

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



IMPLEMENTAÇÃO DE UMA SOLUÇÃO DE  
GESTÃO DOCUMENTAL E WORKFLOW

Rui Sérgio de Lima Elói

MESTRADO EM ENGENHARIA INFORMÁTICA

Sistemas de Informação

2010



UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



# IMPLEMENTAÇÃO DE UM SISTEMA DE GESTÃO DOCUMENTAL E WORKFLOW

Rui Sérgio de Lima Elói

ESTÁGIO

Trabalho orientado pelo Prof. Doutor André Osório Cruz de Azevedo Falcão e co-orientado por Licenciado em Eng. Informática e das Tecnologias da Informação Abel Fernando de Oliveira do Espírito Santo

MESTRADO EM ENGENHARIA INFORMÁTICA

Sistemas de Informação

2010



## **Agradecimentos**

São de realçar várias pessoas que tornaram a conclusão deste projecto possível. O meu orientador da faculdade Prof. Doutor André Osório Cruz de Azevedo Falcão que me ajudou a estruturar o corpo deste relatório, assim como na orientação de ideias e a transição do trabalho realizado para o papel. O meu Director/Coordenador da empresa Ambisig Licenciado em Eng. Informática e das Tecnologias da Informação Abel Fernando de Oliveira do Espírito Santo que sempre se mostrou disponível para dúvidas, debates, críticas e todo o tipo de troca de ideias e ajudas. Todas as pessoas que reviram os textos e deram as suas opiniões: a minha mãe, colegas, amigos, etc. De um modo geral também um obrigado a todos os professores e colegas que me ajudaram a estar apto para este tipo de desenvolvimento. Um muito obrigado a todos eles.



## Resumo

O projecto WebDoc2010 é uma solução de Gestão Documental e Workflow desenvolvida para o cliente ALRAA - Assembleia Legislativa da Região Autónoma dos Açores. Este projecto teve como principais objectivos para a empresa AMBISIG, a criação de uma aplicação Web que permita disponibilizar uma solução competitiva no mercado da gestão de documentos digitais, a reestruturação e melhoramento da lógica de negócio, melhoramento visual de modo a criar um layout agradável ao utilizador, integração com outros mecanismos de entrada e saída de documentos (fax e email). O projecto foi desenvolvido usando as tecnologias .Net Framework 3.5, JAVASCRIPT, AJAX, jQuery e SQL Server 2008. De um modo geral o resultado obtido no desenvolvimento do projecto foi de acordo com as expectativas quer da empresa, quer do cliente.

**Palavras-chave:** Gestão Documental, Workflow, .NET, AMBISIG, FAXCOMEXLIB



## **Abstract**

WebDoc2010 project is a Document Management and Workflow solution developed for ALRAA - Assembleia Legislativa da Região dos Açores. This project had as main targets for the company AMBISIG, the creation of a web application that allows to provide a competitive solution in the digital document management market, restructuring and improving the business logic, visual improvement to create a nice and friendly layout to user, integration with other forms of input and output of documents (example: fax and email). The project was developed using the technology. Net Framework 3.5, JAVASCRIPT, AJAX, jQuery, and SQL Server 2008. In general the results obtained in developing the project were in line with expectations of the company and client.

**Keywords:** Document Management, Workflow, .NET, AMBISIG, FAXCOMEXLIB



# Conteúdo

Capítulo 1	Introdução.....	1
1.1.	AMBISIG.....	1
1.1.1.	Mapa de Processos .....	4
1.1.2.	Integração na Equipa e Formações .....	4
1.3.	Âmbitos do projecto.....	5
1.4.	O que é uma gestão documental?.....	5
1.5.	WebDoc2010 .....	6
1.6.	Relatório.....	7
Capítulo 2	Enquadramento.....	8
2.1.	Plataformas de desenvolvimento .....	8
2.2.	Ferramentas Utilizadas.....	9
2.3.	Software Existente .....	9
2.4.	Soluções Anteriores .....	10
Capítulo 3	Projecto.....	11
3.1.	Modelos de diagramas de aplicação.....	11
3.1.1.	Arquitectura física do sistema.....	11
3.1.2.	Arquitectura aplicacional do sistema .....	12
3.1.3.	Casos de uso.....	12
3.1.2.	Diagrama de packages .....	21
3.2.	Lista de funcionalidades Desenvolvidas .....	22
Capítulo 4	Trabalho Realizado .....	24
4.1.	Envio e Recepção de Faxes.....	24
4.2.	Projecto Ambisig.Logging.....	30
4.3.	Configurações e interface adaptável ao cliente.....	35
4.4.	Estruturação de classes de negócio .....	38
Capítulo 5	Conclusões .....	42
5.1.	Envio e Recepção de Faxes.....	43

5.2. Ambisig.Logging .....	43
5.3. Estruturação de classes de negócio .....	43
Capítulo 6 Referências .....	44



# Lista de Figuras

Figura 1 - Mapa de Processos do SGI da AMBISIG .....	4
Figura 2 – Arquitectura física do sistema .....	11
Figura 3 - Arquitectura applicacional do sistema .....	12
Figura 4 - Caso de uso do utilizador – Meta funcionalidades.....	12
Figura 5 – Caso de uso do utilizador – Gerir Documentos.....	13
Figura 6 - Caso de uso do utilizador – Gerir Processos .....	15
Figura 7 – Caso de uso do utilizador – Gerir Entidades .....	16
Figura 8 - Caso de uso do utilizador – Gerir tipos de assunto .....	17
Figura 9 - Caso de uso do utilizador – Gerir Frases personalizadas .....	18
Figura 10 - Caso de uso do utilizador – Gerir delegações .....	19
Figura 11 - Caso de uso do administrador .....	20
Figura 12 - Diagrama de packages do WebDoc2010 .....	21
Figura 13 - Exemplo estrutura de classes do WebDoc2010 .....	22
Figura 14 - Exemplo da estrutura das classes físicas do WebDoc2010.....	22
Figura 15 - Arquitectura de Gestão de faxes .....	24
Figura 16 - Diagrama de classes da funcionalidade de envio faxes.....	25
Figura 17 - Exemplo da lista de faxes recebidos .....	26
Figura 18 - Interface de envio de faxes.....	27
Figura 19 - Exemplo da lista de faxes enviados .....	27
Figura 20 - Exemplo de um ficheiro de configurações .....	31
Figura 21 - Diagrama de arquitectura do Ambisig.Logging.....	31
Figura 22 - Diagrama de classes do Ambisig.Logging .....	32
Figura 23 - Directório do cliente.....	35
Figura 24 - Diagrama da arquitectura das configurações adaptáveis ao cliente .....	36
Figura 25 - Exemplo de um ficheiro de configuração do cliente.....	36
Figura 26 - Classe “DesignTools” .....	37
Figura 27 - <i>packages</i> “XML” .....	37
Figura 28 - Estrutura física WebDocLibrary .....	39

Figura 29 - Modelo de classes das *packages* do WebDocLibrary ..... 39



# Lista de Tabelas

Tabela 1 - Métodos da classe FAX .....	25
Tabela 2 - Métodos da classe Logger.....	32
Tabela 3 - Métodos e classes da package Log4Net .....	33
Tabela 4 - Métodos de uma “BaseClass” .....	41

# Acrónimos

**AJAX** - *Asynchronous JavaScript and XML*

**ALRAA** – Assembleia Legislativa da Região Autónoma dos Açores

**AMBISIG** – Ambiente e Sistemas de Informação Geográfica, SA

**CLR** - *Common Language Runtime*

**CMMI DEV** - *Capability Maturity Model Integration for Development* – Nível 5

**CMMI SRV** – *Capability Maturity Model Integration for Services* - Nível 2

**CSS** – *Cascade Style-Sheet*

**DREC** – Direcção Regional de Educação do Centro

**FCUL** – Faculdade de Ciências da Universidade de Lisboa

**GA** – Gestão de Arquivo

**GD** – Gestão Documental

**ISMS** – *Information Security Management System*

**JQuery** - *JavaScript Library*

**MOREQ** – *Model Requirements for the Management of electronic records*

**MVC** – *Model-View-Controller*

**PEI** – Projecto Engenharia Informática

**SCAMPI** - *Standard CMMI Appraisal Method for Process Improvement*

**SIGI** - Sistema de Gestão Integrado

**TFS** – *Team Foundation Server*

# Capítulo 1

## Introdução

Este relatório tem como objectivo, detalhar e explicar todo o processo de desenvolvimento e realização do projecto “Implementação de uma solução de Gestão Documental e *Workflow*” realizado pelo aluno Rui Sérgio de Lima Elói, na cadeira “Projecto em Engenharia Informática” do Mestrado em Engenharia Informática na Faculdade de Ciências da Universidade de Lisboa. O projecto desenvolvido, é constituído por um sistema de Gestão Documental e *Workflow* para o cliente ALRAA – Assembleia Legislativa da Região Autónoma dos Açores, embora seja uma solução que será comercializada para outros clientes. Nas alíneas abaixo deste capítulo, é possível identificar as vantagens deste projecto, quer para a Empresa, quer para o aluno, sendo também descrita a estrutura do relatório.

Neste primeiro capítulo, será apresentada a empresa na qual o projecto foi realizado e suas metodologias. Vai também ser apresentado o projecto desenvolvido, seus objectivos e planeamento.

### 1.1. AMBISIG

A AMBISIG é uma empresa que desenvolve soluções informáticas á medida. Foi criada em 1994. Entre as várias soluções desenvolvidas destaque para a digitalização das organizações (Gestões Documentais entre outras), Sistemas de Informação Geográfica, *eBank Solutions* para o mercado Financeiro e *MobileMapping* para a gestão integrada de infra-estruturas de transporte rodoviário e ferroviário. Emprega de momento cerca de 60 profissionais nas áreas de Engenharia Informática, Engenharia Ambiental, Qualidade, Geografia, Engenharia Civil, Marketing, Arquitectura, Gestão, Design e Comunicação Empresarial.

A AMBISIG tem como visão a optimização e desenvolvimento das organizações através da tecnologia e como missão a disponibilização de soluções robustas, fiáveis e *user friendly* que proporcionem aos clientes ganhos de eficácia, rentabilidade e melhoria de qualidade no seu trabalho.

O modelo de gestão da AMBISIG é denominado por SGI (Sistema de Gestão Integrado) que tem em vista cobrir os seguintes referenciais normativos:

- NP EN ISO 9001:2008 – Sistemas de Gestão da Qualidade – Requisitos
- NP EN ISO 14001:2004 + Emenda 1:2006 – Sistemas de Gestão Ambiental – Requisitos e linhas de orientação para a sua utilização
- ISO/IEC 27001:2005 – Sistemas de Segurança de Informação – Requisitos
- NP 4457:2007 – Requisitos do sistema de Gestão de IDI
- CMMI DEV - Capability Maturity Model Integration for Development – Nível 5
- CMMI SRV – Capability Maturity Model Integration for Services - Nível 2

Este modelo de gestão descreve os cinco módulos de políticas presentes actualmente na AMBISIG, estes são: “Políticas Organizacionais”, “Políticas Qualidade”, “Políticas Ambiente”, “Segurança de Informação” e “Políticas Investigação, Desenvolvimento e Inovação”.

As “Políticas Organizacionais” são sustentadas pelo plano estratégico e pelos objectivos estabelecidos. Este compromisso assenta na garantia do adequado funcionamento do Sistema de Gestão Integrado, definido de forma a sustentar práticas que vão em linha com os requisitos das normas de referência consideradas pela empresa.

As “Políticas Qualidade” assentam em quatro compromissos, satisfazer sistematicamente os clientes, cumprindo os seus requisitos, tanto no que respeita aos produtos fornecidos como ao serviço prestado. Assegurar a elevada competência de todos os colaboradores e um clima que favoreça o seu desenvolvimento técnico. Melhorar de forma contínua a eficácia do nosso funcionamento, não só reagindo adequadamente às falhas, mas também antecipando potenciais problemas. Para finalizar, promover uma gestão por objectivos e monitorizar continuamente os indicadores processuais, de forma a incrementar o desempenho dos mesmos.

As “Políticas Ambientais” assentam em seis directrizes, a Administração da AMBISIG define a Política Ambiental da organização, assegura a execução das directrizes nela contidas e a sua divulgação a todos os colaboradores. A empresa considera a protecção do Meio Ambiente uma importante responsabilidade, e zelará para que esta seja levada a cabo através da consecução de objectivos e metas. A empresa revê regularmente o seu desempenho ambiental, visando a detecção de oportunidades de melhoria. A empresa crê que o compromisso de melhoria contínua e de prevenção da

poluição deverá ser o mote orientador para direccionar esforços no sentido de procurar uma maior compatibilidade ambiental com as técnicas do processo aplicadas. A empresa sensibiliza e forma os seus colaboradores relativamente aos aspectos ambientais ligados às suas actividades e funções, motivando-os para que o seu comportamento no posto de trabalho constitua o reflexo dessa sensibilização ambiental. A empresa elabora anualmente programas ambientais, a partir desta responsabilidade partilhada, que tornam possível a melhoria contínua do seu desempenho ambiental.

A “Segurança de Informação” encontra-se descrita nos documentos POL.01.VV - Política da Segurança de Informação e POL.02.VV - Políticas Detalhadas da Segurança de Informação e assenta em seis princípios, “Audiência”, “Valor da Informação”, “Importância da Segurança de Informação”, “Modelo de Segurança da Informação”, “Responsabilidades da Segurança de Informação” e “Manutenção e Comunicação da Política de Segurança”. O princípio “Audiência” descreve que todos têm de estar em conformidade com a Política de Segurança de Informação e com os restantes documentos relacionados com a Segurança de Informação. O princípio “Valor da Informação” indica que é da responsabilidade de todos os colaboradores contribuírem pro-activamente para a protecção da informação, inclusive aquando da partilha de informação sensível verbalmente. O princípio “Importância da Segurança de Informação” afirma que deve ser assegurada a manutenção prevenindo a materialização de riscos inerente, para mitigar/limitar os potenciais danos provocados pela exploração de vulnerabilidades e incidentes de segurança, garantindo assim que o negócio opera conforme esperado. O “Modelo de Segurança da Informação” assenta em 3 factores: Confidencialidade, Integridade e Disponibilidade. O princípio “Responsabilidades da Segurança de Informação” define as políticas de Segurança de Informação definem os objectivos de controlo tal como devem ser aplicados a toda a empresa. Para finalizar o princípio “Manutenção e Comunicação das Políticas de Segurança” indica que se deve assegurar uma comunicação efectiva das políticas de segurança de informação a todos os colaboradores de modo a que estes fiquem cientes das obrigações individuais sobre este tema.

As “Políticas Investigação, Desenvolvimento e Inovação” assentam em cinco compromissos, estimular e apoiar o aparecimento de nova ideias e o seu desenvolvimento. Promover a gestão das interfaces tecnológicas, de mercado e organizacional do processo de inovação, de forma a assegurar a circulação e transferência de conhecimento entre a actividade inovadora da organização e o seu ambiente. Promover uma gestão por objectivos e monitorizar continuamente os indicadores processuais, de forma a incrementar o desempenho da inovação dentro da organização. Melhorar de forma contínua a eficácia das actividades de inovação, não só reagindo adequadamente às falhas, mas também antecipando problemas potenciais.

