações feitas podemos finalmente comunicar com o serviço. Vamos começar por autenticar o utilizador concatenando os detalhes de *log in* da Segurança Social que foram inseridos previamente no menu da configuração da empresa. Esta concatenação obedecerá ao seguinte formato: “utilizador:palavra-passe”, esta *string*

depois será codificada em Base64 e adicionada como parâmetro de autenticação do *header* do pedido *http*; juntamos o corpo, que foi construído pela nossa implementação da classe do pedido do serviço, ao *header* e temos uma mensagem preparada para enviar ao serviço. De seguida conectamos ao servidor da Segurança Social que faz *host* ao serviço, efetuando testes antes de enviarmos o pedido para garantir que a ligação foi efetuada com sucesso, e enviamos a nossa mensagem pelo *socket*. Este mesmo *socket* recebe a mensagem com a resposta do serviço e analisando o código *http* de resposta determinamos se ocorreu um problema com o serviço. Contudo se o código corresponder a um dos documentados na documentação, a mensagem de resposta é processada pelo método “FillResult”. Dada a resposta do serviço apresentamos uma mensagem de erro ou sucesso do pedido ao utilizador, sendo que em caso de erro fazemos *rollback* à transação SQL que tínhamos iniciado para cancelar as alterações que tínhamos submetido na base de dados e em caso de sucesso fazemos *commit* à transação. Com a possibilidade de os dados da ficha de trabalhador terem sido atualizados forçamos a interface a recarregar a informação do trabalhador.



# Capítulo 5

## Resultados

Neste capítulo vamos apresentar recortes do ecrã a mostrar o funcionamento do projeto na sua totalidade.

### 5.1 Tabelas

A configuração das tabelas implementadas acedem-se a partir do mesmo menu acedido pelo ecrã principal do ERP ARTSOFT, como é possível observar na seguinte imagem:

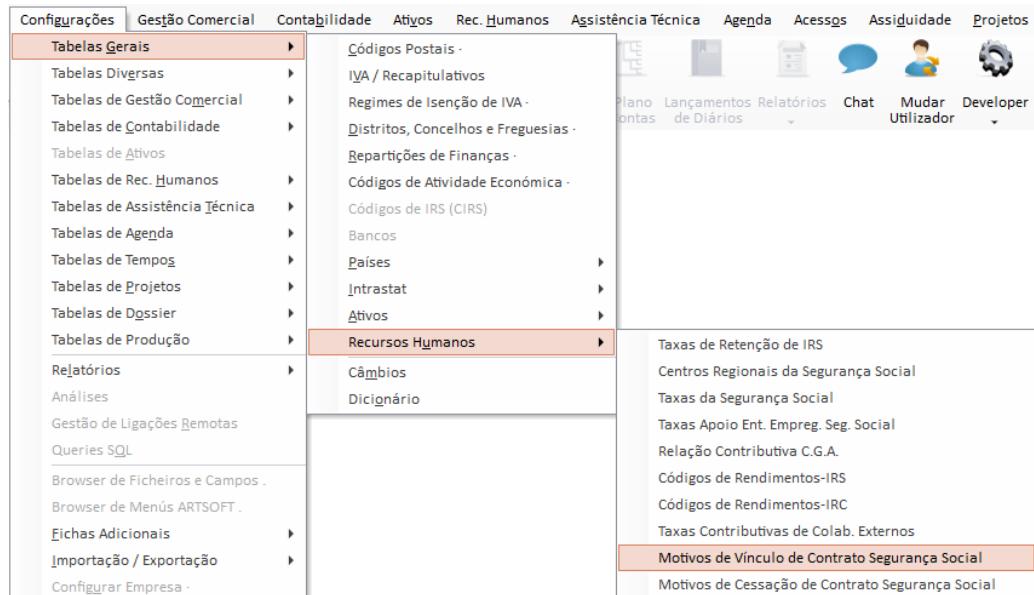


Figura 5.1: Aceder às Tabelas

**Manutenção da Tabela de Motivos de Vínculo de Contrato Segurança Social**

Código	Descrição
<b>Lista</b>	
01	AET - Acréscimo excepcional de atividade
02	ATSA - Atividade sazonal
03	CTSD - Contratação trabalhador situação desemprego muito longa dura...
04	EOPA - Execução de obra
05	EXTO - Execução tarefa ocasional
06	IFEE - Início de funcionamento de empresa/estabelecimento com menos ...
07	LNAT - Lançamento nova atividade duração incerta em empresa/estabelec...
08	STAJ - Substituição trabalhador com ação judicial despedimento
09	STAT - Substituição trabalhador ausente ou temporariamente impedido
10	STLR - Substituição trabalhador com licença sem retribuição
11	STTC - Substituição trabalhador tempo completo por prestar trabalho te...

