

INTEGRAÇÃO INTRANET – PRO/ENGINEER WILDFIRE 3.0 VIA PRO/WEB.LINK

Maria João Alves Gamelas

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Arquiteturas, Sistemas e Redes**

Orientador: Prof. Doutor Paulo Gandra de Sousa

Co-orientador: Engenheiro Carlos Pinheiro

Júri:

Presidente:

Vogais:

...às minhas avós, aos meus pais, irmão, namorado e amigos

Resumo

O *software* de CAD Pro/ENGINEER Wildfire 3.0, desenvolvido pela PTC, permite ser programado usando várias APIs (Pro/Toolkit, J-Link, VB, Pro/Web.Link), estendendo as suas funcionalidades e permitindo integração com outros sistemas informáticos. A presente dissertação *Integração Intranet - Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link* discute e apresenta essencialmente o trabalho efetuado num estágio de 8 meses, realizado em 2011 no departamento de Sistemas de Informação da unidade de Aparelhagem de Média e Alta Tensão (AMT) da Efacec, que teve como principais objetivos o estudo da API Pro/Web.Link, uma API que suporta a linguagem JavaScript, e o desenvolvimento de duas aplicações *Web* usando essa API para permitir a integração entre a intranet AMT e o *software* de CAD Pro/ENGINEER Wildfire 3.0. As duas aplicações desenvolvidas foram: a aplicação ProPEditor, um formulário *Web* para a edição da legenda dos desenhos do Pro/ENGINEER, apresentando alguns automatismos de tradução e sugestão de campos; e a aplicação ProCatalog, um catálogo *Web* que permite navegar facilmente pela livraria de acessórios de fixação (parafusos, anilhas, etc.) do Pro/ENGINEER, escolher o acessório pretendido e incorporá-lo no conjunto. Após o período de estágio, e face às dificuldades enfrentadas durante o desenvolvimento das aplicações, foi feita uma análise mais detalhada à API Pro/Web.Link que permitiu documentar algumas das suas regras e propor um método de trabalho adequado para implementar uma determinada aplicação com esta ferramenta. Pretende-se com este trabalho simplificar e complementar a literatura existente, fornecendo um instrumento útil para os muitos profissionais que queiram conhecer e utilizar a API Pro/Web.Link.

Palavras-chave: API, CAD, Efacec, Pro/ENGINEER Wildfire 3.0, Pro/Web.Link, PTC.

Abstract

The CAD software Pro/ENGINEER Wildfire 3.0, developed by PTC, can be programmed using several APIs (Pro/Toolkit, J-Link, VB, Pro/Web.Link), extending its functionality and allowing integration with other systems. This dissertation entitled Integration Intranet - Pro/ENGINEER Wildfire 3.0 by Pro/Web.Link essentially discusses and presents the work done during an 8 month internship, in 2011, in the Information Systems Department of Medium and High Voltage Switchgear Unit of Efacec, which mainly aimed to study the API Pro/Web.Link, an API that supports JavaScript language, and to develop two Web applications using this API to allow integration between the Switchgear intranet and the CAD software Pro/ENGINEER Wildfire 3.0. The two applications developed were: the ProPEditor application, a web form for editing Pro/ENGINEER's drawing's labels, providing some automatic translation and suggestions for the fields; and, the ProCatalog application, a Web catalog that allows to navigate easily through a library of fastener accessories (screws, washers, and so forth) of Pro/ENGINEER, choose the desired accessory and incorporate it in the assembly. After the internship, and due to the difficulties faced during the development of these applications, a more detailed analysis was made of the API Pro/Web.Link that allowed to document some of its rules and to propose a suitable working method to implement an application using this tool. This work intends to simplify and complement the existing literature by providing a useful tool for many professionals who want to learn and use the API Pro / Web.Link.

Keywords: API, CAD, Efacec, Pro/ENGINEER Wildfire 3.0, Pro/Web.Link, PTC.

Agradecimentos

Ao *Prof. Paulo Sousa*, por me ter guiado ao longo da fase do estágio e do desenvolvimento da dissertação e pelas suas dicas, transmissão de conhecimentos, revisões e correções.

Ao *Carlos Pinheiro*, por me ter ajudado durante e após o Estágio. O seu suporte dentro da Efacec foi fundamental para me integrar na empresa, para me motivar e me empenhar na realização das minhas funções. Os conhecimentos que me transmitiu foram essenciais para o desenvolvimento deste trabalho.

À *Drª. Fátima Rodrigues*, por me ter ajudado com alguns problemas relativos ao meu ingresso no ISEP. Sem a sua ajuda não estava aqui e por isso muito Obrigada.

À *Efacec*, por ter sugerido o estágio e por me ter recebido tão bem nas suas instalações, e aos seus colaboradores que tiveram a testar as aplicações.

Aos colaboradores do departamento de SI *Luís Gomes*, Luís e Paulo que tiveram um papel importante no desempenho das minhas funções.

Ao meu namorado *Márcio*, pela paciência e pela compreensão que demonstrou ao longo deste tempo.

Aos meus amigos, um obrigado por se preocuparem comigo e um pedido de desculpas pelas vezes que rejeitei as saídas à noite.

Aos meus pais, Francisco Gamelas e Maria Alves, por apoiarem a minha formação e os meus objetivos, e ao meu irmão, Bruno Gamelas, pela sua amizade e apoio.

À minha avó Maria, por me ter acolhido na sua casa durante os dois anos de mestrado, por ter sido uma grande companhia e por me ter feito as vontades todas.

Aos meus *colegas do ISEP*, pelo companheirismo e pelo empenho nos trabalhos.