Assegurar uma elevada competência de todos os colaboradores, tendo em vista assegurar as competências necessárias para a investigação, desenvolvimento e inovação.

### 1.1.1. Mapa de Processos

O SGI da AMBISIG é essencialmente representado como três camadas, uma para a gestão do sistema que identifica um processo orientador de todos os restantes processos, uma para a descrição dos processos operacionais que contém os processos de referentes ao desenvolvimento, manutenção e venda dos produtos da AMBISIG e ainda uma outra camada que identifica todos os processos de suporte. Esta representação encontra-se na figura seguinte (Figura 1).

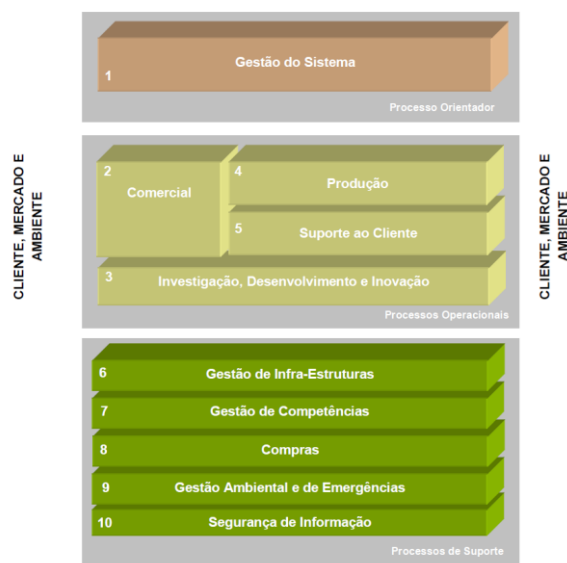


Figura 1 - Mapa de Processos do SGI da AMBISIG  
(SGI da AMBISIG 2010)

De modo a apresentar a realidade da empresa os parágrafos anteriores com as descrições da empresa e suas política foram todos recolhidos do manual de SGI da empresa AMBISIG.

### 1.1.2. Integração na Equipa e Formações

Existindo já uma ligação contratual entre o aluno e a empresa desde inícios de 2008, a integração na empresa já se encontra bem enraizada e mecanizada, pelo que não foi preciso no decurso do projecto passar por um processo de adaptação. A equipa na qual o aluno se encontra inserido é constituída por um coordenador técnico e dois outros colaboradores programadores. No decurso do trabalho foram requisitados outros recursos humanos para o desenvolvimento, neste caso a inclusão de um designer gráfico, formadores e testers.

Ao longo do estágio ocorreram formações, algumas com impacto directo e indirecto no desenvolvimento do projecto. Foi realizado uma formação de um componente Web (ActiveX) que permite ao utilizador operar sobre um documento alojado no servidor. Esta formação foi produtiva para o desenvolvimento do projecto, pois deu a conhecer uma nova ferramenta, suas funcionalidades e configurações. As restantes formações tiveram como objectivo a apresentação dos procedimentos existentes nos processos de desenvolvimento e serviços de acordo com as certificações de CMMI realizadas durante o prazo do projecto (desde o final do ano de 2009 que a AMBISIG é certificada em CMMI DEV Nível 5 e encontra-se em fase final de avaliação de CMMI SRV Nível 3).

### **1.3. Âmbitos do projecto**

Para a empresa AMBISIG o âmbito deste projecto teve como principal foco o desenvolvimento de um produto de gestão documental, que além de conter todos os requisitos pedidos pelo cliente, também servirá para a criação de um produto melhorado, quer em nível de funcionalidades, quer em nível de metodologias de desenvolvimento (em comparação com produtos já existentes na empresa).

Para o instituto de educação, neste caso FCUL, este projecto será também usado como projecto final da cadeira “Projecto de Engenharia Informática”, do curso Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa pelo aluno Rui Sérgio de Lima Elói, com o respectivo coordenador de PEI: Professor André Falcão.

### **1.4. O que é uma gestão documental?**

A Gestão Documental é um meio cada vez mais importante em qualquer empresa, as suas vantagens passam pelo acesso facilitado e rápido à informação, o cruzamento de dados e o registo de todas as acções efectuadas nos documentos.

Num processo de gestão documental, o seu início é quando ocorre a recepção do documento, passando este por uma fase de desmaterialização, ou seja, digitalização do documento geralmente em formato papel para um formato electrónico. Numa segunda fase, os documentos em formato electrónico são submetidos a uma classificação. De seguida há uma definição dos vários estados do ciclo de vida do documento ao longo da sua existência, como por exemplo, a publicação, aprovação, distribuição, reencaminhamento e anulação.

Alguns exemplos de utilizações práticas de um sistema de gestão documental são a substituição do arquivo documental em papel de uma organização, por cópias em formato digital, a digitalização dos arquivos de clientes e a digitalização de todo o

correio de uma empresa. Estes exemplos as seguintes vantagens, redução de espaço físico, acesso rápido a todos os documentos da organização e redução de papel.

O uso de um sistema de Workflow em conjunto com o sistema de Gestão Documental permite uma standardização nos processos do documento, criando uma melhor dinamização e controlo, com a salvaguarda e registo de todas as operações efectuadas.

## **1.5. WebDoc2010**

Como já referido anteriormente, este projecto teve como principal objectivo a criação de uma solução de Gestão Documental e Workflow para o cliente ALRAA - Assembleia Legislativa da Região Autónoma dos Açores.

Para além deste objectivo, o gestor deste projecto Abel Espírito do Santo em conjunto com os elementos da direcção técnica, acharam por bem aproveitar o presente projecto para melhorar e otimizar algumas funcionalidade e metodologias de desenvolvimento presentes na empresa. Como tal, este projecto será também uma espécie de esboço (não pode ainda ser considerado um protótipo) da nova versão do produto de gestão documental da AMBISIG, mais informações sobre este aspecto do projecto serão apresentadas no capítulo 4.

De referir que este produto está focado para o cliente ALRAA mas irá também ser usado para outros clientes, tratando-se deste modo duma solução “mult-cliente”. O produto é designado WebDoc2010 e pretende-se que seja uma plataforma de novas metodologias dentro da empresa e funcionalidades práticas no restante departamento de Bussiness Process Management.

O planeamento inicial encontra-se esquematizado no modelo de Gantt em Anexo. De modo a explicar melhor o modelo serão aqui descritas as fases principais: as tarefas de “Reuniões”, “Comunicações do progresso” e “Actas de Reunião” são executadas pelo gestor de projecto e são usadas para o estabelecimento de requisitos, recolha de feedback do cliente e justificação do trabalho produzido ao cliente. As tarefas de “Plano de Projecto” e “Criação inicial de Estrutura Técnica do Projecto” são também tarefas executadas pelo gestor de projecto que em análise dos requisitos e reuniões com a equipa técnica cria o plano de desenvolvimento do projecto. As tarefas dentro do “Desenvolvimento Técnico do projecto” são as mais demoradas e onde se vai desenvolver a maior parte do projecto, estas são executadas pelos programadores, designers, consultores e testers. Nestas tarefas engloba-se a análise e levantamento de processos da organização, desenvolvimento das funcionalidades de acordo com os requisitos e a bateria de testes. Para finalizar o projecto são executadas as tarefas de instalação no cliente da aplicação e a formação aos utilizadores.

De um modo geral o planeamento foi seguido, ouve tarefas que “derraparam” como é o caso das tarefas “Importação e parametrização de Workflows”, “Desenvolvimento de requisitos” e “Instalação da solução em Produção” devido a principalmente indecisões do cliente e erros na recolha de Workflows. Embora tenha havido tarefas em que o seu prazo não foi cumprido ouve outras que demoraram menos que previsto como é o caso das tarefas “Testes finais” e “Desenho de formulários” devido ao aumento dos recursos na tarefa dos “Testes finais” e à simplicidade dos formulários desenhados. De acordo com o sistema de SGDB da AMBISIG (intranet) o projecto desviou-se cerca de 9% do previsto. O desvio foi causado pela inclusão de mais recursos na tarefa dos “Testes finais” e pelo aumento de algumas horas de desenvolvimento.

## **1.6. Relatório**

O restante documento encontra-se dividido em cinco capítulos. Um capítulo com o enquadramento do projecto, onde serão descritas as plataformas, ferramentas e soluções anteriores que permitiram a execução do projecto. No capítulo seguinte será apresentado o projecto, sua arquitectura, finalidade, diagramas de casos de uso, diagramas de classes e lista de funcionalidades desenvolvidas. O capítulo “Trabalho realizado” descreve em pormenor as funcionalidades desenvolvidas. Para finalizar um capítulo com as conclusões do projecto e referências usadas no presente documento.

O relatório actual pretende reportar as funcionalidades desenvolvidas pelo aluno Rui Elói dentro do projecto WebDoc2010, sua contribuição dentro do projecto e de uma maneira geral justificar o trabalho realizado ao longo de nove meses na empresa AMBISIG. De referir que o WebDoc2010 não foi o único projecto desenvolvido pelo aluno, tendo integrado o desenvolvimento de outros projectos, quer na componente de desenvolvimento quer na componente de manutenção e consultoria.

## Capítulo 2

### Enquadramento

Este capítulo visa descrever as principais plataformas de desenvolvimento do projecto assim como as ferramentas disponíveis para o desenvolvimento deste projecto, componentes externos que auxiliam o desenvolvimento e a existência de projectos anteriores, que serão utilizados como base para o projecto a desenvolver.

#### 2.1. Plataformas de desenvolvimento

Existem duas grandes plataformas de desenvolvimento no qual o projecto se baseia. A plataforma de desenvolvimento. NET versão 3.5 SP1 e a plataforma de repositório de dados SQL Server 2008.

A plataforma de desenvolvimento. NET é uma Framework da Microsoft que possui visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações de modo que qualquer código gerado é executado em qualquer dispositivo que tenha essa Framework instalada, permitindo assim o desenvolvimento para uma plataforma em vez do desenvolvimento para um dispositivo ou sistema específico. A plataforma .NET é executada sobre uma CLR (Common Language Runtime) interagindo com um conjunto de bibliotecas (Framework). Esta CLR é capaz de executar várias linguagens de programação, interagindo entre si como uma única. As grandes vantagens desta plataforma são como referido antes o desenvolvimento uniforme para uma única plataforma em vez do desenvolvimento específico, outra grande vantagem é o vasto leque de informação disponível em livros e internet que permite ao programador um desenvolvimento mais rápido e acompanhado e para finalizar a responsabilização de uma entidade pela evolução e estabilização da plataforma (Microsoft).

A plataforma SQL Server 2008 é um SGBD (sistema de gestão de bases de dados) relacional criado pela Microsoft. Um SGBD permite criar repositórios de dados para uma ou várias aplicações obedecendo a regras relacionais. A versão 2008 é uma plataforma de dados confiável, produtiva e inteligente que permite a execução de aplicações críticas mais exigentes com reduzido tempo e custo de desenvolvimento.

Uma das grandes alterações desta versão é a introdução de um sistema intellisense na criação de queries.

## **2.2. Ferramentas Utilizadas**

As ferramentas utilizadas para este projecto são as ferramentas base para qualquer desenvolvimento na AMBISIG, as escolhas destas ferramentas são efectuadas pela Administração em conjunto com a Direcção Técnica.

O principal ambiente de desenvolvimento é a ferramenta Visual Studio 2008 na linguagem de programação VB.NET, embora as linguagens de programação HTML e Javascript também tenham sido bastante usadas, dado que o sistema é uma aplicação Web. De modo a guardar os dados da aplicação, como já referido no ponto anterior, o sistema usa o motor SQL do SQL Server 2008, usando a ferramenta SQL Server Management 2008. Para finalizar, de modo a manter um registo de versões, cópias de segurança, e desenvolvimento cooperativo, é usada a ferramenta Team Foundation Server que integra com o Visual Studio.

## **2.3. Software Existente**

De modo a aumentar a produtividade de algumas funcionalidades, o desenvolvimento desta solução, faz uso de software (componentes) interno e externo. De seguida serão apresentados esses componentes e suas aplicações.

- Framework da AMBISIG (componente interno)
- Aspose Total
- Telerik
- ABCPDF

A Framework da AMBISIG é um conjunto de Class Libraries, cada uma dessas Class Libraries possui funções e métodos que permitem simplificar e melhorar a tempo de desenvolvimento do programador da AMBISIG. A Framework da AMBISIG possui também um pacote de controlos Web que têm inerentes a lógica de negócio de Workflows.

O componente Aspose Total é um conjunto de vários componentes criados pela Aspose. Permite uma variedade de aplicações que vão desde a conversão de Words em PDF ou vice-versa a acesso de caixas Exchange.

O componente “Telerik: RadControls for ASP.NET Ajax” é um conjunto de vários controlos Web que permitem uma melhor usabilidade e performance com recurso á

tecnologia Ajax e jQuery. Para além destas vantagens o design dos controlos é bastante atractivo.

O componente ABCPDF é um componente que permite a edição de PDF's, bastante usado nas soluções da AMBISIG para a criação de PDFs recorrendo a HTML.

## **2.4. Soluções Anteriores**

O Projecto a desenvolver vai ser baseado maioritariamente em duas aplicações já existentes na AMBISIG: WebDoc2008 e a “Aplicação de Gestão Documental e Workflow da DREC”. Ambos estes projectos foram desenvolvidos em ASP.NET, utilizando a .Net Framework v3.5 e ambos usam bases de soluções anteriores. Para o desenvolvimento do projecto WebDoc2010 foi definido que seriam recolhidas as duas funcionalidades e metodologias do projecto WebDoc2008. Criação de uma Class Library com toda a lógica do negócio. No projecto WebDoc2008 já existia a metodologia da criação de uma Class Library para a agregação de toda a lógica de negócio da Gestão Documental. Infelizmente essa Class Library encontrava-se bastante incompleta e desorganizada. A outra funcionalidade/metodologia recolhida foi a criação de um design atractivo. No projecto WebDoc2008 o uso do controlo RadToolBar oferece um design e funcionalidade ao estilo do Microsoft Office 2007.

Do projecto “Aplicação de Gestão Documental e Workflow da DREC” foi recolhido toda a lógica de negócio e estrutura da base de dados pois era no momento de arranque do projecto WebDoc2010 a solução mais recente e mais robusta. De referir que este projecto foi também desenvolvido por mim, antes do início do estágio, de modo que já me encontrava bastante integrado na lógica do negócio e organização do código da solução.

## Capítulo 3

### Projecto

Este capítulo visa detalhar todas as características do projecto desenvolvido, funcionalidades desenvolvidas e metodologias implementadas. Serão apresentados diagramas de modo a auxiliar uma melhor percepção do universo do projecto desenvolvido.