*Esta tabela é GERAL. Qualquer alteração nesta tabela afeta todas as empresas do sistema.*

Sair

Figura 5.3: Tabela Motivos de Vínculo de Contrato

**Manutenção da Tabela de Motivos de Cessação de Contrato Segurança Social**

Código	Descrição
<b>Lista</b>	
01	CCAI - Despedimento pelo administrador de insolvência
02	CCCT - Caducidade do contrato a termo
03	CCEE - Extinção de pessoa coletiva / encerramento da empresa / morte d...
04	CCFM - Caducidade do contrato de militar
05	CCMT - Caducidade do contrato por impossibilidade superveniente abso...
06	CCRI - Caducidade do contrato por reforma por invalidez
07	CCRV - Caducidade do contrato por reforma por velhice
08	CDT - Cedência definitiva de trabalhador (Cessão da posição contratual)
09	IECC - Cessação de comissão de serviço ou situação equiparada por inici...
10	IEDC - Despedimento coletivo por iniciativa do empregador
11	IEEX - Despedimento por extinção do posto de trabalho por iniciativa do...
12	IEIN - Despedimento por inadaptação por iniciativa do empregador
13	IEJC - Justa causa por iniciativa do empregador
14	IEPE - Denúncia contrato no período experimental por iniciativa do empr...
15	IIAT - Abandono do trabalho
16	IIDD - Denúncia do contrato de trabalho/demissão por iniciativa do trab...
17	IIDE - Denúncia contrato no período experimental por iniciativa do tra...
18	IIJC - Justa causa por iniciativa do trabalhador
19	IIIA - Justa causa por salários em atraso por iniciativa do trabalhador
20	RADC - Acordo de revogação nos termos do n.º 4 do art.º 10º
21	RANE - Acordo de revogação - sem redução do nível de emprego
22	RAOT - Acordo de revogação não previsto nos números anteriores
23	RARC - Acordo de revogação - empresa em processo de recuperação
24	RARD - Acordo de revogação - empresa em recuperação judicial

*Esta tabela é GERAL. Qualquer alteração nesta tabela afeta todas as empresas do sistema.*

Sair

Figura 5.2: Tabela de Motivos de Cessação de Contrato

Uma vez configuradas as tabelas podemos utilizar os novos serviços:

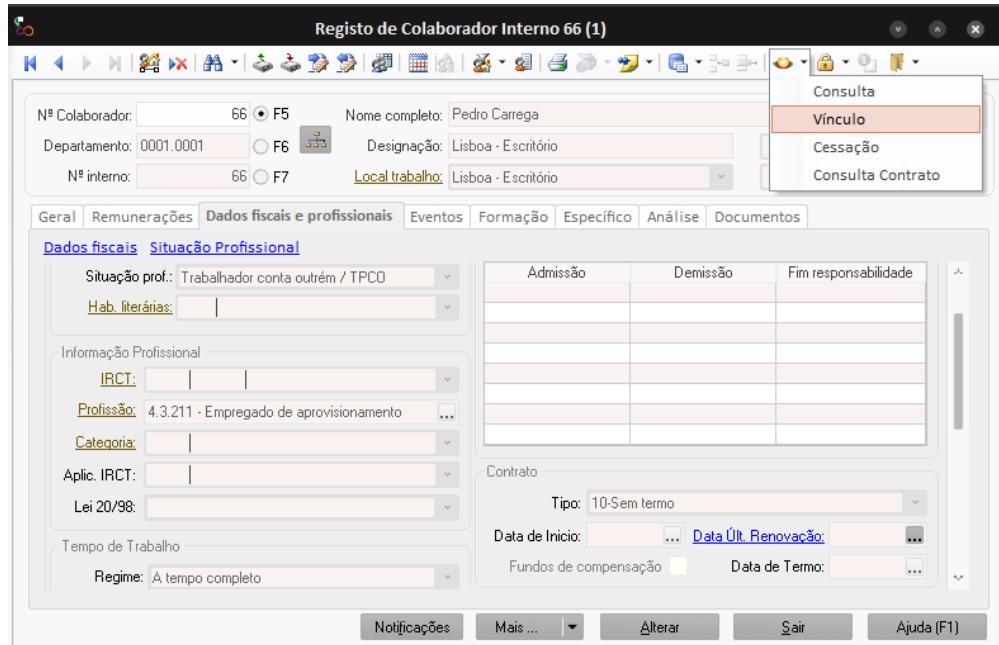


Figura 5.4: Menu Serviços Segurança Social

Se selecionarmos o serviço de comunicar o vínculo:

The screenshot shows the 'Vínculo Trabalhador - Modo Teste / Qualidade' (Employee Contract - Test / Quality Mode) window. The main area displays a table of parameters to be submitted:

Paramétricos a Submeter	Valor
NISS do Trabalhador	02-06-1997
Data de Nascimento	02-06-1997
Data Início de Contrato	12-07-2022
Data Fim de Contrato	12-06-2022
Regime do Trabalhador	Presencial
Código de Profissão	4.3.211 - Empregado de aprovisionamento
Local de Trabalho	Estabelecimento de Lisboa
Modalidade de Contrato	Sem termo, tempo completo
Motivo de Contrato	ATSA - Atividade sazonal
Remuneração Base Mensal Ilíquida	1000.00
Diuturnidades	
Nº de Horas de Trabalho	40
Nº de Dias de Trabalho	30.00
Percentagem de Trabalho	100
NISS do Trabalhador a Substituir	

Figura 5.5: Interface Vínculo Trabalhador

Caso o utilizador insira um valor inválido ou não insira um campo obrigatório aparece uma mensagem descriptiva a indicar o erro:

The screenshot shows a software window titled "Vínculo Trabalhador - Modo Teste / Qualidade". The main form contains several input fields for employee information, such as NISS, birth date, start date, end date, working regime, profession code, and percentage of work. Below the form, a modal dialog box is displayed with the title "ALERTA" (Alert). The message in the dialog states: "A data de fim de contrato tem de ser posterior à data de início de contrato: 12-07-2022". A red "X" icon is visible on the left side of the alert message. At the bottom right of the alert box is a "OK" button. At the very bottom of the window are "Enviar" (Send) and "Cancelar" (Cancel) buttons.

Figura 5.6: Interface Erro na Validação

O serviço de cessação do vínculo de contrato tem a seguinte interface:

The screenshot shows a software window titled "Cessar Contrato - Modo Teste / Qualidade". The main form contains fields for employee identification, termination date, reason for termination, communication of unemployment, and justification. Below the form, there is a section labeled "Paramétricos a Submeter" (Parameters to Submit) which is currently empty. At the bottom are "Enviar" (Send) and "Cancelar" (Cancel) buttons.

Figura 5.7: Interface Cessar Contrato



# Capítulo 6

## Conclusão

Este projeto ~~de tese~~ foi proposto pela empresa ARTSOFT à disciplina de Mestrado em Engenharia Informática da FCUL com o propósito de demonstrar o processo de efetuar desenvolvimento numa aplicação orientada a objetos. O trabalho deste projeto foi realizado durante um período de estágio na empresa ARTSOFT, na aplicação desenvolvida e comercializada internamente, ERP ARTSOFT. O objetivo era a integração de diferentes serviços desenvolvidos e disponibilizados pela Segurança Social. Em ordem a ir ~~em~~ ~~encontro~~ a esses objetivos houve um foco em efetuar um estudo da aplicação e módulos relevantes para o desenvolvimento e da documentação técnica, disponibilizada pela Segurança Social, dos serviços a integrar na aplicação.

O módulo abrangido por este desenvolvimento foi o dos Recursos Humanos, que trata de todo o *front e backend* de todas as funcionalidades relacionadas com trabalhadores da empresa.

Para este projeto foram desenvolvidas uma especificação de requisitos para cada serviço, delimitando toda a informação relevante para o desenvolvimento. Foi efetuado um estudo do problema, oferecendo contexto ao mesmo e indicando o comportamento desejado. Daí determina-se os principais objetivos do projeto, as funcionalidades chaves e os requisitos funcionais a implementar no desenvolvimento; apresentando também esboços das interfaces a implementar.

Do estudo da documentação foi determinado que teria de ser adicionada informação à base de dados de ERP ARTSOFT para satisfazer todos os requisitos; foram adicionados novos campos à tabela da ficha do trabalhador e criadas duas novas tabelas, Motivos de Vínculo de Contrato e Motivos de Cessar Vínculo de Trabalhador, para complementar estes novos campos. Foram implementadas as novas interfaces de utilizador esboçadas na especificação de requisitos. Estas permitem ao utilizador visualizarem e editarem os dados a serem comunicados aos serviços e permitem a aplicação identificar possíveis erros no *input* do utilizador.

A comunicação entre a aplicação e o servidor é efetuada a partir de um pedido *http* utilizando um formato SOAP XML. A autenticação é efetuada a partir do *header* “*authentication*”, a partir da concatenação dos detalhes de acesso à Segurança Social codificando essa concatenação em base 64. A resposta ao serviço é efetuada no mesmo

formato, tendo sido implementado um *parse* seguindo a documentação disponibilizada; da resposta do serviço apresentamos *feedback* ao utilizador sobre o resultado da operação.

## 6.1 Trabalho Futuro

No futuro irá ser efetuado trabalho iterativo e de manutenção sobre os serviços integrados para garantir e melhorar a qualidade e funcionamento dos mesmos. Será também efetuado um estudo para analisar se existe interesse do mercado em integrar os outros serviços oferecidos pela Segurança Social e caso esse interesse exista esses serviços serão integrados na aplicação ERP ARTSOFT.