Índice

Resumo	v
Abstract	vii
Agradecimentos	ix
Índice	xi
Lista de Figuras	xv
Lista de Tabelas.....	xix
Acrónimos.....	xxi
Glossário	xxiii
1 Introdução	1
1.1 Introdução	1
1.2 Enquadramento	3
1.3 Objetivos.....	5
1.3.1 Organizar Dados dos Acessórios de Fixação no GlobalArt	5
1.3.2 Estudar Framework da Intranet AMT (SynergyNet).....	6
1.3.3 Selecionar API do Pro/ENGINEER	6
1.3.4 Estudar API Seleccionada (Pro/Web.Link).....	7
1.3.5 Implementar Aplicação ProPEditor.....	8
1.3.6 Implementar Aplicação ProCatalog	10
1.4 Plano de Trabalho	11
1.5 Método de Trabalho.....	13
1.6 Contribuições	15
1.7 Apresentação da Empresa	16
1.8 Estrutura da Dissertação	18
2 Intranet AMT - SynergyNet	21
2.1 Introdução	21
2.2 Arquitetura da SynergyNet	24
2.3 Framework da SynergyNet	25
2.3.1 Aplicações de Geração e Edição dos Portais dos Departamentos.....	29
2.4 GlobalArt.....	30
2.4.1 Organização dos Dados dos Acessórios de Fixação no GlobalArt.....	33

2.5	Sumário.....	39
3	PDS - Product Development System	41
3.1	Introdução	41
3.2	Pro/ENGINEER Wildfire 3.0.....	43
3.2.1	Tipos de Modelos	43
3.2.2	Características.....	45
3.2.3	APIs	46
3.3	Windchill	54
3.3.1	Windchill PDMLink 9.0 (Product Data Management)	55
3.4	Sumário.....	62
4	Pro/Web.Link.....	65
4.1	Introdução	65
4.2	Arquitetura da API Pro/Web.Link.....	66
4.3	Classes.....	67
4.3.1	Pro/ENGINEER - Related	68
4.3.2	Module - Level	68
4.3.3	Compact Data	69
4.3.4	Enumeration.....	69
4.3.5	Union	70
4.3.6	Sequence.....	71
4.3.7	Array.....	72
4.4	Herança	73
4.5	Pro/Web.Link API Wizard	74
4.6	Sumário.....	76
5	Integração Intranet - Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link.....	77
5.1	Introdução	77
5.2	Arquitetura	78
5.3	Configuração do Ambiente de Desenvolvimento/Execução.....	79
5.3.1.	Configuração das Definições de Segurança no Internet Explorer	79
5.3.2.	Registo da API Pro/Web.Link no Windows	80
5.3.3.	Ativação da Opção web_enable_javascript no ficheiro config.pro	81
5.4	Módulo JavaScript	81
5.5	Método de Trabalho Proposto	83
5.6	Aplicações Desenvolvidas	86
5.6.1	Tecnologias Utilizadas	87
5.6.2	Arquitetura	88
5.7	Sumário.....	90

6	ProPEditor	93
6.1	Contextualização	93
6.2	Modelação do Novo Sistema	100
6.2.1	Caso de Uso: Editar Legenda	101
6.3	Aplicação Final.....	102
6.4	Vantagens.....	104
6.5	Sumário	105
7	ProCatalog	107
7.1	Contextualização	107
7.2	Modelação do Novo Sistema	110
7.2.1	Caso de Uso: Adicionar Artigo.....	110
7.3	Aplicação Final.....	111
7.3.1	Primeira Página - Classes	111
7.3.2	Segunda Página - Normas	112
7.3.3	Terceira Página - Artigos.....	112
7.3.4	Funcionalidades	114
7.3.5	Quarta Página - Pesquisa Global.....	119
7.4	Vantagens.....	120
7.5	Sumário	121
8	MassProp	123
8.1	Contextualização	123
8.2	Aplicação Final.....	125
8.3	Sumário	127
9	Conclusões	129
9.1	Objetivos Cumpridos e Contributos	130
9.1.1.	Organizar Dados dos Acessórios de Fixação no GlobalArt	130
9.1.2.	Estudar Framework da Intranet AMT (SynergyNet).....	131
9.1.3.	Selecionar API do Pro/ENGINEER	132
9.1.4.	Estudar API Seleccionada (Pro/Web.Link).....	133
9.1.5.	Implementar Aplicação ProPEditor e Aplicação ProCatalog	134
9.2	Avaliação dos Objetivos	136
9.3	Limitações e Trabalho Futuro	138
	Referências	141
	Anexos	147
	Anexo 1 - Enunciado do ProPEditor	147

Anexo 2 - Enunciado do ProCatalog.....	149
Anexo 3 - Newsletter da Efacec AMT de Abril de 2011	151
Anexo 4 - Newsletter da Efacec AMT de Julho de 2011.....	153
Apêndices	
Apêndice A - Aplicação Geração dos Portais	
Apêndice B - Aplicação Edição dos Portais	
Apêndice C - Configuração do Ambiente Desenvolvimento e Deployment	
Apêndice D - Módulo JavaScript	
Apêndice E - Caso de Uso Editar Legenda	
Apêndice F - Caso de Uso Adicionar Artigo	
Apêndice G - ProPEditor: Classes Pro/Web.Link.....	
Apêndice H - ProCatalog: Classes Pro/Web.Link	
Apêndice I - MassProp: Classes Pro/Web.Link.....	