#### 3.1. Modelos de diagramas de aplicação

##### 3.1.1. Arquitectura física do sistema

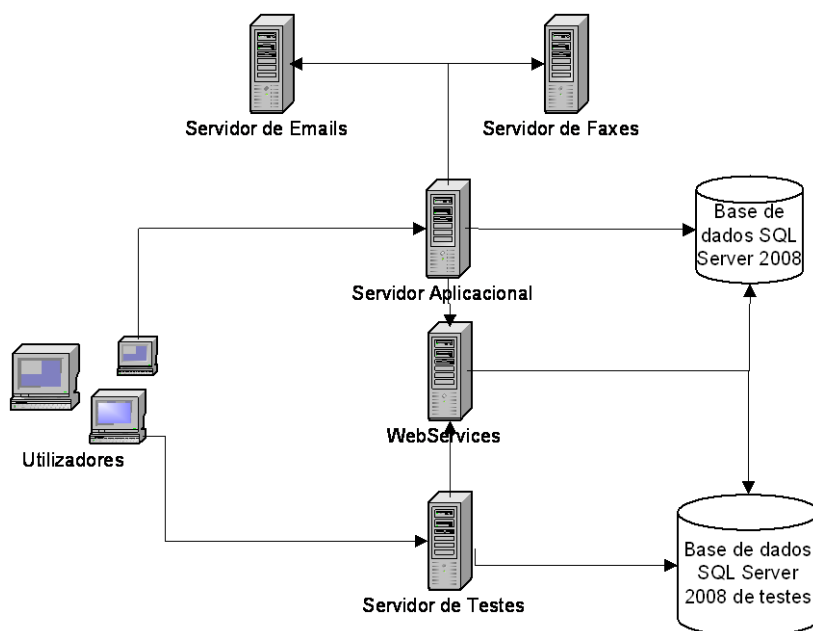


Figura 2 – Arquitectura física do sistema

O sistema desenvolvido possui a arquitectura acima descrita (Figura 2). A aplicação de gestão documental vai estar alojada num servidor Web, que vai permitir a recolha de pedidos por parte do utilizado via Web e aceder a um “DataStorage”, que contém a base de dados da aplicação em SQL Server.

A existência de um servidor para envio de Faxes via Web e envio de Emails (SMTP), a ser acedido pela aplicação.

Como é política da empresa, para cada aplicação são criados ambientes de testes, de modo a quando for pretendida uma alteração/correção ao código o cliente possa previamente testar as funcionalidades antes de actualizar no ambiente de produção.

### 3.1.2. Arquitectura applicacional do sistema

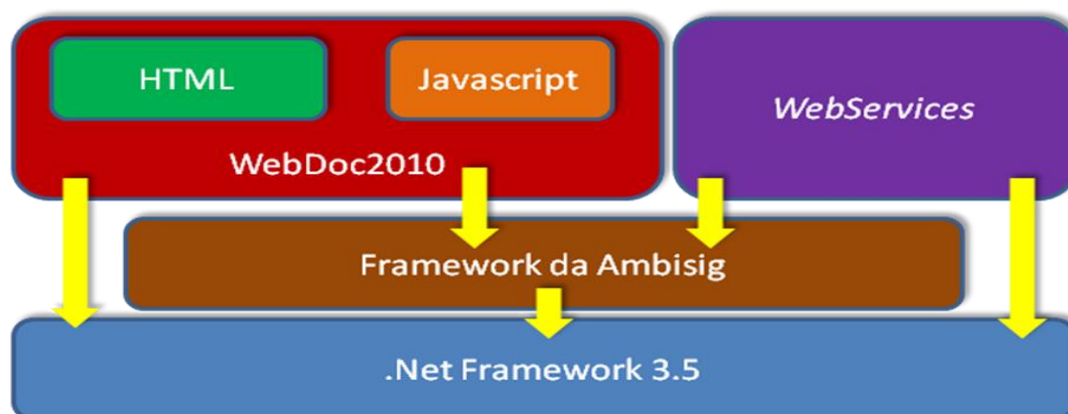


Figura 3 - Arquitectura applicacional do sistema

Como é possível visualizar na figura 3, a arquitectura applicacional é baseada fundamentalmente na Framework .NET da Microsoft e na Framework da Ambisig, sendo que a Framework da Ambisig consome por sua vez recursos da Framework .NET. O WebDoc2010 por sua vez gera e apresenta ao utilizador interfaces usando HTML e Javascript. A camada de WebServices é paralela ao WebDoc2010.

### 3.1.3. Casos de uso

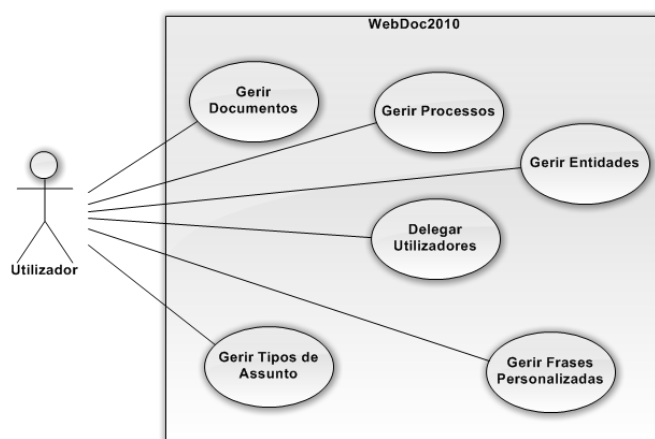


Figura 4 - Caso de uso do utilizador – Meta funcionalidades

Este diagrama (Figura 4) serve para exemplificar quais as meta actividades que um utilizador poderá recorrer ao aceder à aplicação. Muitas destas actividades têm uma

permissão associada logo nem todos os utilizadores podem executar todas estas actividades (o administrador, por seu lado, pode executar qualquer uma destas tarefas). De referir que não existem mais nenhuns perfis estipulados, só administrador e utilizador genérico. Qualquer utilizador que não seja administrador é considerado um utilizador genérico, as suas acções são reguladas pelas permissões que possuem (dadas pelo administrador) não estando ligadas directamente a um perfil.

De seguida Serão apresentados os casos de uso específicos para cada uma destas meta funcionalidades.

### 3.1.3.1. Contexto dos documentos

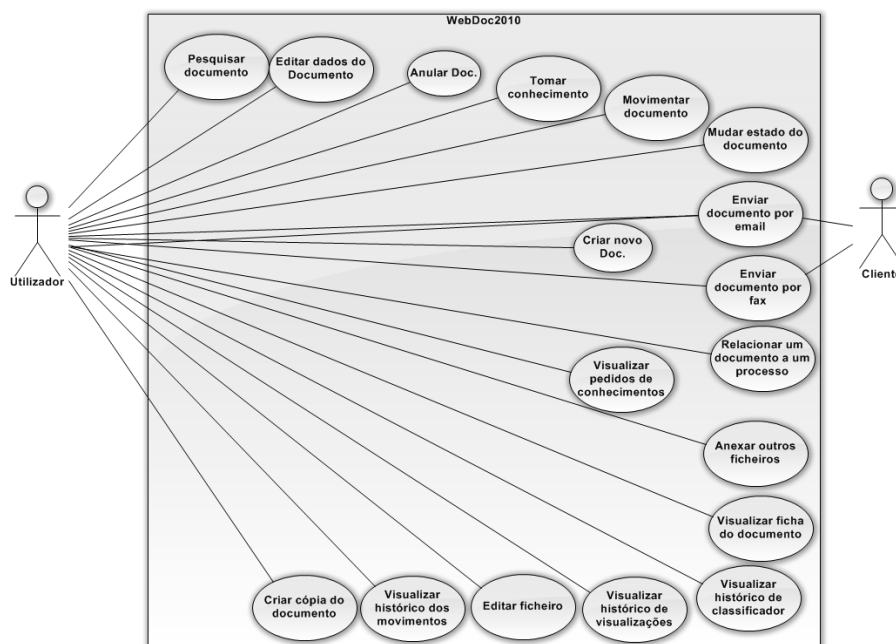


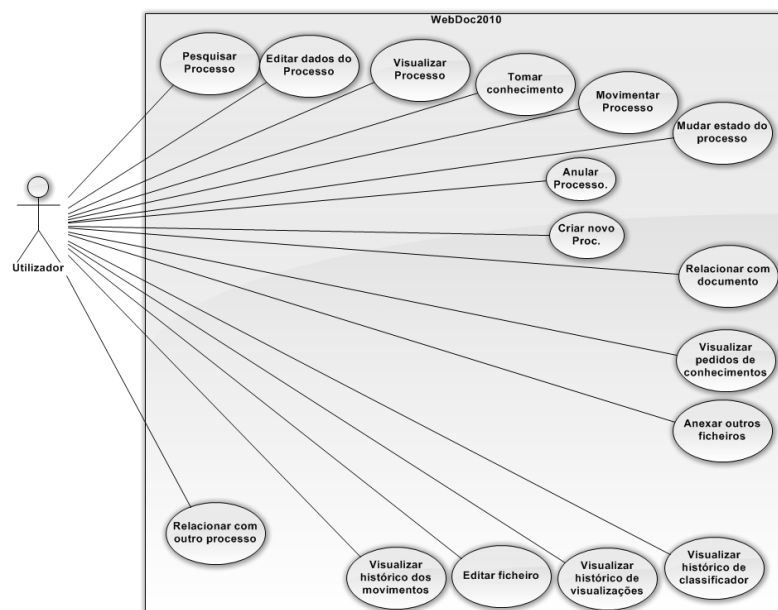
Figura 5 – Caso de uso do utilizador – Gerir Documentos

Este diagrama (Figura 5) visa descrever, mais detalhadamente, quais as acções possíveis do contexto do documento. Como é possível visualizar, existem 17 operações possíveis para o utilizador:

- **Pesquisar documento:** o utilizador pode pesquisar por vários campos de um documento.
- **Criar novo documento:** o utilizador pode criar um novo documento, inserindo a meta data e anexando o ficheiro.
- **Editar dados do Documento:** o utilizador se tiver o documento na sua posse (ou permissões) pode modificar a meta data do documento.
- **Movimentar documento:** o utilizador pode, ao avançar no Workflow, movimentar o documento para outra pessoa ou grupo.
- **Mudar estado do documento:** o utilizador pode avançar o documento no estado do Workflow.
- **Anular Doc. :** se o utilizador tiver permissões, este pode anular o documento. O documento nunca pode ser apagado.

- **Enviar documento por correio electrónico:** o utilizador pode enviar o ficheiro do documento para um ou mais destinatários, via WebDoc2010.
- **Enviar documento por fax:** o utilizador pode enviar o ficheiro do documento para um ou mais destinatários, via WebDoc2010.
- **Relacionar um documento a um processo:** se o utilizador tiver permissão, pode relacionar um documento a um processo. Um documento pode estar associado a vários processos.
- **Visualizar pedidos de conhecimentos:** o utilizador pode, quando executa uma movimentação, dar conhecimento a um outro utilizador ou grupo. Esta acção permite ver a lista de todos os conhecimentos dados para um determinado documento.
- **Anexar outro ficheiro:** é possível associar novos anexos ao documento.
- **Visualizar histórico do classificador:** sempre que é alterado o classificador do documento, este é registado numa lista, podendo depois ser visualizada pelos utilizadores.
- **Visualizar histórico de visualizações:** sempre que um documento é visualizado é criado um registo, que pode ser visualizado pelo utilizador.
- **Editar ficheiro:** um documento tem sempre um ficheiro associado, este documento pode ser editado pelo utilizador se este tiver permissão.
- **Visualizar histórico dos movimentos:** o utilizador pode ver todos os movimentos do documento.
- **Criar cópia do documento:** o utilizador pode efectuar uma cópia integral do documento.
- **Tomar conhecimento:** se foi efectuado um pedido para conhecimento o utilizador tem de efectuar uma tomada de conhecimento, que regista quando é que este teve conhecimento do documento.
- **Visualizar ficha do documento:** o utilizador pode visualizar a ficha do documento que contém toda a meta data, relações e outras características do documento.

### 3.1.3.2. Contexto dos processos



## Figura 6 - Caso de uso do utilizador – Gerir Processos

Este diagrama (Figura 6) visa descrever, mais detalhadamente, quais as acções possíveis do contexto do processo. Como é possível visualizar, existem 16 operações possíveis para o utilizador:

- **Pesquisar processo:** o utilizador pode pesquisar por vários campos de um processo.
- **Criar novo processo:** o utilizador pode criar um novo processo, inserindo a meta data e anexando o ficheiro (não obrigatório).
- **Editar dados do processo:** o utilizador se tiver o processo na sua posse, ou permissões, pode modificar a meta data do processo.
- **Movimentar processo,** o utilizador pode ao avançar no Workflow e movimentar o processo para outra pessoa ou grupo.
- **Mudar estado do processo:** o utilizador pode avançar o processo no estado do Workflow.
- **Anular Proc. :** se o utilizador tiver permissões, este pode anular o processo. O processo nunca pode ser apagado.
- **Relacionar um processo a um documento:** se o utilizador tiver permissão, pode relacionar um processo a um documento. Um processo pode estar associado a vários documentos.
- **Relacionar com outro processo:** se o utilizador tiver permissão, pode relacionar um processo a outro processo.
- **Visualizar pedidos de conhecimentos:** o utilizador quando executa uma movimentação, pode dar conhecimento a um outro utilizador ou grupo. Esta acção permite ver a lista de todos os conhecimentos dados para um determinado processo.
- **Anexar outro ficheiro:** é possível associar novos anexos ao processo.
- **Visualizar histórico de classificador:** sempre que é alterado o classificador do processo, este é registado numa lista podendo depois ser visualizada pelos utilizadores.
- **Visualizar histórico de visualizações,** sempre que um processo é visualizado é criado um registo que pode ser visualizado pelo utilizador.
- **Editar ficheiro,** um processo pode ter um ficheiro associado, este processo pode ser editado pelo utilizador se este tiver permissão.
- **Visualizar histórico dos movimentos,** o utilizador pode ver todos os movimentos do processo.
- **Tomar conhecimento,** se foi efectuado um pedido para conhecimento, o utilizador tem de efectuar uma tomada de conhecimento que regista, quando é que este teve conhecimento do processo.
- **Visualizar ficha do processo,** o utilizador pode visualizar a ficha do processo que contem toda a meta data, relações e outras características do processo.

### 3.1.4.3. Contexto das Entidades

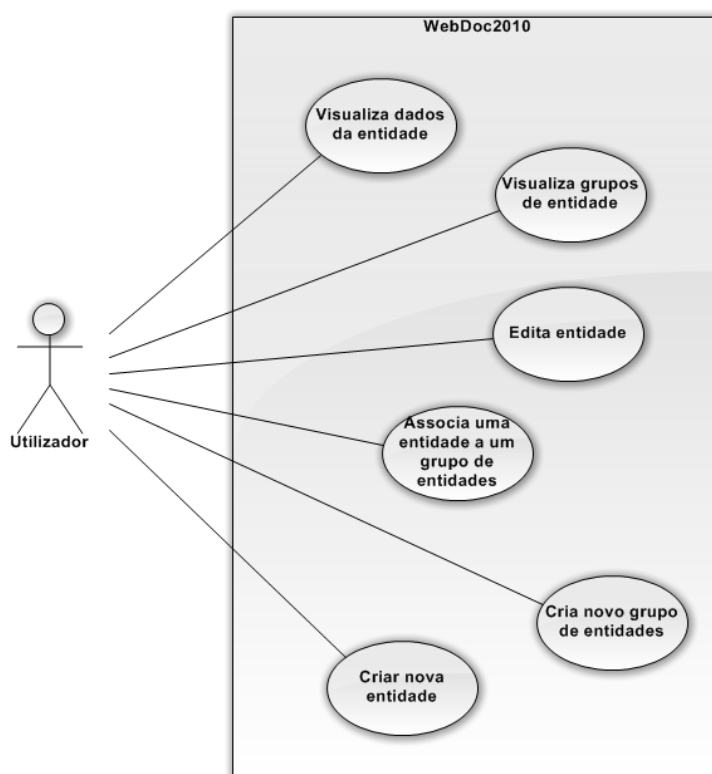


Figura 7 – Caso de uso do utilizador – Gerir Entidades

Este diagrama (Figura 7) visa descrever, mais detalhadamente, quais as acções possíveis do contexto das entidades. Como é possível visualizar, existem 6 operações possíveis para o utilizador:

- **Visualizar os dados da entidade:** o utilizador pode visualizar os dados de uma entidade. Os campos possíveis de serem visualizados estão dependentes de uma permissão
- **Visualiza grupos de entidades:** o utilizador pode visualizar quais entidades que pertencem a um dado grupo.
- **Edita entidade,** o utilizador se tiver permissão pode alterar os dados da entidade (ex: morada, telefone).
- **Cria nova entidade:** o utilizador se tiver permissão pode criar uma nova entidade. A nova entidade é visível por todos os outros utilizadores.
- **Cria novo grupo de entidades:** o utilizador se tiver permissão pode criar um novo grupo.
- **Associa uma entidade a um grupo de entidades:** o utilizador se tiver permissão pode associar uma entidade a um grupo de entidades.