Lista de Figuras

Figura 1 - Legenda de um desenho no Pro/ENGINEER Wildfire 3.0	8
Figura 2 - Janela de edição dos parâmetros de um modelo	9
Figura 3 - Interface do ProCatalog proposta por Rui Marinho	10
Figura 4 - Calendário de objetivos gerais e específicos do estágio	11
Figura 5 - Processo de desenvolvimento das aplicações	14
Figura 6 - Mapa-mundo com 8 regiões de atividade do grupo Efacec [Efacec, 2012b]	17
Figura 7 - Barra de ferramentas da SynergyNet.....	22
Figura 8 - Página do ProCatalog traduzida para (a) Inglês, (b) Castelhana e (c) Português.....	23
Figura 9 - Arquitetura física da SynergyNet (<i>Three-Tier</i>)	24
Figura 10 - Header e menu da SynergyNet	26
Figura 11 - Diagrama de classes (Adaptação do diagrama de [Pinheiro C., 2011])	26
Figura 12 - URL para o módulo de criação de um novo artigo da página GlobalArt.....	27
Figura 13 - Classe CForm e classes hereditárias.....	28
Figura 14 - Fragmento do antigo portal do departamento de Controlo.....	30
Figura 15 - Plataforma GlobalArt	30
Figura 16 - Plataforma GlobalArt com dados de um artigo	32
Figura 17 - Dados de um acessório de fixação – Freio exportado do Windchill	35
Figura 18 - Características de um acessório de fixação - Freio	35
Figura 19 - Artigo caracterizado no GlobalArt	36
Figura 20 - Documento de classificação dos freios e <i>circlips</i>	37
Figura 21 - Página do GlobalArt para criação de um artigo no Baan	39
Figura 22 - Tipo de objetos do Pro/ENGINEER.....	44
Figura 23 - Impacto da modificação do comprimento de uma peça no <i>Assembly</i>	45
Figura 24 - <i>Model Tree</i> do Pro/ENGINEER	46
Figura 25 - Módulos PDMLink 9.0 e ProjectLink 9.0 no navegador embutido do Pro/E	55
Figura 26 - Identificação dos elementos da versão de um objeto [PTC, 2007a].....	56
Figura 27 - Arquitetura multicamada do Windchill PDMLink [VMware, 2008]	57
Figura 28 - Opção <i>Server Registry</i> do menu <i>Tools</i> do Pro/ENGINEER [PTC, 2007b]	59
Figura 29 - Fluxo da comunicação entre o Windchill e o Pro/ENGINEER [PTC, 2007b]	59
Figura 30 - Operações PDM e locais de armazenamento [PTC, 2009a].....	61
Figura 31 - Diagrama de classes Pro/Web.Link 1	74
Figura 32 - API Wizard da API Pro/Web.Link.....	75
Figura 33 - <i>Browser</i> interno - Pro/Web.Link - Pro/ENGINEER Wildfire 3.0.....	78
Figura 34 - Ações executadas pelo módulo JavaScript	82
Figura 35 - Arquitetura ProPEditor, ProCatalog, MassProp (adaptação [Alexandre L., 2011]) .	89
Figura 36 - Janela dos parâmetros de um modelo (repetição).....	94
Figura 37 - Atributos de um modelo apresentado no <i>commonsplace</i> do Windchill PDMLink...	95
Figura 38 - Janela da Family Table de uma anilha (adaptação de [PTC, 2009c])	96
Figura 39 - Legenda de um desenho no Pro/ENGINEER Wildfire 3.0 2.....	97
Figura 40 - Texto variável e estático de uma legenda.....	97

Figura 41 - Campos de texto variável	98
Figura 42 - Diagrama de casos de uso do ProPEditor.....	101
Figura 43 - Interface da aplicação ProPEditor 1	102
Figura 44 - Interface da aplicação ProPEditor 2	103
Figura 45 - Interface da aplicação ProPEditor 3	103
Figura 46 - Interface da aplicação ProPEditor 4	104
Figura 47 - <i>Model tree</i>	109
Figura 48 - Diagrama de casos de uso do ProCatalog	110
Figura 49 - Interface da aplicação ProCatalog 1.....	111
Figura 50 - Interface da aplicação ProCatalog 2.....	112
Figura 51 - Interface da aplicação ProCatalog 3.....	113
Figura 52 - Interface da aplicação ProCatalog 4.....	114
Figura 53 - Interface da aplicação ProCatalog 5.....	114
Figura 54 - Interface da aplicação WebBaan.....	115
Figura 55 - Interface da aplicação ProCatalog 6.....	115
Figura 56 - Interface da aplicação ProCatalog 7.....	116
Figura 57 - Interface da aplicação ProCatalog 8.....	116
Figura 58 - Interface da aplicação ProCatalog 9.....	117
Figura 59 - Interface da aplicação GlobalArt.....	117
Figura 60 - Interface da aplicação ProCatalog 10.....	118
Figura 61 - Interface da aplicação ProCatalog 11.....	118
Figura 62 - Aplicação ProCatalog: incorporação de um acessório no Pro/E Wildfire 3.0	119
Figura 63 - Interface da aplicação ProCatalog 12.....	119
Figura 64 - Identificação de um parafuso na classe de artigo das porcas.....	121
Figura 65 - Estrutura de um <i>assembly</i>	125
Figura 66 - Interface da aplicação MassProp (parte 1)	126
Figura 67 - Interface da aplicação MassProp (parte 2)	126
Figura 68 - Proposta de apresentação da interface do ProCatalog (repetição).....	149
Figura 69 - Newsletter da Efacec AMT de Abril de 2011.....	151
Figura 70 - Newsletter da Efacec AMT de Julho de 2011.....	153
Figura 71 - Antigo portal do departamento de Produção	
Figura 72 - Elemento <i>tab</i>	
Figura 73 - Elemento <i>link</i>	
Figura 74 - Elemento <i>general item</i>	
Figura 75 - Elemento <i>subtitle</i>	
Figura 76 - Classe <i>tbpage_parameters</i>	
Figura 77 - Dados do departamento de Produção (parte 1)	
Figura 78 - Dados do departamento de Produção (parte 2)	
Figura 79 - Novo portal do departamento de Produção	
Figura 80 - Novo portal de informática	
Figura 81 - Aplicação de edição dos portais dos departamentos 1	
Figura 82 - Aplicação de edição dos portais dos departamentos 2	
Figura 83 - Aplicação de edição dos portais dos departamentos 3	

Figura 84 - Aplicação de edição dos portais dos departamentos 4	
Figura 85 - ProPEditor: Diagrama de classes Pro/Web.Link 1.....	
Figura 86 - ProPEditor: Diagrama de classes Pro/Web.Link 2.....	
Figura 87 - ProPEditor: Diagrama de classes Pro/Web.Link 3.....	
Figura 88 - ProPEditor: Diagrama de classes Pro/Web.Link 4.....	
Figura 89 - ProPEditor: Diagrama de classes Pro/Web.Link 5.....	
Figura 90 - ProCatalog: Diagrama de classes Pro/Web.Link 1	
Figura 91 - ProCatalog: Diagrama de classes Pro/Web.Link 2	
Figura 92 - MassProp: Diagrama de classes Pro/Web.Link 1	
Figura 93 - MassProp: Diagrama de classes Pro/Web.Link 2	