### 3.1.4.4. Contexto dos tipos de assunto

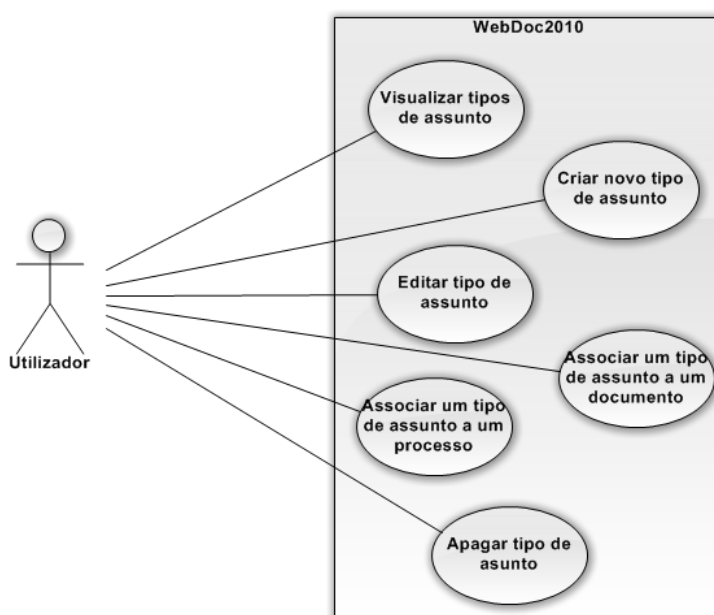


Figura 8 - Caso de uso do utilizador – Gerir tipos de assunto

Este diagrama (Figura 8) visa descrever, mais detalhadamente, quais as acções possíveis do contexto dos tipos de assunto. Um tipo de assunto serve, essencialmente, para ajudar o utilizador a estandardizar os seus tipos de assunto, de modo a que a sua pesquisa seja mais fácil. Como é possível visualizar, existem 6 operações possíveis para o utilizador:

- **Visualizar os tipos de assunto:** o utilizador pode ver os tipos assunto.
- **Criar novo tipo de assunto:** o utilizador se tiver permissão, pode criar um novo tipo de assunto.
- **Apagar tipo de assunto:** o utilizador se tiver permissão, pode apagar um tipo de assunto. Esta acção não tem repercussões nos documentos/processos já criados.
- **Associar um tipo de assunto a um documento:** o utilizador na criação do documento pode se auxiliar por uma lista de assuntos já criados.
- **Associar um tipo de assunto a um processo:** o utilizador na criação do processo pode se auxiliar por uma lista de assuntos já criados.

### 3.1.4.5. Contexto das frases personalizadas



Figura 9 - Caso de uso do utilizador – Gerir Frases personalizadas

Este diagrama (Figura 9) visa descrever, mais detalhadamente, quais as acções possíveis do contexto das frases personalizadas. Uma frase personalizada é visível e usada apenas pelo utilizador que a criou, é por exemplo, usada para melhorar a performance do utilizador na movimentação de documentos/processos. Como é possível visualizar, existem 4 operações possíveis para o utilizador:

- **Visualizar Frases personalizadas:** o utilizador pode visualizar as suas frases personalizadas.
- **Criar nova frase personalizada:** o utilizador pode criar uma nova frase personalizada.
- **Editar nova frase personalizada:** o utilizador pode altear o conteúdo da frase por si criada.
- **Apagar frase personalizada:** o utilizador pode apagar a frase personalizada por si criada.

### 3.1.4.6. Contexto das delegações

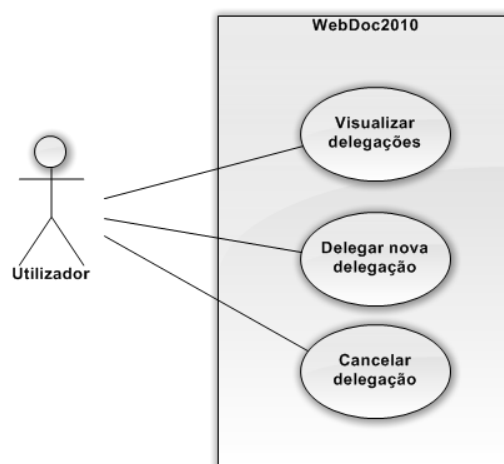


Figura 10 - Caso de uso do utilizador – Gerir delegações

Este diagrama (Figura 10) visa descrever, mais detalhadamente, quais as acções possíveis do contexto das delegações. Um utilizador pode delegar o seu trabalho, isto é, todos os seus documentos/processos e outras funcionalidades que o utilizador tenha de efectuar na Gestão Documental, vão também estar disponíveis para o utilizador delegado (bastante usado para quando um utilizador vai de férias). Como é possível visualizar, existem 3 operações possíveis para o utilizador:

- **Visualizar delegações:** o utilizador pode visualizar a quem delegou ou se alguém o delegou.
- **Delegar nova delegação:** o utilizador pode criar uma nova delegação, estabelecendo um prazo e um destinatário.
- **Cancelar delegação,** o utilizador pode remover uma delegação por si criada.

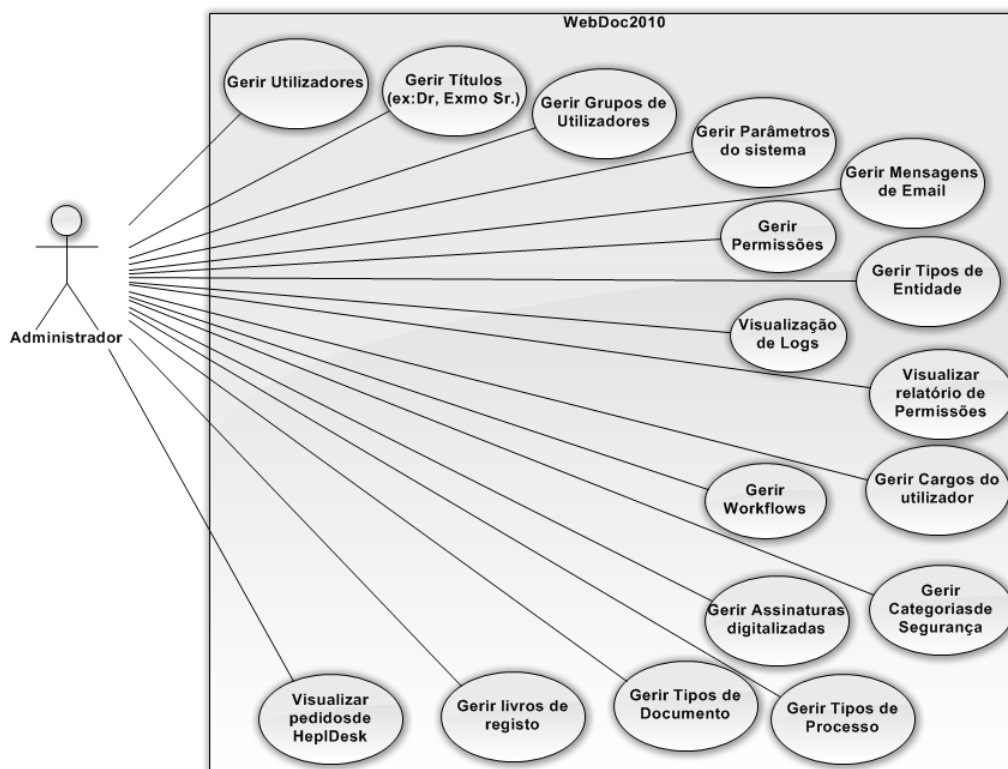


Figura 11 - Caso de uso do administrador

Este diagrama (Figura 11) serve para ilustrar todas as actividades que um utilizador do tipo administrador (pertence a um grupo de Administradores) pode executar. De referir que um administrador tem todas as permissões existentes na aplicação e consegue efectuar qualquer acção que um utilizador normal consiga.

Como é possível visualizar existem 17 operações possíveis para o administrador:

- **Gerir Utilizadores:** o administrador pode visualizar o registo do utilizador, pode editar os seus dados e pode apagar o registo. É possível também visualizar o histórico de edições aos dados do utilizador e associar o utilizador a grupos.
- **Gerir títulos:** o administrador pode visualizar o registo do título, pode editar os seus dados e pode apagar o registo. Os títulos do documento são usados para denominar num documento qual o título da entidade em questão.
- **Gerir Mensagens de email:** o administrador pode visualizar o registo da mensagem de email e pode editar os seus dados. As mensagens de email são o texto que será submetido por email, por exemplo, quando envia um documento de saída para o exterior.
- **Gerir Grupos de utilizadores:** o administrador pode visualizar o registo do grupo, pode editar os seus dados e pode apagar o registo. É possível também visualizar o histórico de edições aos dados do grupo e associar o grupo a utilizadores.
- **Gerir parâmetros do sistema,** o administrador pode visualizar o registo do parâmetro e pode editar os seus dados. Nem todos os parâmetros podem ser editados.
- **Gerir permissões,** o administrador pode relacionar uma permissão a um grupo ou utilizador. Um utilizador herda permissões dos grupos a que está relacionado.

- **Gerir tipos de entidade**, o administrador pode visualizar o registo do tipo de entidade, pode editar os seus dados e pode apagar o registo.
- **Visualização de logs**, o administrador pode visualizar os logs existentes.
- **Visualizar relatório de permissões**, o administrador pode visualizar as permissões associadas a um utilizador ou grupo ou por permissão.
- **Gerir Workflows**, o administrador pode importar novos Workflows ou reimportar. Pode também editar os dados de um Workflow ou acciona-lo como inactivo.
- **Gerir categorias de segurança**, o administrador pode visualizar o registo a categoria de segurança, pode editar os seus dados e pode apagar o registo.
- **Gerir assinaturas digitalizadas**, o administrador pode associar uma imagem de uma assinatura a um utilizador.
- **Gerir tipos de processo**, o administrador pode visualizar o registo do tipo de processo, pode editar os seus dados e pode apagar o registo.
- **Gerir tipos de documento**, o administrador pode visualizar o registo do tipo de documento, pode editar os seus dados e pode apagar o registo.
- **Gerir livros de registo**, o administrador pode visualizar o registo do livro de registo, pode editar os seus dados e pode apagar o registo.
- **Visualizar pedidos de helpdesk**, o administrador pode visualizar os pedidos de helpdesk e editar o seu conteúdo.

### 3.1.2. Diagrama de packages

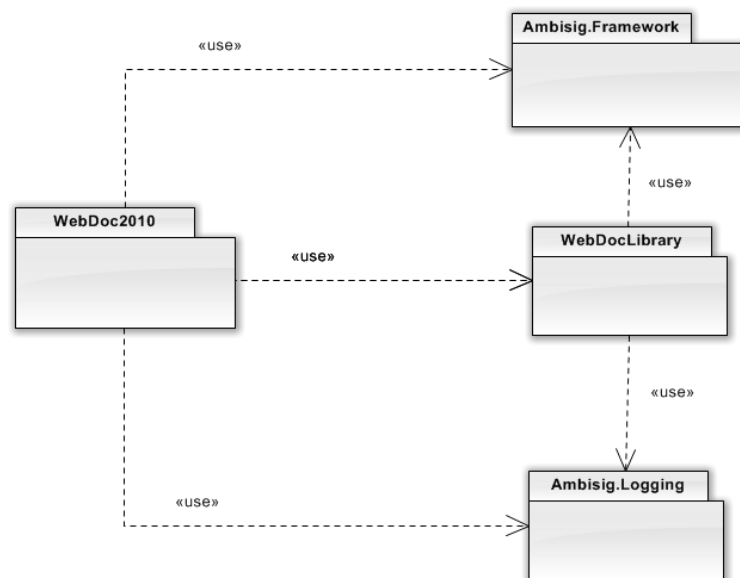


Figura 12 - Diagrama de packages do WebDoc2010

O diagrama (Figura 12) exemplifica as packages da solução (sem contar com as packages externas). Como é possível visualizar a solução WebDoc2010 faz uso de três grandes componentes internos: Ambisig.Logging para o registo de logs, WebDocLibrary para a camada de negocio da GD e Ambisig.Framework que possui todos os componentes básicos da AMBISIG.

### 3.1.2.1. Modelo de classes do WebDoc2010

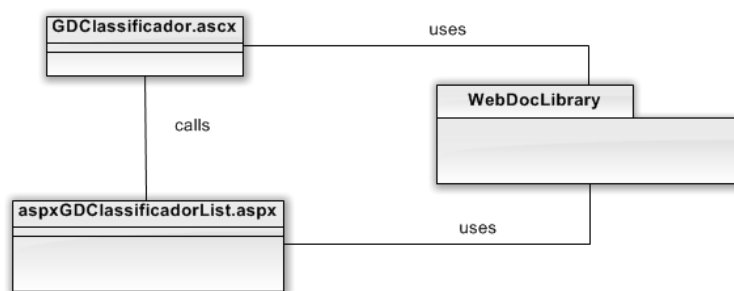


Figura 13 - Exemplo estrutura de classes do WebDoc2010

Dado que a solução do **WebDoc2010** é bastante extensa irá ser apresentado só a estrutura de um contexto (Figura 13), neste caso do classificador. Praticamente todos os contextos do WebDoc2010 são semelhantes, um contexto consiste sempre numa listagem (exemplo classe *aspxGDClassificadorList.aspx*) que possui sempre uma pesquisa, logo faz uso das classes de listagem SQL pertencentes a *package WebDocLibrary* e chama sempre o formulário do objecto em questão (neste caso o objecto *GDClassificador.ascx*). O objecto *GDClassificador.ascx* usa o objecto *GDClassificador* pertencente á *package WebDocLibrary* (todas as classes pertencentes á *package WebDocLibrary* serão descritas no ponto **Estruturação de classes de negócio**).

De modo a descrever melhor este diagrama e apresentada também a seguinte figura que mostra a estrutura física das classes em questão:

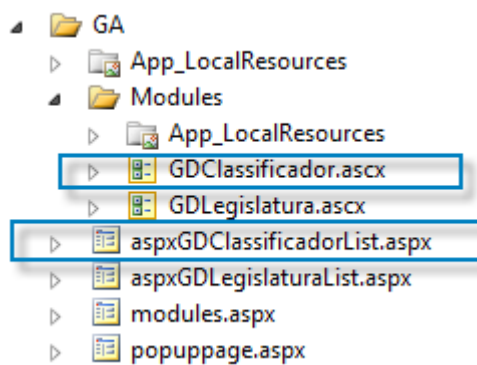


Figura 14 - Exemplo da estrutura das classes físicas do WebDoc2010

## 3.2. Lista de funcionalidades Desenvolvidas

O desenvolvimento do projecto teve como principais desenvolvimentos: a criação de um novo projecto, migração da base de dados da aplicação DREC, limpar dados dessa base de dados migrada, recolha das funcionalidades das aplicações anteriormente identificadas, criação de um projecto WebDoclibrary com todas as classes de negócio,

desenvolvimento da funcionalidade de Envio de Faxes, desenvolvimento do projecto Ambisig.Logging, desenvolvimento da funcionalidade “Configurações e interface adaptável ao cliente”, execução de testes, formação ao formador da AMBISIG, apoio à formação e esclarecimento de dúvidas do cliente e para finalizar correcção de bugs e manutenção do sistema.

## Capítulo 4

### Trabalho Realizado

Neste capítulo serão descritas algumas funcionalidades que foram realizadas no âmbito do projecto. Devido á extensibilidade do projecto só serão apresentadas estas 3 funcionalidades pois estas são as mais relevantes em termos de importância e inovação nos produtos da AMBISIG. De referir que praticamente todo o projecto foi realizado por mim, excepto a criação de layouts e algumas funcionalidades foram realizadas por outro colaborador.

#### 4.1. Envio e Recepção de Faxes

Esta funcionalidade consiste na criação de um módulo de gestão de faxes com ligação directa á consola de faxes de um servidor *Microsoft*, ou seja, esta funcionalidade consiste em acções de registo de entrada de faxes e de envio de documentos por fax.

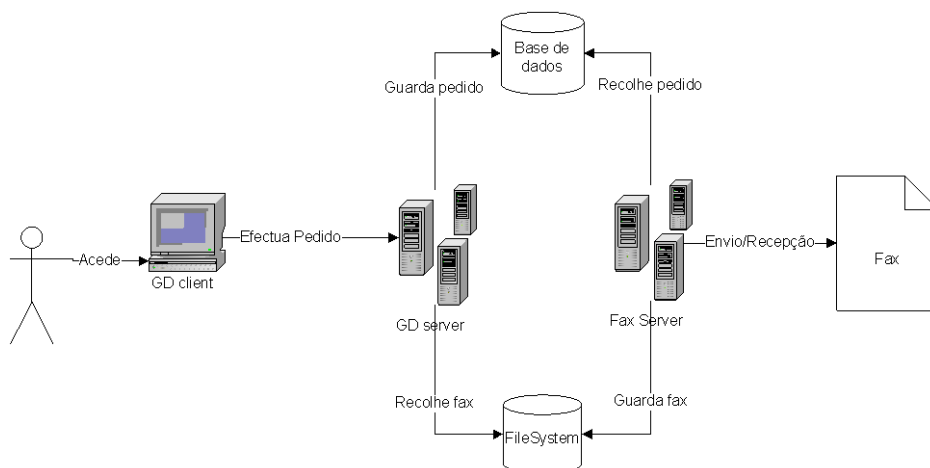


Figura 15 - Arquitectura de Gestão de faxes

O diagrama (Figura 15) descreve a arquitectura usada por esta funcionalidade. Como podemos visualizar a arquitectura é em quase tudo semelhante às restantes

funcionalidades do projecto tendo como única diferença o recurso a um servidor para o envio de faxes. O servidor de faxes possui uma placa de rede que se encontra ligado a uma linha telefónica e é gerido pela consola de faxes da *Microsoft (Windows)*.

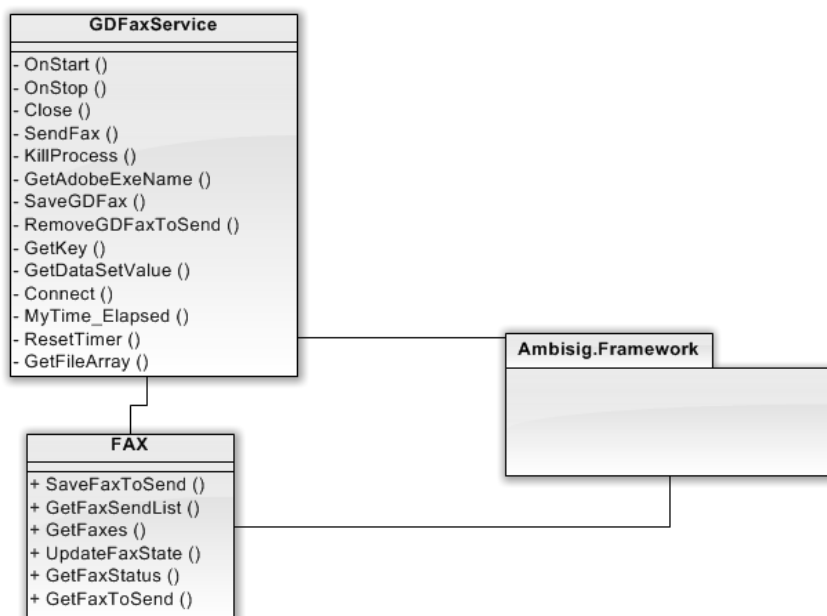


Figura 16 - Diagrama de classes da funcionalidade de envio faxes

O diagrama (Figura 16) mostra uma classe *GDFaxService* que possui as funcionalidades de um serviço Windows. Essa classe recorre a uma classe *Fax* que possui os métodos para acesso aos dados de um fax dentro da base de dados. Ambas estas classes recorrem a métodos e funcionalidades da *package Ambisig.Framework* (não estão aqui indicados quais os métodos e funcionalidades usados dentro da *package Ambisig.Framework* pois são métodos e funcionalidades típicas de acesso e tratamento de dados). De seguinte tabela são descritos os métodos destas classes:

Tabela 1 - Métodos da classe FAX

Método	Descrição
SaveFaxToSend()	Regista novo documento para ser enviado por fax. Regista os dados necessários para o servido de faxes (GDFAXSERVICE).
GetFaxSendList()	Recolhe os registos da tabela GDFAX e cria uma datatable que será usada para popular a <i>grid</i> da listagem de faxes recebidos.
GetFaxes()	Recolhe Datatable com os vários possíveis estados que se encontram na tabela GDFAXESTADO.
GetFaxStatus()	Devolve o ID do estado actual (na base de dados) do registo.

UpdateFaxState()	Faz <i>update</i> ao campo IDGDFAXESTADO
GetFaxToSend()	Recolhe o primeiro registo da tabela GDFAXTOSEND

De seguida será apresentado o procedimento para o uso desta funcionalidade dentro do WebDoc2010.

### Recepção:

1. Servidor de faxes do *Windows* recebe o fax e grava o documento numa pasta previamente configurada na consola de faxes. Nesta pasta o servidor que possui a GD precisa de ter acesso de escrita e leitura.

2. Utilizador com permissões de registo de faxes acede á página com a lista de faxes recebidos. Sistema vai ler ficheiros que se encontram na pasta de faxes recebidos.

#### LISTA DE FAXES RECEBIDOS

Ficheiro	Tamanho	Data		
ambisig.png	3331	17-11-2009 11:44:13	Guardar	✘

Figura 17 - Exemplo da lista de faxes recebidos

3. Como podemos ver pela figura 17, o utilizador pode visualizar a data de recepção e o tamanho do ficheiro. Pode também eliminar o ficheiro, pressionando o botão (✘). Quando o utilizador regista o documento, (*Guardar*) o sistema apresenta a ficha normal do documento com alguns dados já inseridos (ex: Livro de Registo e Tipo de documento). Após registo do documento o sistema, automaticamente, remove o ficheiro da pasta de faxes recebidos.

### Envio:

1. Utilizador regista um novo documento.
2. No Workflow deverá estar estipulado a transição para envio do fax. O utilizador escolhe essa transição e será apresentado o seguinte *popup* (Figura 18):

Figura 18 - Interface de envio de faxes

Onde vai ser possível ao utilizador modificar os dados de envio dos faxes, isto é, a interface vai recolher os dados da entidade mas o utilizador pode modificar esses dados, mais propriamente o nome da entidade, o número de fax assim como agregar ou não os anexos do documento no envio.

3. Após carregar no botão **Enviar Fax** o sistema cria um registo na tabela “GDFAXTOSEND” e informa o utilizador que o fax foi enviado para a lista de espera.

4. O serviço GDFAXSERVICE que se encontra a correr no servidor de faxes temporariamente acede á base de dados e recolhe o primeiro registo.

5. Quando recolhe esse registo, vai recolher o ficheiro (ou ficheiros se os anexos estiverem definidos para envio) e envia o pedido á consola de faxes do *Windows*.

6. Se o envio para a consola de faxes do *Windows* for bem sucedido o serviço regista na tabela de faxes enviados “GDFAX” e elimina o registo que se encontra na tabela “GDFAXTOSEND”.

Lista de faxes enviados:

LISTA DE FAXES ENVIADOS

Pesquisa

Assunto

Entidade

Fax

Utilizador

Documento

Estado

Pesquisar

Limpar Campos

Página: 1 de 1

Exportar

do Registo 1 ao 1 de um Total de 1

Documento	Assunto	Destinatário	Data	Nº Fax	Enviado Por	Estado
FAX Teste	Genérico -	Entidade Teste	01-02-2010 00:00:00	123456789	administrator	A Reenviar

Página: 1 de 1





Exportar

do Registo 1 ao 1 de um Total de 1

⚠ Não existem faxes enviados!

⚠ Não existe servidor de fax definido!

Figura 19 - Exemplo da lista de faxes enviados

1. A listagem de faxes enviados (figura 19) permite a cada utilizador visualizar os faxes que este enviou. Os utilizadores pertencentes ao grupo de Administradores no entanto podem ver qualquer fax enviado.
2. Cada vez que o utilizador acede á listagem
3. Nesta listagem existem funcionalidades de reenvio , reenvio para outro número  e a criação da confirmação do fax (criação de um anexo ao documento do fax com o estado do fax) . A imagem  indica se o fax já possui uma confirmação.

Esta funcionalidade oferece duas grandes vantagens ao sistema da Gestão Documental e a organização que a adquire. O facto de todo o sistema de envio e recepção de faxes se encontrar informatizado permite uma melhoria de performance pois o procedimento anterior passava por:

#### Recepção:

1. Fax é recebido pela Máquina de faxes
2. Utilizador desloca-se á Máquina de faxes
3. Utilizador desloca-se ao seu posto de trabalho
4. Utilizador separa a correspondência
5. Correspondência é enviada

#### Envio:

1. Utilizador envia Fax para expediente (ás vezes até para sector intermédios antes de chegar ao expediente)
2. Utilizador do expediente desloca-se á máquina de faxes
3. Utilizador do expediente envia fax (não pode sair enquanto a máquina de faxes não envia com sucesso o fax)
4. Utilizador do expediente recebe confirmação do fax
5. Utilizador do expediente devolve confirmação ao utilizador que enviou o pedido de envio.
6. Utilizador agrega confirmação do fax ao documento.

Como podemos reparar o facto de toda a gestão de envio e recepção ser feita pela consola de faxes do Windows, permite que o utilizador não se tenha de deslocar á máquina de faxes (em alternativa passa a aceder a uma página para registar o fax de entrada ou efectuar pedido de submissão do documento) nem que tenha de esperar pelo seu correcto envio (a listagem de faxes enviados permite ver os estados actuais de cada fax). Outra grande vantagem do uso desta funcionalidade é a possibilidade de um registo total dos faxes enviados pela organização, sendo possível visualizar esse histórico de uma maneira rápida e eficaz.

No desenvolvimento desta funcionalidade foram encontrados alguns problemas que afectarem o tempo de desenvolvimento. A falta de documentação da package

FAXCOMEXLIB, package utilizada para fazer o acesso á consola de faxes da Microsoft foi o maior desafio desta funcionalidade. A falta de tempo para uma melhor estruturação das classes limitou também a qualidade final do código produzido.

## 4.2. Projecto Ambisig.Logging

Este projecto foi desenvolvido devido ao facto dos projectos da Framework da Ambisig estarem ofuscados, no âmbito da norma ISO/IEC 27001 - Gestão da segurança da informação (*ISMS*) e de modo a criar um mecanismo fácil e rápido de implementar pelo programador.

A ofuscação dos projectos da Framework da Ambisig resulta numa grande desvantagem para o programador pois torna-se impossível correr a aplicação em *DEBUG* dentro desses projectos. A criação de um mecanismo de logs é vital para o programador ter conhecimento do que ocorre dentro desses métodos pois infelizmente a Framework da AMBISIG possui algumas lacunas no tratamento das excepções.

Existem então dois grandes objectivos na realização deste projecto, a criação de um mecanismo de logs, que possa ser suportado, quer nos projectos pertencentes á Framework da Ambisig, quer em qualquer outro projecto em .NET e a criação de um mecanismo de logs fácil e simples de implementar pelo programador no seu código.

A Framework da Ambisig já possuía uma classe responsável para a criação de logs em ficheiros, tínhamos como possíveis soluções o melhoramento desta classe ou o uso de um componente externo. Após alguma análise da minha parte e do colaborador Luís Gomes assim como elementos do departamento de R&D (Ricardo Beja e Joaquim Lourenço) optou-se pelo recurso ao componente *open source* **Log4Net** pois apresentava uma solução já bastante estável e utilizada por muitos programadores, a diminuição do tempo de desenvolvimento nos melhoramentos da classe de logs antiga e o facto de com esta ferramenta ser possível a criação de vários outputs de logs (será explicado em detalhe nos capítulos seguintes) assim como vários níveis de logs (será também explicado em detalhe nos pontos seguintes).

O projecto como referido anteriormente recorre a um pacote *open source* denominado por **log4net** que por sua vez pertence a um projecto com bastantes anos chamado *Apache Logging Services*.

Este projecto é baseado numa arquitectura já usado no componente **log4j** (para java) com melhorias nas novas funcionalidades da linguagem. NET.