Lista de Tabelas

Tabela 1 - Características globais dos Acessórios de Fixação	34
Tabela 2 – Características das APIs J-Link, VB e Pro/Web.Link	48
Tabela 3 - Avaliação do objetivo <i>Organizar Dados dos Acessórios de Fixação</i>	136
Tabela 4 - Avaliação do objetivo <i>Estudar Framework da Intranet AMT (SynergyNet)</i>	137
Tabela 5 - Avaliação do objetivo <i>Selecionar API do Pro/ENGINEER</i>	137
Tabela 6 - Avaliação do objetivo <i>Estudar API Selecionada (Pro/Web.Link)</i>	138
Tabela 7 - Avaliação dos objetivos <i>Implementar Aplicação ProPEditor e ProCatalog</i>	138
Tabela 8 - Parâmetros do ProPEditor	147

Acrónimos

2D	Duas Dimensões
3D	Três Dimensões
ADO	<i>ActiveX Data Object</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
AMT	Aparelhagem de Média e Alta Tensão
API	<i>Application Programming Interface</i>
CAD	<i>Computer Aided Design</i>
CAE	<i>Computer Aided Engineering</i>
CAID	<i>Computer Aided Industrial Design</i>
CAM	<i>Computer Aided Manufacturing</i>
COM	<i>Component Object Model</i>
DIN	<i>Deutsches Institut für Normung</i>
DLL	<i>Dynamic-Link Library</i>
ERP	<i>Enterprise Resource Planning</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
I&D	Investigação e Desenvolvimento
IE	Internet Explorer
ISO	<i>International Standards Organization</i>
J2EE	<i>Java 2 Platform Enterprise Edition</i>

JAR	<i>Java Archive</i>
JDBC	<i>Java Database Connectivity</i>
JFC	<i>Java Foundation Classes</i>
JNI	<i>Java Native Interface</i>
JSP	<i>Java Server Pages</i>
PDA	<i>Personal Digital Assistant</i>
PDM	<i>Product Data Management</i>
PDS	<i>Product Development System</i>
PFC	<i>Parametric Foundation Classes</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
PLM	<i>Product Lifecycle Management</i>
Pro/E	<i>Pro/ENGINEER</i>
PTC	<i>Parametric Technology Corporation</i>
RMI	<i>Remote Method Invocation</i>
RPC	<i>Remote Procedure Calls</i>
SI	<i>Sistemas de Informação</i>
SOAP	<i>Simple Object Access Protocol</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>

Glossário

Acessório de Fixação - Acessório que une dois ou mais objetos. Exemplos: porcas, parafusos, anilhas, rebites, cavilhas, golpilhas, *circlips*, freios.

Artigo - Acessório ou produto comprado ou fabricado pela empresa (parafuso, transformador, termóstato). Cada artigo é representado por um modelo do Pro/ENGINEER.).

Assembly - Modelo composto por várias peças, isto é, vários modelos do tipo *part*. Exemplo: uma anilha encaixada num parafuso.

Automation - Tecnologia baseada no COM (*Component Object Model*) que permite que as aplicações exponham os seus objetos a outras aplicações e que estas manipulem esses objetos.

Check-in (dar entrada) - Opção do Windchill para dar entrada do modelo permitindo que os restantes utilizadores possam solicitar a sua edição. Esta opção apenas está disponível se o modelo estiver *checked out*.

Check-out (dar saída) - Opção do Windchill que permite a edição exclusiva de um modelo no Pro/E por parte do utilizador que seleciona essa opção, bloqueando o modelo e fornecendo apenas acessos de leitura aos restantes utilizadores. Esta opção apenas está disponível se o modelo estiver no Windchill com o estado "*Product Change*".

COM (Component Object Model) – Sistema independente da plataforma de utilização e orientado a objetos que permite desenvolver objetos (componentes binários) que podem interagir com outros objetos.

COM Client (cliente COM) – Código ou objeto que obtém um apontador para um servidor *COM* e utiliza os seus serviços através da invocação dos métodos das suas interfaces. O cliente *Automation* é um tipo de cliente *COM*.

COM Server (servidor COM) - Objeto que fornece serviços aos clientes *COM*, disponibilizando os seus objetos *COM*.

Drawing (Desenho) - É um modelo 2D usado para representar um modelo 3D (*part* ou *assembly*).

Family Table - A *family table* de um modelo genérico é uma tabela cujos membros são todos os seus modelos instância associados.

Legenda - A legenda é constituída por campos, de texto variável ou estático, que contêm informação acerca do modelo 3D representado no desenho. Os campos de texto variável podem estar associados a parâmetros que são criados pelo utilizador para esse modelo 3D.

Modelo Genérico - Modelo padrão. Exemplo: parafuso de diâmetro 10 mm.

Modelo Instância - Modelo criado a partir da *family table* de um modelo genérico com o objetivo de se assemelhar a este. Exemplo: parafuso de diâmetro 5 mm (instância) criado a partir do parafuso de diâmetro 10 mm (genérico).

Modelo/Objeto - Um modelo/objeto pode ser um documento do tipo “*part*” (.prt), “*assembly*” (.asm), “*drawing*” (.drw).

Part (Peça) - Modelo que representa uma única peça. Exemplo: anilha.

Pro/ENGINEER (Pro/E) - Programa de CAD que permite a criação de modelos em 2D e 3D.

Regeneração - Atualização de um objeto para que este reflita as alterações que o utilizador efetuou ao mesmo.

Sólido - O sólido pode ser um modelo do tipo “*part*” (.prt) ou “*assembly*” (.asm).

User Parameter - Campo criado pelo utilizador para um modelo com o objetivo de armazenar determinada informação. Exemplo: parâmetro *DESENHADOR* com o valor *Maria João*.

Windchill - Programa que gere o ciclo de vida de um produto. Quando integrado com o Pro/ENGINEER, armazena todos os produtos aí desenvolvidos e controla as permissões de edição e leitura, entre outras funcionalidades.

1 Introdução

1.1 Introdução

Em 2008, a Efacec¹ iniciou o programa “Piloto Colombo Efacec” para “incentivar a geração e implementação de ideias com vista a fomentar os aumentos de produtividade” [Efacec, 2012a]. No decorrer desta nova política o colaborador Rui Marinho, projetista do departamento de Inovação e Desenvolvimento da Efacec, sugeriu o desenvolvimento de duas aplicações *Web* para integrar a intranet da unidade de Aparelhagem de Média e Alta Tensão (AMT) da Efacec com o principal sistema de CAD utilizado na empresa, o Pro/ENGINEER Wildfire 3.0 (Pro/E), desenvolvido pela PTC² (*Parametric Technology Corporation*): (1) uma aplicação para o preenchimento assistido da legenda dos desenhos do Pro/E; e (2) um catálogo para automatizar a montagem de acessórios no Pro/E.

A Efacec aprovou a ideia por duas razões: exequibilidade da mesma, uma vez que a PTC fornece algumas APIs (*Application Programming Interfaces*) que disponibilizam um conjunto de funções do Pro/E, permitindo integrá-lo com outros sistemas informáticos; obter ganhos de produtividade no processo de desenvolvimento de produtos. Após esta aprovação, o departamento de Sistemas de Informação (SI) da unidade AMT ficou encarregue de desenvolver as aplicações do programa Colombo. Estas aplicações têm como destinatários as equipas dos departamentos de Engenharia de Produto, Engenharia Industrial, e de I&D (Inovação e Desenvolvimento), existentes em Portugal, Argentina e Índia.

¹ Efacec, <http://www.efacec.pt>

² PTC, Product & Service Advantage, <http://www.ptc.com>

1 Introdução

O departamento de SI concordou que as aplicações solicitadas devem ser aplicações *Web* porque, entre outras razões, o processo de distribuição, instalação e atualização é mais eficiente, havendo a possibilidade de integrá-las na intranet da unidade AMT e de utilizar alguns dos seus mecanismos.

Neste contexto, surge a proposta de estágio com duração de 8 meses. Embora o autor não possuísse conhecimentos na área de CAD, decidiu envolver-se neste projeto aceitando o desafio de explorar uma das APIs do Pro/ENGINEER e de implementar as aplicações do programa Colombo para contribuir para o crescimento da Efacec, uma empresa de elevado prestígio, com a introdução de uma diferente filosofia de trabalho. **Pretende-se fundamentalmente com este estágio:**

- **Selecionar a API do Pro/ENGINEER mais adequada para este projeto** com base na análise e avaliação das APIs J-Link, VB e Pro/Web.Link do Pro/ENGINEER Wildfire 3.0 e 4.0 e nos requisitos e funcionalidades das aplicações ProPEditor e ProCatalog. Apesar da API VB estar disponível apenas a partir da versão Wildfire 4.0, esta deve ser analisada uma vez que a Efacec encontra-se recetiva a efetuar a transição do Pro/ENGINEER Wildfire 3.0 para o Wildfire 4.0 se esta API for selecionada. Note-se que a decisão final relativamente à seleção da API será do departamento de SI da Efacec;
- **Estudar com maior detalhe a Pro/Web.Link**, a API sugerida pelo autor e aceite pelo departamento de Sistemas de Informação após o processo de avaliação e seleção realizado na alínea anterior, **para desenvolver as aplicações do programa Colombo;**
- **Implementar com essa API as duas aplicações propostas (ProPEditor - um formulário para o preenchimento automático das legendas dos desenhos construídos no Pro/E e ProCatalog - um catálogo para a localização e incorporação de acessórios de fixação num *assembly* do Pro/E)** para integrar a intranet da Efacec com o Pro/E de forma a aumentar a produtividade dos utilizadores na realização das tarefas de edição das legendas dos desenhos e de montagem de acessórios no Pro/E.

A aplicação ProPEditor (ideia n.º 366 do programa Colombo) deve possibilitar o preenchimento dos campos da legenda dos desenhos através de um formulário *Web*. Pretende-se essencialmente com esta nova aplicação fornecer alguns automatismos que diminuam o tempo de realização desta tarefa, tal como o preenchimento

1.2 Enquadramento

automático de alguns campos: nome do projetista, data de modificação do desenho, etc.

A aplicação ProCatalog (ideia n.º 348 do programa Colombo) consiste num catálogo de acessórios de fixação (parafusos, porcas, entre outros.) para auxiliar os utilizadores na incorporação de um acessório no seu modelo de CAD. Esta aplicação vem automatizar a integração entre o sistema de CAD e os sistemas de gestão de produtos (Windchill e GlobalArt), evitando dessa forma que o utilizador necessite de procurar manualmente o acessório desejado e de recordar o seu código para o encontrar no Pro/E. O objetivo do catálogo é encurtar esses passos e apresentar uma interface gráfica para que os utilizadores possam escolher os acessórios de forma mais intuitiva e importá-los para o Pro/E mais rapidamente. Este catálogo também deve estar apto para incluir outro tipo de acessórios.

Este estágio dá origem à presente dissertação **“Integração Intranet – Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link”**. Para além de **discutir e apresentar o trabalho efetuado pelo autor durante o estágio**, que permitiu integrar a intranet da Efacec com o Pro/ENGINEER Wildfire 3.0 utilizando a API Pro/Web.Link, **esta dissertação pretende aprofundar o estudo realizado à API Pro/Web.Link durante esta etapa, identificando e documentando as suas regras e propondo uma metodologia de trabalho adequada para implementar uma aplicação com esta ferramenta**, para auxiliar outros programadores a utilizar esta ferramenta e responder a uma necessidade atual uma vez que, após uma revisão bibliográfica, se constatou que existem profissionais interessados nesta API e que não compreendem alguns detalhes da documentação escrita pela PTC.

Pretende-se com esta dissertação simplificar, organizar e complementar a literatura existente, fornecendo um instrumento útil para os muitos profissionais que queiram conhecer e utilizar a API Pro/Web.Link.