O componente log4Net tem como principais vantagens, o facto de ser um componente otimizado para velocidade (performance), ser baseado em hierarquias de logs, é *Thread-safe*, o seu comportamento pode ser definido em *runtime* usando um ficheiro de configuração (XML), possibilidade de implementação de vários *loggers* definidos, possibilidade de criar vários outputs possíveis para o mesmo *logger* e o uso de níveis de logs: DEBUG, INFO, WARN, ERROR e FATAL

O uso desta ferramenta permite que um programador que queira implementar uma ferramenta de logs na sua aplicação apenas necessite de referenciar a DLL do projecto Ambisig.Logging, criar um ficheiro de configurações do log (XML) ou então o uso do web.config ou app.config (exemplo de um ficheiro de configurações na Figura 20). Bastando depois executar o método “WriteLog” para registar o log.

```

<!-- log4net configuration-->
<log4net>
  <logger name="Logger">
    <level value="DEBUG" />
  </logger>
  <root>
    <level value="debug" />
    <appender-ref ref="AdoNetAppender" />
  </root>
  <appender name="LogFileAppender" type="log4net.Appender.RollingFileAppender">
    <param name="File" value="D: \Log.log" />
    <param name="AppendToFile" value="true" />
    <rollingStyle value="Size" />
    <rollingSlyte value="Composite" />
    <maxSizeRollBackups value="10" />
    <maximumFileSize value="2MB" />
    <staticLogFileName value="true" />
    <layout type="log4net.Layout.PatternLayout">
      <param name="ConversionPattern" value="%-5p %d{yyyy-MM-dd hh:mm:ss} - %m%n" />
    </layout>
  </appender>
</log4net>

```

Figura 20 - Exemplo de um ficheiro de configurações

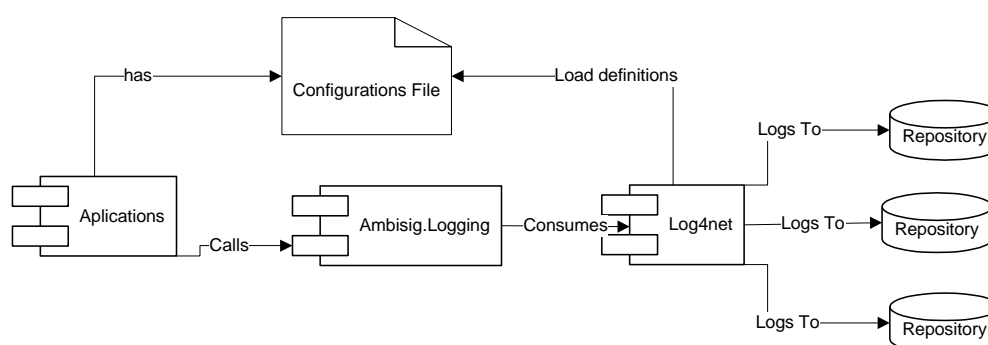


Figura 21 - Diagrama de arquitectura do Ambisig.Logging

Este diagrama (Figura 21) permite visualizar a arquitectura dos sistemas que utilizem o Projecto Ambisig.Logging. Como é possível reparar o projecto Ambisig.Logging é referenciado por um ou mais projectos que possuem um ficheiro de configuração (XML). O projecto Ambisig.Logging por sua vez recorre as funcionalidades existentes no pacote Log4net que por sua vez carrega as definições do ficheiro de configuração do

projecto e guarda os registos nos repositórios identificados nesse ficheiro de configuração.

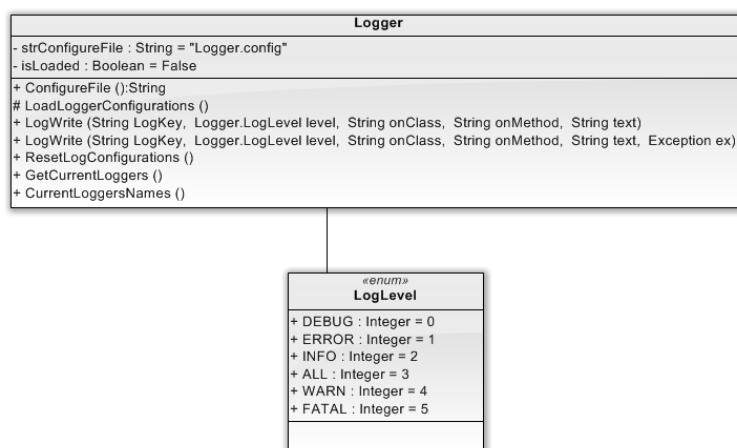


Figura 22 - Diagrama de classes do Ambisig.Logging

O diagrama de classes (Figura 22) do projecto Ambisig.Logging só a classe Logger e a enumeração “LogLevel”.

Tabela 2 - Métodos da classe Logger

Método	Descrição
<b>ConfigureFile</b>	Define qual o caminho para o ficheiro de configurações, por base é o ficheiro logger.config. Depois de modificada esta propriedade é necessário executar o método “ResetLogConfigurations” para carregar as novas definições dos loggers.
<b>LoadLoggerConfigurations</b>	Carrega o ficheiro de configuração, se não tiver nenhum especificado usa o app.config ou web.config.
<b>LogWrite</b>	Escreve a afirmação pretendida no logger. Existem duas assinaturas deste método, uma para imprimir a exceção e outra para imprimir só a mensagem.
<b>ResetLogConfigurations</b>	Faz o reset das configurações do log e carrega novamente o ficheiro de configuração. Método a executar sempre que é efectuada uma alteração ao ficheiro de configuração.
<b>GetCurrentLoggers</b>	Devolve uma lista de Ilog ou seja a lista de loggers carregados no momento.
<b>CurrentLoggersNames</b>	Devolve uma lista de strings com os nomes dos loggers carregados no momento

Tabela 3 - Métodos e classes da package Log4Net

Método	Descrição
<b>llog</b>	Classe que define um Logger. Possui o nome do Logger todas as características definidas no ficheiro de configuração assim como possui os métodos para o registo dos logs
<b>LogManager</b>	Classe que permite gerir os Loggers, ou seja, permite a criação, remoção e alteração das definições dos Loggers.
<b>XML.Configure</b>	Método que executa o carregamento das características dos Loggers

Como já foi possível reparar o processo de registo de logs é bastante “dependente” de um ficheiro de configurações. Neste ponto são apresentados exemplos de possíveis configurações.

Um *appender* é uma definição de um output do registo do log, um Logger pode ter vários outputs definidos. Na página do componente Log4Net é possível visualizar uma tabela com todos os *appenders* existentes.

No decurso do desenvolvimento deste componente foram criadas um conjunto de boas práticas no uso do mecanismo de logs, estando estas actualmente documentadas no manual de desenvolvimento do projecto Ambisig.Logging: Uso de um ficheiro de configurações separado do ficheiro de web.config ou app.config, qualquer excepção recolhida pelo mecanismo de Try-Catch tem de ficar registada no log com o nível de log ERRO ou WARN (no caso de a excepção não influenciar o bom funcionamento do método), sempre que existe uma chamada a um método potencialmente crítico é aconselhável o registo no log antes e depois da chamada e para finalizar, para cada método de cada classe pertencente a um projecto da Framework criar registos de logs no nível de DEBUG no inicio e fim do método.

O componente Log4net é um componente *open source* não pago sendo que a única condição do seu uso é o agrupamento da licença com o projecto. A licença pode estar alojada num ficheiro de texto inserido no projecto.

Esta ferramenta de logs possui várias vantagens internas, pois é uma ferramenta simples de implementar (depois de aprender a configuração). Permite um registo automático de logs que torna a sua administração mais simples. O registo do log é feito numa única linha, não é preciso inicializar componentes. Permite o uso de vários tipos de configurações e outputs, tanto para ficheiros como para bases de dados. E o objectivo desta funcionalidade, permite perceber o que ocorre dentro de produtos “fechados”, isto é, produtos que não seja possível efectuar “debug” em *runtime*.

O desenvolvimento desta ferramenta foi bastante fácil de executar, a documentação do componente log4Net está bastante acessível assim como vários exemplos. Um dos principais problemas foi compreender o funcionamento e principais configurações da ferramenta.

### 4.3. Configurações e interface adaptável ao cliente

Esta funcionalidade não é considerada uma funcionalidade do sistema mas mais um mecanismo de configuração do produto. Este mecanismo consiste na apresentação do grafismo da aplicação (ex: CSS e imagens) dependendo do cliente.

No panorama anterior ao desenvolvimento desta solução cada projecto possuía um único ficheiro de CSS o que tornava muito difícil o desenvolvimento de uma solução para mais que um cliente, a resolução deste problema passava pelo uso do mesmo aspecto gráfico para cada um dos cliente ou pela criação de um novo ambiente de desenvolvimento (duplicação de ficheiros) tornando-se muito complicado a resolução de bugs e novas funcionalidades, pois era preciso fazer as operações em cada ambiente aumentando o tempo de desenvolvimento.

Outro aspecto negativo nas metodologias anteriores, era que qualquer configuração para se adequar a cada cliente tinha de estar configurado no ficheiro de configuração da aplicação, o que tornava-se complicado de gerir pelo programador, por exemplo quando um programador puxava uma nova versão do TFS este vinha com as definições de outro cliente tendo o programador que voltar a reescrever os parâmetros de configuração do seu cliente.

Este mecanismo consiste na junção entre duas *keys* inseridas no ficheiro de configuração da aplicação (web.config) e uma classe denominada “DesignTools”. Sempre que se pretende aceder a características próprias de cada cliente é invocado um método da classe “DesignTools” de modo a retornar o caminho relativo para a pasta do Cliente da aplicação.

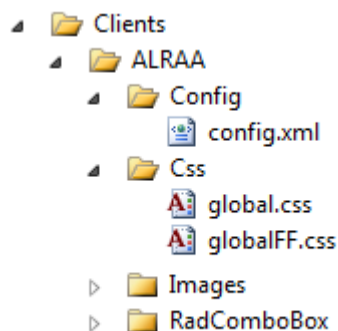


Figura 23 - Directório do cliente

Como é possível reparar na imagem anterior (Figura 23), o retorno do caminho da pasta do cliente permite ao programador aceder a toda a informação necessária do cliente, ou seja, é possível aceder á pasta de CSS's onde se encontram os ficheiros de CSS necessários, todas as imagens específicas para cada cliente (ex: banners específicos do cliente) e também o ficheiro de configuração do cliente (config.xml).

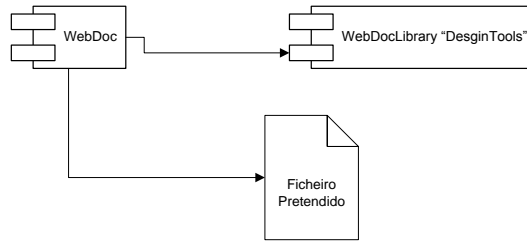


Figura 24 - Diagrama da arquitectura das configurações adaptáveis ao cliente

A arquitectura deste mecanismo (Figura 24) é bastante simples, a aplicação invoca um método da classe “DesignTools” que por sua vez retorna um caminho relativo para a pasta indicada do cliente (CSS, Config, RadCombo, etc.). Sabendo o caminho relativo para as pasta pretendida a aplicação depois acede ao ficheiro pretendido.

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <LogOnSettings>
    <ShowUserName Type="Boolean">0</ShowUserName>
    <UseDomain Type="Boolean">1</UseDomain>
    <AutoFillUserName Type="Boolean">1</AutoFillUserName>
    <DefaultDomain Type="String">ALRA</DefaultDomain>
    <ShowDomain Type="String">1</ShowDomain>
    <ValidateLastNPasswords Type="Integer">0</ValidateLastNPasswords>
    <HasExpirationDate Type="Boolean">0</HasExpirationDate>
    <ValidateNewPassword Type="Boolean">0</ValidateNewPassword>
    <MinNewPasswordLength Type="Integer">0</MinNewPasswordLength>
  </LogOnSettings>