1.2 Enquadramento

Esta dissertação decorre no âmbito de um estágio curricular de 8 meses (2 de Novembro de 2010 a 30 de Junho de 2011) realizado no departamento de SI da unidade de negócio da AMT da Efacec Energia, Equipamentos e Máquinas Eléctricas, S.A., uma das empresas do Grupo

1 Introdução

Efacec situada em Leça do Balio, freguesia pertencente ao concelho de Matosinhos e distrito do Porto. O departamento de SI é responsável pela programação de aplicações *Web*, que estão alojadas na intranet da AMT, para otimizar o trabalho dos colaboradores da unidade AMT e de outras unidades da Efacec.

A unidade AMT é especialista no desenvolvimento e fabricação de equipamentos e de soluções no domínio da Alta e Média Tensão. As equipas de Engenharia de Produto, Engenharia Industrial, e de I&D de Portugal, Argentina e Índia, são responsáveis pela criação e personalização desses produtos de média e alta tensão e para tal utilizam como sistema de CAD o Pro/ENGINEER Wildfire 3.0 da PTC em ambiente Windows, que está integrado com o Windchill, para realizar esses projetos. O Windchill é um sistema PLM (*Product Lifecycle Management*) que pode estar integrado com o Pro/E e fazer a gestão de todos os artigos que são criados no Pro/E (desenhos, modelos) armazenando-os numa base de dados [PTC, 2007b]. Entre os tipos de objetos mais usados no Pro/E destacam-se os modelos 3D (Três Dimensões) *part* (peça) e *assembly* (conjunto de peças) e os objetos 2D (Duas Dimensões) *drawing* (desenho). Os desenhos são usados para representar os modelos 3D através da apresentação das diferentes perspetivas/vistas desses modelos e da sua legenda.

Em conjunto com o Pro/ENGINEER Wildfire 3.0 a PTC disponibiliza três interfaces de programação de aplicações (APIs) para automatizar as tarefas dos utilizadores do Pro/E, permitindo a integração deste com outros sistemas informáticos: Pro/Toolkit, J-Link e Pro/Web.Link. As APIs e as respetivas linguagens de programação suportadas por cada uma delas são: a Pro/Toolkit (C++), J-Link (Java) e a Pro/Web.Link (JavaScript) [PTC, 2006a][PTC, 2006b][PTC, 2006b][PTC, 2006e]. Com a introdução do Pro/ENGINEER Wildfire 4.0 surgiu uma quarta API: VB, que suporta as linguagens Visual Basic, VBA, VBScript e outras linguagens que usam servidores COM (*Component Object Model*) como o JavaScript ou C# [PTC, 2012b]. Enquanto a API Pro/Toolkit requer licença, as restantes são gratuitas. Cada uma destas APIs é construída com base nas classes da Pro/Toolkit, sendo esta a mais completa em termos de funcionalidades [PTC, 2006a].

A intranet da unidade AMT, uma rede de comunicação/informação interna da Efacec, possui um *Framework PHP*, criado no departamento de SI, para permitir o desenvolvimento de aplicações *Web*, contendo um conjunto de classes com variáveis e métodos. Estas classes estão documentadas e permitem desenhar as interfaces com tabelas, formulários, etc.,

1.3 Objetivos

efetuar pedidos às bases de dados, traduzir conteúdos utilizando um mecanismo de traduções, entre outras funcionalidades.

Uma das plataformas que integra a intranet da unidade AMT é o GlobalArt, um sistema de gestão de artigos que foi desenvolvido para evitar a repetição de artigos e ultrapassar algumas limitações do ERP (*Enterprise Resource Planning*) da Efacec, o Baan, estando totalmente integrado com este [Efacec Switchgear, 2009].

1.3 Objetivos

Os objetivos gerais principais do estágio/dissertação são: **Selecionar API do Pro/ENGINEER; Estudar API selecionada (Pro/Web.Link); e Implementar as aplicações ProPEditor e ProCatalog.** Para a implementação das aplicações devem ser atingidos dois objetivos gerais adicionais: **Organizar dados dos acessórios de fixação no GlobalArt**, a fonte de informação do ProCatalog; **Estudar o Framework da intranet AMT**, para conhecer o *Framework* da SynergyNet que deve ser utilizado no desenvolvimento das aplicações ProPEditor e ProCatalog.

1.3.1 Organizar Dados dos Acessórios de Fixação no GlobalArt

A plataforma GlobalArt, aplicação *Web* que faz a gestão dos artigos da Efacec, como fonte de informação da aplicação ProCatalog deve estar devidamente preparada para conseguir responder aos seus pedidos e requisitos. A aplicação ProCatalog deve agrupar e permitir a filtragem dos artigos por determinadas características globais (características comum a todas as companhias), todavia, os artigos de fixação não possuem quaisquer características definidas no GlobalArt, estando estas implícitas unicamente no campo da descrição dos artigos. Para suprir essa lacuna é necessário efectuar as seguintes etapas:

1. Definir no GlobalArt as características globais para cada um dos tipos de acessórios de fixação (parafusos, anilhas, etc.) com o auxílio do livro das normas da Efacec;
2. Importar do Windchill o código e a descrição de cada um dos acessórios de fixação para inferir os valores das características a partir da sua descrição;
3. Por fim atualizar os dados dos acessórios de fixação através da migração dos valores das características, inferidas no passo anterior, para o GlobalArt.

1 Introdução

Para que os utilizadores consigam usufruir da aplicação ProCatalog encontrando facilmente um acessório, é importante que pelo menos os artigos de fixação do GlobalArt estejam corretamente classificados. Da mesma forma, para que estes possam classificar corretamente novos modelos criados no GlobalArt, será elaborado um documento de apoio com a classificação dos acessórios de fixação.

Os resultados deste objetivo são descritos na secção 2.4.1 Organização dos Dados dos Acessórios de Fixação no GlobalArt.

1.3.2 Estudar *Framework* da Intranet AMT (SynergyNet)

É imperativo que as aplicações do Pro/E estejam integradas na intranet da organização e utilizem o seu *Framework* para o desenho da interface das aplicações, para a ligação às bases de dados e para as traduções das mesmas para as línguas suportadas pela intranet AMT da Efacec (Português, Espanhol e Inglês). Neste contexto, o *Framework* é formado por um conjunto de classes, implementadas em linguagem de programação PHP pelo departamento de Sistemas de Informação da unidade AMT, que visam auxiliar o desenvolvimento de novos programas. Para estudar o *Framework* e ao mesmo tempo colocar os conhecimentos em prática, serão criadas duas aplicações em PHP para integrar a intranet: a primeira aplicação deve gerar automaticamente os portais dos vários departamentos e a segunda aplicação deve permitir a edição de informação destes mesmos portais. Estas aplicações foram definidas pela Efacec.