  <GridSettings>
    <Grid name="aspxDocumentoList">
      <SortSettings>
        <sortexpression>Estado ASC, DataEntrada DESC,Numero DESC</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="aspxProcessoList">
      <SortSettings>
        <sortexpression>Leg DESC, processo.DataProcesso DESC, processo.GDPProcesso DESC</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDDocumentosRelacionados">
      <SortSettings>
        <sortexpression>DataEntrada DESC</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDProcessosRelacionados">
      <SortSettings>
        <sortexpression>GDDocumentoGDProcesso.Data, GDPProcesso.GDPProcesso</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDProcessosRelacionadosProc">
      <SortSettings>
        <sortexpression>DataProcesso DESC</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDDocumentosAnexos">
      <SortSettings>
        <sortexpression>GDDocumentoAnexo.Data</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDProcessosAnexo">
      <SortSettings>
        <sortexpression>GDProcessoAnexo.Data</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDProcessoAlertaList">
      <SortSettings>
        <sortexpression>processo.DataProcesso DESC, processo.GDPProcesso DESC</sortexpression>
      </SortSettings>
    </Grid>
    <Grid name="GDDocumentoAlertaList">
      <SortSettings>
        <sortexpression>Estado ASC, DataEntrada DESC,Numero DESC</sortexpression>
      </SortSettings>
    </Grid>
  </GridSettings>
</configuration>

```

Figura 25 - Exemplo de um ficheiro de configuração do cliente

A imagem acima (Figura 25) é um exemplo de um ficheiro de configuração do cliente, como é possível visualizar possui dois grandes módulos de configuração: **LogOnSettings** e **GridSettings**, embora o código permita a criação de n-módulos.

O módulo **LogOnSettings** descreve os parâmetros usados pelo controlo de “LogOn” da AMBISIG, cada parâmetro possui um tipo (Type=”Boolean”) e um valor.

O módulo **GridSettings** descreve os parâmetros de ordenação das datagrids de modo a que cada cliente possa ter a ordenação desejada.

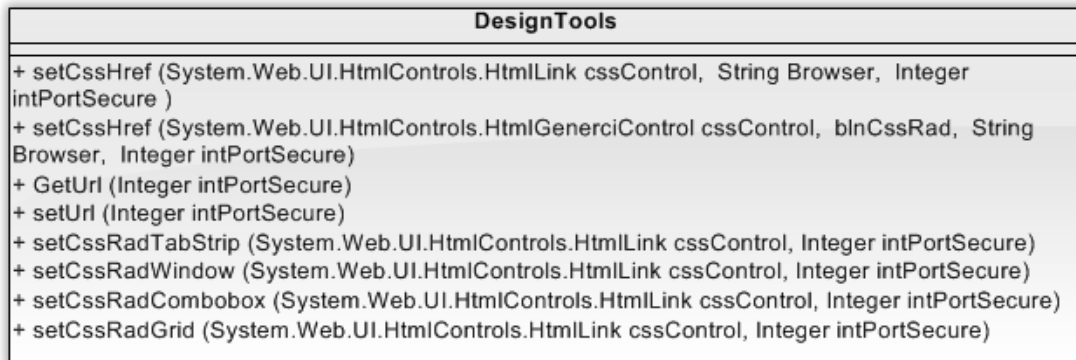


Figura 26 - Classe “DesignTools”

Cada método desta classe (Figura 26) retorna um caminho relativo para a pasta indicada dependendo da chave inserida no Web.config.

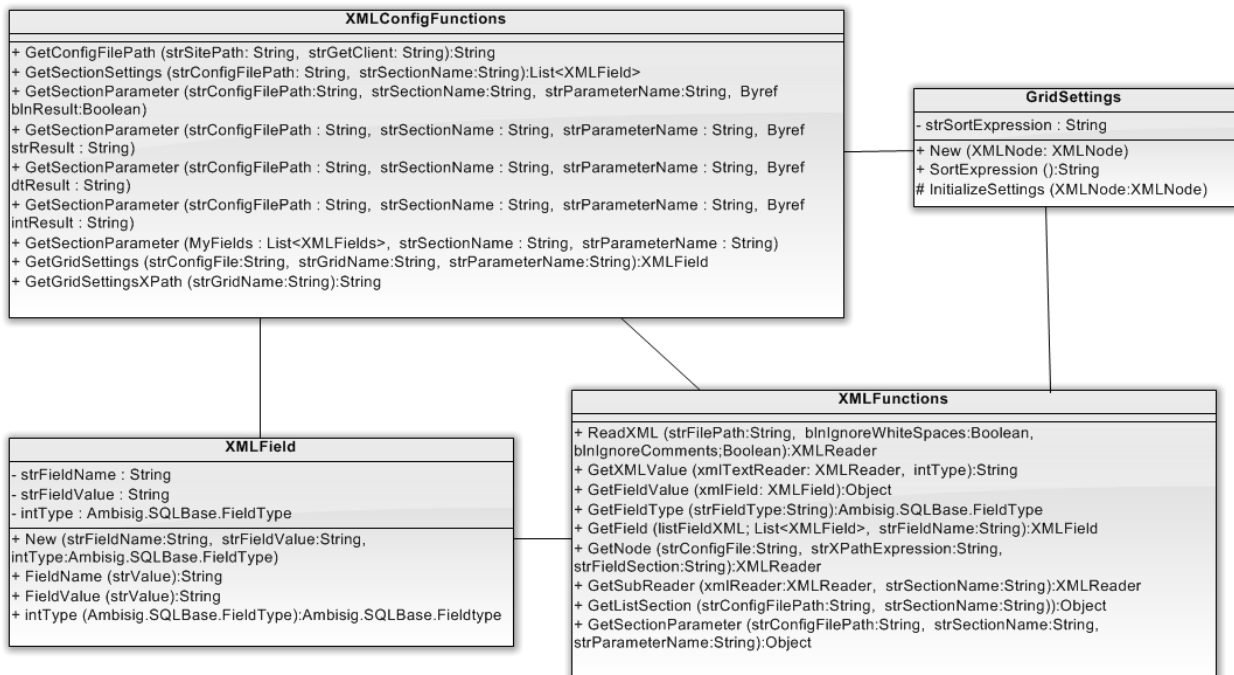


Figura 27 - packages “XML”

A *package* XML (Figura 27) é um conjunto de classes pertencentes ao WebDocLibrary que permite aceder aos dados de um XML. O uso do método

“GetSectionParameters” permite aceder aos parâmetros de configuração dos módulos existentes no ficheiro de configuração da aplicação.

As vantagens desta funcionalidade/metodologia são bastantes para a criação de um produto como o WebDoc2010 que é “partilhado” por vários clientes, cada um com as suas visões para o seu software. O facto de existirem várias configurações, uma para cada cliente, permite que o programador não tenha de configurar os dados para cada um quando recolhe as últimas versões do TFS. Isto permite que o programador não tenha de estar ciente de todos os parâmetros de configuração de cada cliente. Existe uma melhor organização dos ficheiros e configurações dos clientes pois estes encontram-se sempre na pasta específica do cliente. Ainda como vantagens esta metodologia permite que o WebDoc2010 possa ter mais do que um layout, uma redução das keys no ficheiro de configuração da aplicação e as classes de acesso aos ficheiros de configuração, nomeadamente a package XML pode ser reutilizada para outros propósitos.

Um grande problema encontrado foi a desorganização das imagens e ficheiros necessários, estes encontravam-se mal distribuídos pelo código da aplicação sendo necessário a identificação e reestruturação desses ficheiros. Muitos desses ficheiros encontravam-se repetidos no código, principalmente imagens.

Esta funcionalidade foi bastante simples de desenvolver mas tornou o tempo de desenvolvimento bastante mais curto do que era habitual.

#### **4.4. Estruturação de classes de negócio**

Como já foi anteriormente descrito, uma das grandes desvantagens no desenvolvimento de soluções anteriores de gestão documental na AMBISIG era o facto de não haver a metodologia de um conjunto de classes de negócio, que permitiria ao programador melhorar o seu tempo e qualidade de desenvolvimento.

O gestor do projecto, em discussão com a restante equipa de desenvolvimento, achou por bem a criação dessa camada de negócio, embora este requisito não estivesse planeado no contracto de trabalho com o cliente.

Para desenvolver esta camada de negócio foi então criado um projecto pertencente à solução do WebDoc2010 designado de WebDocLibrary.

O desenvolvimento desta funcionalidade teve as seguintes fases: criação do projecto WebDocLibrary na solução do WebDoc2010, criação de vários *namespaces* (mapeiam a estrutura física representada na figura 28), revisão do código do projecto WebDoc2010 para identificar os objectos a serem criados no WebDocLibrary, a criação de classes para acesso a dados e lógica de negócio, modificação do código do projecto WebDoc2010 para usar as novas classes e para finalizar a execução de testes.

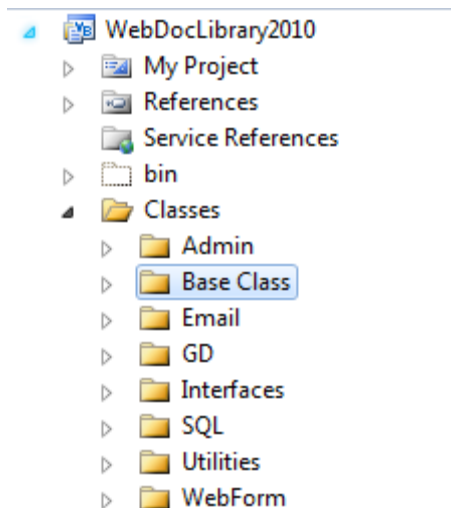


Figura 28 - Estrutura física WebDocLibrary

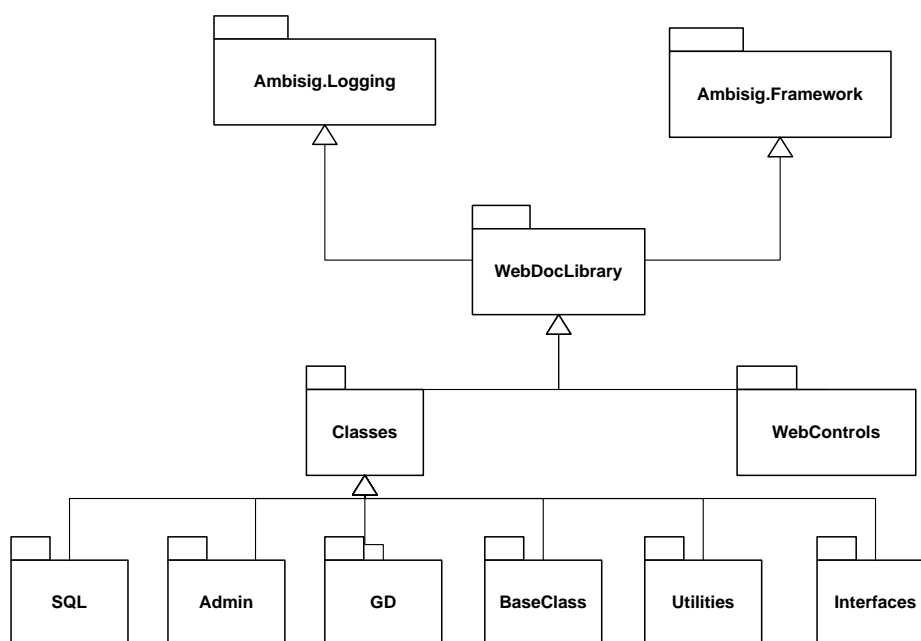


Figura 29 - Modelo de classes das *packages* do WebDocLibrary

A figura 29 mostra o modelo de classes do projecto WebDocLibrary (“biblioteca”) que contém todas as classes com a lógica de negócio e de recolha de dados da Base de Dados.

Como podemos ver a “biblioteca” de classes vai referenciar dois projectos internos da empresa, nomeadamente a Framework da Ambisig e o projecto produzido no âmbito deste projecto para registo de logs nas aplicações, ou seja, o Ambisig.Logging.

A package WebDocLibrary é ramificada em dois *packages*, o package “WebControls” é constituído por todas as classes que herdam da classe System.Web.UI.WebControl enquanto a package “Classes” engloba todos os outros tipos de classes.

A package “Classes” é também por sua vez ramificada em seis *packages*, a package “SQL” que contém todas as classes de acesso a dados, a package “Interfaces” que possui todas as interfaces utilizadas, a *package* “Utilities” que possui todas as classes que auxiliam o processamento, a *package* “BaseClass” que possui todas as classes abstractas utilizadas pela GD e as *packages* “Admin” e “GD” que possuem todos os objectos referentes ao modelo de negócio nos contextos de administração e gestão documental.

#### 4.4.1. BaseClass

Como referido anteriormente o *namespace* “BaseClass” possui todas as classes abstractas que representam uma tabela da base de dados. As classes abstractas têm a seguinte estrutura (Figura 30):

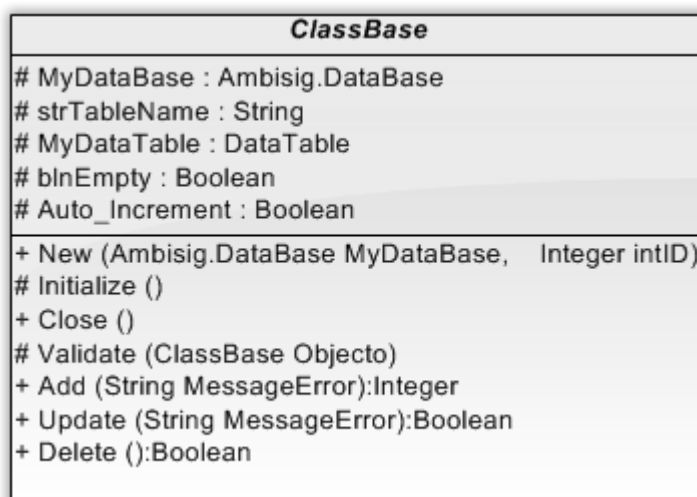


Figura 30 - Estrutura de uma BaseClass

Uma metodologia implementada é a validação dos dados do objecto com as regras de meta dados, estes encontram-se guardados em duas tabelas “nrField” e “nrTable”. Estas tabelas já existentes na metodologia de desenvolvimento da AMBISG desde a sua existência, representam as colunas de uma tabela e as tabelas respectivamente.

A cada registo da tabela “nrTable” corresponde uma tabela e por sua vez a cada registo da tabela “nrField” corresponde uma coluna dessa tabela. Na tabela “nrField” é possível identificar qual o tipo de dados dessa coluna, se é *primary key*, se é obrigatório, se é *foreign key* e visível entre muitas outras características.

A validação é feita verificando cada propriedade do objecto, que por sua vez tem o mesmo nome da coluna associada, com as características do campo do registo pertencente à tabela “nrField” correspondente. Desta maneira conseguimos validar se o objecto possui todos os campos obrigatórios, se a *foreign key* é válida e quais os valores base, permitindo posteriormente criar automaticamente as *quers* de inserção ou actualização do registo correspondente.

Tabela 4 - Métodos de uma “BaseClass”

Método	Descrição
<b>New</b>	Construtor da classe.
<b>Initialize</b>	Inicializa as variáveis do objecto. Se ID do objecto maior que 0 então recolhe o registo da base de dados e actualiza as variáveis, caso contrário, inicializa as variáveis com os valores base.
<b>Close</b>	Liberta as variáveis de memória.
<b>Validate</b>	Valida as variáveis para a inserção ou actualização do registo.
<b>Add</b>	Adiciona um novo registo com os dados do objecto actuais.
<b>Update</b>	Actualiza na base de dados o registo com os dados actuais.
<b>Delete</b>	Remove o registo actual da base de dados.

#### 4.4.2. SQL

As classes pertencentes a este namespace são classes estáticas que possuem única e exclusivamente métodos de acesso a dados. Cada classe possui métodos responsáveis pela modelação de uma tabela da base de dados.

Com o recurso das classes pertencentes ao namespace SQL ganha-se assim uma maior organização do código, estando a maioria da lógica de negócio toda incorporada no *namespace* “BaseClass”, enquanto a lógica de acesso a dados se encontra no namespace SQL.

A criação desta estrutura teve vantagens muito significativas para o desenvolvimento e qualidade do projecto, pois permitiu uma melhor organização e estruturação do código, uma identificação de métodos replicados que só ocupavam espaço e obrigavam á alteração do código em vários locais sempre que era preciso efectuar melhoramentos a esses métodos. Como o uso destes métodos é maior a identificação de erros é mais rápida e como tal aumenta a qualidade dos métodos (depois de corrigidos esses erros). Outra grande vantagem é que a camada de negócio da GD é mais facilmente transposta para outra solução.

## Capítulo 5

### Conclusões

Neste capítulo serão apresentadas as conclusões do trabalho, quer referentes ao projecto realizado quer aos objectivos deste. Serão também identificadas possíveis melhorias para as funcionalidades realizadas.

O projecto realizado encontra-se actualmente em melhorias e correcções, está já em funcionamento em mais dois clientes IGFSE - Instituto Geral Fundo Social Europeu e na Câmara Municipal de Arganil. Dada a boa adesão de clientes e possíveis interessados é seguro dizer que o produto WebDoc2010 está a cumprir os seus objectivos embora do meu ponto de vista possa ainda ser muito melhorado. O facto de não ter sido criado um planeamento específico para a criação do produto, (o planeamento foi inicialmente previsto para uma solução de gestão documental com base única e exclusivamente da solução da DREC), foi a causa de um desenvolvimento inicial lento, pois algumas definições base foram bastante debatidas entre mim e o gestor do projecto antes de inicializar qualquer nova funcionalidade. Outra falha que implicou alterações ao ritmo de desenvolvimento foi a recolha incorrecta dos requisitos (inexistente em alguns casos), esta falha é um dos grandes problemas que ocorre no desenvolvimento das aplicações informáticas. O cliente quase sempre não sabe o que quer o que obriga a alterações numa fase mais avançada do desenvolvimento.