Este contacto inicial com o *Framework* da intranet AMT é essencial para aprender as bases e ultrapassar eventuais dificuldades, permitindo na fase de implementação das aplicações do projeto Colombo focar essencialmente nas funcionalidades da API Pro/Web.Link do Pro/E.

Os resultados deste objetivo são descritos na secção 2.3.1 Aplicações de Geração e Edição dos Portais dos Departamentos.

1.3.3 Selecionar API do Pro/ENGINEER

Para selecionar a API mais adequada para o desenvolvimento das aplicações do programa Colombo, é fundamental efetuar um breve estudo às três APIs do Pro/ENGINEER: J-Link, VB e

1.3 Objetivos

Pro/Web.Link. A API Pro/Toolkit não está incluída neste estudo uma vez que a sua licença é dispendiosa.

Pretende-se que este estudo seja feito de forma genérica para que o departamento de SI conheça algumas destas APIs, e que sejam resumidas algumas das suas características principais numa tabela para responder a algumas questões, como por exemplo: se as licenças são gratuitas; se as APIs suportam aplicações síncronas e/ou assíncronas e aplicações *Web* e/ou *standalone*; se estas suportam *browsers* externos ao Pro/ENGINEER.

Após o resumo das características das APIs, por parte do autor, pretende-se iniciar o processo de seleção da API nomeando os requisitos das aplicações ProPEditor e ProCatalog que estão relacionados com esta e identificando todas as funcionalidades que estas devem implementar e que estão relacionadas com o Pro/E (ex: obter o modelo genérico a partir de um modelo instância). Para a identificação destas funcionalidades do Pro/E pretende-se realizar uma análise às aplicações ProPEditor e ProCatalog, definindo os seus casos de uso e descrevendo-os de forma detalhada. Para auxiliar esta análise, o departamento de SI irá fornecer um enunciado com algumas especificações e funcionalidades funcionais e não funcionais.

Com base nos requisitos e funcionalidades das aplicações ProPEditor e ProCatalog deve ser selecionada a API mais adequada. Esta seleção deve ser sugerida ao departamento de SI da Efacec, cabendo a este decidir se é ou não uma boa opção.

Os resultados da análise às APIs e do processo de decisão são apresentados nas subsecções 3.2.3.1 Tabela Comparativa e 3.2.3.2 Processo de Seleção e Decisão Final.

1.3.4 Estudar API Selecionada (Pro/Web.Link)

Este objetivo consiste em compreender a API Pro/Web.Link, a API selecionada anteriormente, para implementar as aplicações do programa Colombo na Efacec e para complementar a documentação existente sobre essa API.

A realização de um estudo a esta API tem como principal finalidade entender quais os tipos de classes que a constituem, como se configura o ambiente de desenvolvimento e de *deployment* de uma aplicação, identificar regras e perceber outros conceitos relevantes para o desenvolvimento de aplicações *Web* para o Pro/ENGINEER utilizando esta ferramenta, e propor uma metodologias de trabalho adequada.

1 Introdução

Os resultados deste objetivo são descritos no capítulo 4 Pro/Web.Link e capítulo 5 Integração Intranet – Pro/ENGINEER via Pro/Web.Link.

1.3.5 Implementar Aplicação ProPEditor

A aplicação ProPEditor deve permitir que o preenchimento da legenda dos desenhos do Pro/E possa ser feito através de um formulário *Web*. A ideia desta aplicação surgiu fundamentalmente para fornecer alguns automatismos para um preenchimento mais eficiente.

Os desenhos apresentam as diferentes vistas de um modelo 3D e são constituídos por uma legenda com o seguinte formato:

Tolerâncias gerais: ISO 2768-m ou I.O, MT.80.2.01		General Tol	
This document is the sole property of EFMAET. As confidential, reproduction, conveyance or information to third parties and other unauthorized use are not permitted except in exact accordance with prior express permission.		Este documento é propriedade comunicada a terceiros.	
Scale Escala 1/1	Mat.: -	Prof.: -	-
Projected by Projectado por	J.F.Capelo	Used by Usado em:	32209152-01
Drawn by Desenhado por	J.F.Capelo		32209229-01
Date Data	2009-08-05		32209234-02
Approved by Aprovado por	L.Tovar		Quality Level Nível de Qual.: -
Date Data	2009-08-05	BACK TOP PLATE ASSEMBLED CHAPA SUPERIOR RETAGUARDA EQUIPADA	
Weight Peso (kg)	-	-	
Area Área (mm ²)	-	Normafix 24kV	

Figura 1 - Legenda de um desenho no Pro/ENGINEER Wildfire 3.0

Esta legenda é constituída por vários campos, como por exemplo: *Projected by/Projectado por*; *Drawn by/ Desenhado por*; *Aproved by/ Aprovado por*. Alguns destes campos estão relacionados com parâmetros do modelo que está associado ao desenho.

A alteração de todos os campos pode ser feita diretamente no desenho ou, apenas no caso dos campos que estão associados a parâmetros, através de uma opção do Pro/E para editar os parâmetros.

1.3 Objetivos

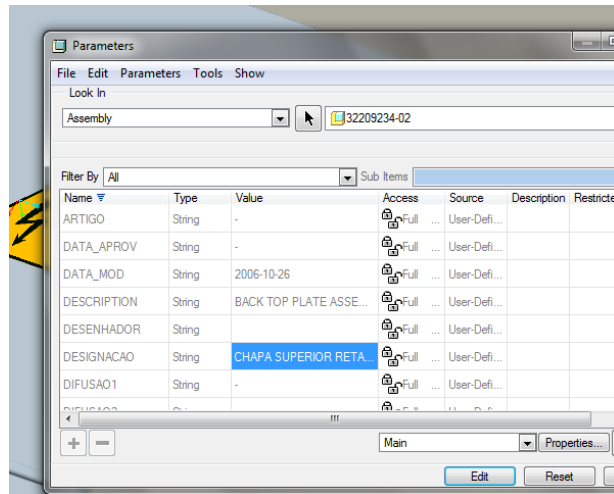


Figura 2 - Janela de edição dos parâmetros de um modelo

Inicialmente com a nova aplicação pretendia-se que o utilizador editasse apenas os campos que estão ligados aos parâmetros, contudo durante a fase de testes da aplicação determinou-se que o utilizador deve ser capaz de alterar todos os campos da legenda independentemente de estes estarem associados a parâmetros.