Para finalizar, o maior problema para uma empresa menor como AMBISIG é o facto de existirem poucos recursos, financeiros e humanos, que origina uma recolocação do recurso quando existem projectos em atraso, ou seja, quando um projecto se encontra em atraso muitas são as vezes em que é necessário um acréscimo de programadores, sendo preciso retirar programadores de outros projectos, atrasando assim o seu desenvolvimento. Este problema faz com que o tempo para desenvolvimento do projecto decresça sendo necessário por vezes reduzir a qualidade do código ou até mesmo adiar o desenvolvimento de algumas funcionalidades.

As possíveis melhorias identificadas para cada funcionalidade são as seguintes:

## **5.1. Envio e Recepção de Faxes**

Nesta funcionalidade era útil a criação de um histórico dos faxes recebidos (data e utilizador), cada fax recebido é listado no contexto de faxes recebidos mas se um utilizador apagar um fax este não fica registado, assim como os faxes que são registados na gestão documental.

Cada vez que um utilizador entra na listagem, são percorridos todos os faxes enviados que ainda não se encontram num estado arquivado. Isto origina um peso algo desnecessário na aplicação e pode ser uma melhoria na rapidez da funcionalidade se porventura fosse usado um serviço externo para esta sincronização.

## **5.2. Ambisig.Logging**

A ferramenta encontra-se actualmente disponível para todos os programadores da AMBISIG, após formação conduzida por mim. Próximos passos vão passar por implementar o sistema de logs na Framework da AMBISIG, devido a prioridades de tempo e falta de testes sérios de carga a este projecto, este ainda não se encontra integrado na Framework.

## **5.3. Estruturação de classes de negócio**

O facto de esta funcionalidade não ter sido planeada como requisito base do projecto, o seu planeamento e estruturação foi mais difícil de desenvolver e retirou o tempo de desenvolvimento para os restantes requisitos.

As metodologias de desenvolvimentos anteriores que faziam uso de classes da Framework da Ambisig constituíram um grande entrave, nalguns casos ainda não foi possível modificar o código para esta nova metodologia assim como não foi possível criar todos os objectos existentes na base de dados devido à falta de tempo resultante.

Para concluir, na minha óptica o projecto encontra-se bastante bem produzido face ao tempo de desenvolvimento e recursos utilizados. É certo que ainda existem algumas lacunas que estão a ser colmatadas mas de um modo geral o projecto é adequado aos requisitos dos clientes e pode vir a ser um produto com futuro no mercado da AMBISIG.

## Capítulo 6

### Referências

ASP.NET CONTROLS, AJAX UI ENABLED COMPONENTS, Telerik, 01 de Setembro de 2010, <http://www.telerik.com/products/aspnet-ajax.aspx>

COMPONENT PACKS FOR .NET JAVA JASPERREPORTS SHAREPOINT & SSRS, Aspose, 01 de Setembro de 2010, <http://www.aspose.com/categories/product-family-packs/aspose.total-product-family/default.aspx>

ABCPDF - Free PDF COMPONENT LIBRARY FOR DYNAMIC ASP VB AND .NET, WebSupergoo Software, 01 de Setembro de 2010, <http://www.websupergoo.com/abcpdf-1.htm>

APACHE LOG4NET - APACHE LOG4NET MANUAL: CONFIGURATION, Última actualização a 30 de Agosto de 2007, Apache Software Foundation, 01 de Setembro de 2010, <http://logging.apache.org/log4net/release/manual/configuration.html>

GESTÃO DOCUMENTAL - WIKIPÉDIA, A ENCICLOPÉDIA LIVRE, Última actualização a 13 de Novembro de 2009, Wikipédia, 01 de Setembro de 2010, [http://pt.wikipedia.org/wiki/Gest%C3%A3o\\_documental](http://pt.wikipedia.org/wiki/Gest%C3%A3o_documental)

CMMI - WIKIPÉDIA, A ENCICLOPÉDIA LIVRE, Última actualização a 13 de Agosto de 2010, Wikipédia, 01 de Setembro de 2010, <http://pt.wikipedia.org/wiki/CMMI>

Joana Maria Pereira, 19-08-2008. POL.01.VV - Política da Segurança de Informação. Documento para uso interno.

Sílvia Rodrigues, 17-10-2008. POL.02.02 - Políticas Detalhadas da Segurança de Informação. Documento para uso interno.

Ana Gomes, 29-06-2010. Manual do Sistema de Gestão Integrado (SGI). Documento público. pp 04 - 11



## **Anexo A - Mapa de Gantt**

ID	WBS	Task Name	Work	Start	Finish	Predecessors	Resource Names
1		<b>Projecto Implementação Gestão Documental</b>	<b>1,689.5 hrs</b>	<b>Mon 05/10/09</b>	<b>Thu 27/05/10</b>		
2	1.1	<b>Gestão de Projecto</b>	<b>1,689.5 hrs</b>	<b>Mon 05/10/09</b>	<b>Thu 27/05/10</b>		
3	1.1.1	<b>Gestão de Projecto e Controlo de Qualidade</b>	<b>94.5 hrs</b>	<b>Wed 07/10/09</b>	<b>Thu 27/05/10</b>		
4	1.1.1.1	Actividades Administrativas de Gestão de Projecto	80 hrs	Mon 23/11/09	Tue 02/03/10	35SS	
5	1.1.1.2	<b>Gestão da Comunicação</b>	<b>14.5 hrs</b>	<b>Thu 29/10/09</b>	<b>Tue 23/02/10</b>		
6	1.1.1.2.1	<b>Agenda de Reunião</b>	<b>2.5 hrs</b>	<b>Thu 29/10/09</b>	<b>Tue 23/02/10</b>		
12	1.1.1.2.2	<b>Reuniões</b>	<b>8 hrs</b>	<b>Mon 02/11/09</b>	<b>Tue 26/01/10</b>		
13	1.1.1.2.2.1	<b>Reuniões de Acompanhamento de Projecto</b>	<b>8 hrs</b>	<b>Mon 02/11/09</b>	<b>Tue 26/01/10</b>		
14	1.1.1.2.2.1.1	Reunião de Acompanhamento #1	2 hrs	Mon 02/11/09	Mon 02/11/09	35FS+1 mon	Gestor de Projecto
15	1.1.1.2.2.1.2	Reunião de Acompanhamento #2	2 hrs	Mon 30/11/09	Mon 30/11/09	14FS+1 mon	Gestor de Projecto
16	1.1.1.2.2.1.3	Reunião de Acompanhamento #3	2 hrs	Tue 29/12/09	Tue 29/12/09	15FS+1 mon	Gestor de Projecto
17	1.1.1.2.2.1.4	Reunião de Acompanhamento #4	2 hrs	Tue 26/01/10	Tue 26/01/10	16FS+1 mon	Gestor de Projecto
18	1.1.1.2.3	<b>Comunicação de Progresso</b>	<b>2 hrs</b>	<b>Fri 30/10/09</b>	<b>Mon 25/01/10</b>		
19	1.1.1.2.3.1	Relatório de Progresso #1	0.5 hrs	Fri 30/10/09	Fri 30/10/09	14SS-1 day	Gestor de Projecto
20	1.1.1.2.3.2	Relatório de Progresso #2	0.5 hrs	Fri 27/11/09	Fri 27/11/09	15SS-1 day	Gestor de Projecto
21	1.1.1.2.3.3	Relatório de Progresso #3	0.5 hrs	Mon 29/12/09	Mon 29/12/09	16SS-1 day	Gestor de Projecto
22	1.1.1.2.3.4	Relatório de Progresso #4	0.5 hrs	Mon 29/01/10	Mon 29/01/10	17SS-1 day	Gestor de Projecto
23	1.1.1.2.4	<b>Actas de Reunião</b>	<b>2 hrs</b>	<b>Tue 03/11/09</b>	<b>Wed 27/01/10</b>		
24	1.1.1.2.4.1	Acta de Reunião #1	0.5 hrs	Tue 03/11/09	Tue 03/11/09	14FS+1 day	Gestor de Projecto
25	1.1.1.2.4.2	Acta de Reunião #2	0.5 hrs	Wed 02/12/09	Wed 02/12/09	15FS+1 day	Gestor de Projecto
26	1.1.1.2.4.3	Acta de Reunião #3	0.5 hrs	Wed 30/12/09	Wed 30/12/09	16FS+1 day	Gestor de Projecto
27	1.1.1.2.4.4	Acta de Reunião #4	0.5 hrs	Wed 27/01/10	Wed 27/01/10	17FS+1 day	Gestor de Projecto
28	1.1.1.3	<b>Gestão de Deliverables</b>	<b>0 hrs</b>	<b>Wed 07/10/09</b>	<b>Thu 27/05/10</b>	35	
34	1.1.2	<b>Actividades de Iniciação de Projecto</b>	<b>17 hrs</b>	<b>Mon 05/10/09</b>	<b>Wed 07/10/09</b>		
35	1.1.2.1	<b>Reunião de Arranque</b>	<b>3 hrs</b>	<b>Mon 05/10/09</b>	<b>Mon 05/10/09</b>		Gestor de Projecto
36	1.1.2.2	<b>Plano de Projecto</b>	<b>6 hrs</b>	<b>Mon 05/10/09</b>	<b>Tue 06/10/09</b>		
37	1.1.2.2.1	Elaboração de Plano Inicial	4 hrs	Mon 05/10/09	Mon 05/10/09	35	Gestor de Projecto
38	1.1.2.2.2	Revisão de Plano de Projecto	2 hrs	Tue 06/10/09	Tue 06/10/09	37	Gestor de Projecto
39	1.1.2.3	<b>Criação Inicial de Estrutura Técnica de Projecto</b>	<b>8 hrs</b>	<b>Tue 06/10/09</b>	<b>Wed 07/10/09</b>		
40	1.1.2.3.1	Estrutura Modular	3 hrs	Tue 06/10/09	Tue 06/10/09	38	Analista Programador 1
41	1.1.2.3.2	Parametrizações de Equipa de Projecto em Ferramentas de Desenvolvimento	5 hrs	Tue 06/10/09	Wed 07/10/09	40	Analista Programador 1
42	1.1.3	<b>Desenvolvimento Técnico de Projecto</b>	<b>1,400 hrs</b>	<b>Wed 07/10/09</b>	<b>Fri 21/05/10</b>		
43	1.1.3.1	<b>Levantamento e Análise de Requisitos</b>	<b>386 hrs</b>	<b>Wed 07/10/09</b>	<b>Thu 03/12/09</b>	94	
48	1.1.3.2	<b>Desenvolvimento e Parametrização</b>	<b>1,014 hrs</b>	<b>Thu 03/12/09</b>	<b>Fri 21/05/10</b>		
49	1.1.3.2.1	Importação e Parametrização Workflow's(55 Unidades)	160 hrs	Thu 03/12/09	Thu 31/12/09	46	Analista Programador 1
50	1.1.3.2.2	Desenvolvimento de Requisitos	550 hrs	Thu 31/12/09	Wed 07/04/10	49	
51	1.1.3.2.3	Desenho dos Formulários (20 Unidades)	260 hrs	Wed 07/04/10	Fri 21/05/10	50	Analista Programador 1
52	1.1.3.2.4	<b>Testes Finais</b>	<b>44 hrs</b>	<b>Thu 31/12/09</b>	<b>Thu 07/01/10</b>		
53	1.1.3.2.4.1	Testes Modulares	4 hrs	Thu 31/12/09	Thu 31/12/09	49	Analista Programador 1
54	1.1.3.2.4.2	Testes de Segurança	2 hrs	Fri 01/01/10	Fri 01/01/10	53	Analista Programador 1
55	1.1.3.2.4.3	Testes Funcionais	16 hrs	Fri 01/01/10	Tue 05/01/10	54	Tester 1
56	1.1.3.2.4.4	Testes de Carga e Desempenho	2 hrs	Tue 05/01/10	Tue 05/01/10	55	Tester 1
57	1.1.3.2.4.5	Elaboração de Relatório de Testes	4 hrs	Tue 05/01/10	Tue 05/01/10	56	Gestor de Projecto
58	1.1.3.2.4.6	Estabilização da Solução	16 hrs	Wed 06/01/10	Thu 07/01/10	57	Analista Programador 1
59	1.1.4	<b>Instalação da Solução em Produção</b>	<b>8 hrs</b>	<b>Mon 11/01/10</b>	<b>Mon 11/01/10</b>		
60	1.1.4.1	Verificação de condições de Infra-estruturas	1 hr	Mon 11/01/10	Mon 11/01/10	52FS+1 day	Analista Programador 1
61	1.1.4.2	Instalação	4 hrs	Mon 11/01/10	Mon 11/01/10	60	Analista Programador 1
62	1.1.4.3	Testes	2 hrs	Mon 11/01/10	Mon 11/01/10	61	Analista Programador 1
63	1.1.4.4	Parametrização local	1 hr	Mon 11/01/10	Mon 11/01/10	62	Analista Programador 1
64	1.1.5	<b>Formação e Apoio ao Arranque</b>	<b>166 hrs</b>	<b>Tue 02/02/10</b>	<b>Wed 26/05/10</b>		
65	1.1.5.1	<b>Formação</b>	<b>122 hrs</b>	<b>Tue 02/02/10</b>	<b>Wed 26/05/10</b>		
66	1.1.5.1.1	Preparação do Dossier de Formação	12 hrs	Fri 21/05/10	Tue 25/05/10	48	Formador 2
67	1.1.5.1.2	Treino de Equipa de Formação	2 hrs	Fri 21/05/10	Fri 21/05/10	48	Analista Programador 1
68	1.1.5.1.3	Elaboração de Manual de Administrador	12 hrs	Mon 24/05/10	Tue 25/05/10	67	Formador 2
69	1.1.5.1.4	Elaboração de Manual de Utilizador	12 hrs	Tue 25/05/10	Wed 26/05/10	68	Formador 2
70	1.1.5.1.5	<b>Sessões</b>	<b>84 hrs</b>	<b>Tue 02/02/10</b>	<b>Tue 16/02/10</b>		
71	1.1.5.1.5.1	<b>Formação Administração - 4utilizador</b>	<b>7 hrs</b>	<b>Tue 02/02/10</b>	<b>Tue 02/02/10</b>		
72	1.1.5.1.5.1.1	Sessão 1	7 hrs	Tue 02/02/10	Tue 02/02/10		Formador 2
73	1.1.5.1.5.2	<b>Formação para Chefias - 8 utilizadores</b>	<b>7 hrs</b>	<b>Wed 03/02/10</b>	<b>Wed 03/02/10</b>		
74	1.1.5.1.5.2.1	Sessão 1	7 hrs	Wed 03/02/10	Wed 03/02/10		Formador 2
75	1.1.5.1.5.3	<b>Formação de Key-users - 5 Formandos</b>	<b>7 hrs</b>	<b>Thu 04/02/10</b>	<b>Thu 04/02/10</b>		
76	1.1.5.1.5.3.1	Sessão 1	7 hrs	Thu 04/02/10	Thu 04/02/10		Formador 2
77	1.1.5.1.5.4	<b>Formação Utilizadores - 105 Formandos</b>	<b>63 hrs</b>	<b>Fri 05/02/10</b>	<b>Tue 16/02/10</b>		
78	1.1.5.1.5.4.1	Sessão 1	7 hrs	Fri 05/02/10	Fri 05/02/10	76	Formador 1
79	1.1.5.1.5.4.2	Sessão 2	7 hrs	Fri 05/02/10	Mon 08/02/10	78	Formador 1
80	1.1.5.1.5.4.3	Sessão 3	7 hrs	Mon 08/02/10	Tue 09/02/10	79	Formador 1
81	1.1.5.1.5.4.4	Sessão 4	7 hrs	Tue 09/02/10	Wed 10/02/10	80	Formador 1
82	1.1.5.1.5.4.5	Sessão 4	7 hrs	Wed 10/02/10	Thu 11/02/10	81	Formador 1
83	1.1.5.1.5.4.6	Sessão 5	7 hrs	Thu 11/02/10	Fri 12/02/10	82	Formador 1
84	1.1.5.1.5.4.7	Sessão 6	7 hrs	Fri 12/02/10	Mon 15/02/10	83	Formador 1
85	1.1.5.1.5.4.8	Sessão 7	7 hrs	Mon 15/02/10	Mon 15/02/10	84	Formador 1
86	1.1.5.1.5.4.9	Sessão 8	7 hrs	Tue 16/02/10	Tue 16/02/10	85	Formador 1
87	1.1.5.2	<b>Apoio ao Arranque</b>	<b>44 hrs</b>	<b>Tue 16/02/10</b>	<b>Wed 24/02/10</b>		
88	1.1.5.2.1	Passagem a Produção	4 hrs	Tue 16/02/10	Wed 17/02/10	86	Analista Programador 1
89	1.1.5.2.2	Acompanhamento de Utilizadores on job	40 hrs	Wed 17/02/10	Wed 24/02/10	88	Suporte 1
90	1.1.6	<b>Actividades de Encerramento</b>	<b>4 hrs</b>	<b>Wed 24/02/10</b>	<b>Wed 24/02/10</b>		
93	1.2	<b>Momentos de aceitação pelo Cliente</b>	<b>0 hrs</b>	<b>Tue 06/10/09</b>	<b>Wed 24/02/10</b>		