Quando a aplicação é invocada deseja-se que esta efetue a leitura do nome e dos valores de todos os campos e os apresente ao utilizador sob a forma de formulário. O utilizador pode preencher o formulário e após a modificação a aplicação registará os valores preenchidos.

Esta aplicação deve incluir automatismos, como por exemplo, na criação de um novo desenho a página deve preencher de forma automática os parâmetros *PROJETISTA* e *DESENHADOR* com o nome do utilizador que está a utilizar o Pro/E.

Os campos do formulário devem ser multilingue (Português, Espanhol e Inglês) usando o mecanismo de tradução da intranet AMT.

Os resultados deste objetivo são descritos no capítulo 6 ProPEditor.

1.3.6 Implementar Aplicação ProCatalog

A aplicação ProCatalog tem como objetivo facilitar a localização e a respetiva seleção de um acessório de fixação para posterior incorporação num modelo do tipo *assembly* (conjunto de peças) do Pro/E. De uma forma geral, um acessório de fixação é um componente físico usado para juntar dois ou mais objetos, podendo ser um parafuso, uma anilha, um rebite. A escolha deste tipo de acessórios para incorporar o catálogo tem como pretensão a satisfação de uma necessidade imediata visto que estes acessórios são os mais utilizados pelos engenheiros. Embora no âmbito desta dissertação a aplicação permita consultar as características dos acessórios de fixação, esta tem de estar preparada para incluir todo o tipo de acessórios que se pretenda catalogar numa fase futura.

O catálogo deve interagir com o GlobalArt para recolher informação sobre as famílias dos artigos e respetivas características. O GlobalArt é um sistema *Web* de gestão global de artigos, desenvolvido pelo departamento de Sistemas de Informação da Efacec AMT, para manter a coerência na criação e qualificação de artigos em várias empresas espalhadas pelo Mundo num ambiente multilingue. Cada artigo está associado a um objeto da ferramenta de *Product Lifecycle Management Windchill*.

Com esta aplicação deseja-se pesquisar, selecionar e adicionar artigos ao *assembly* do Pro/ENGINEER de uma forma mais rápida. À semelhança dos catálogos eletrónicos esta aplicação deve permitir, através de uma interface gráfica, localizar facilmente artigos de fixação das diversas companhias da Efacec.

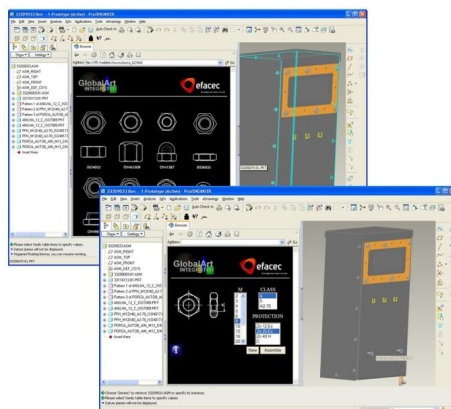


Figura 3 - Interface do ProCatalog proposta por Rui Marinho

1.4 Plano de Trabalho

Sugere-se que a aplicação apresente inicialmente uma lista de imagens em que cada uma corresponde a uma classe de artigo (classe das anilhas e placas oblíquas, classe dos parafusos, etc.). Após a seleção de uma classe todos os seus artigos são agrupados por uma determinada característica, como por exemplo, a norma. Após a escolha da característica é apresentada uma interface com as características desses artigos, permitindo localizar o artigo pretendido e adicionar o mesmo no Pro/E através de um clique num botão.

Os resultados deste objetivo são descritos no capítulo 7 ProCatalog.

1.4 Plano de Trabalho

	Task Name	Start	Finish
1	☐ Estágio Efacec	Tue 02-11-10	Thu 30-06-11
2	☐ Organizar Dados dos Acessórios de Fixação no GlobalArt	Tue 02-11-10	Tue 07-12-10
3	Definir características globais dos acessórios de fixação no GlobalArt	Tue 02-11-10	Mon 22-11-10
4	Tratar dados e acompanhar a sua migração para o GlobalArt	Tue 02-11-10	Mon 22-11-10
5	Elaborar documentos com a classificação dos acessórios de fixação	Tue 23-11-10	Tue 07-12-10
6	☐ Estudar Framework da Intranet AMT	Thu 09-12-10	Fri 14-01-11
7	Estudar Framework da intranet AMT e a linguagem PHP	Thu 09-12-10	Fri 14-01-11
8	Desenvolver aplicação de geração de portais dos departamentos	Thu 09-12-10	Thu 23-12-10
9	Desenvolver aplicação de edição de portais dos departamentos	Fri 24-12-10	Fri 14-01-11
10	☐ Selecionar API do Pro/E	Thu 13-01-11	Tue 15-02-11
11	Estudar API VB	Thu 13-01-11	Wed 19-01-11
12	Estudar API J-Link	Thu 20-01-11	Wed 26-01-11
13	Estudar API Pro/Web.Link	Thu 27-01-11	Wed 02-02-11
14	Comparar APIs	Thu 03-02-11	Mon 07-02-11
15	☐ Iniciar processo de seleção	Tue 08-02-11	Tue 15-02-11
16	Nomear requisitos das aplicações ProPEditor e ProCatalog	Tue 08-02-11	Tue 15-02-11
17	Identificar e descrever os casos de uso das aplicações ProPEditor e ProCatalog	Tue 08-02-11	Tue 15-02-11
18	Selecionar a API mais adequada	Tue 08-02-11	Tue 15-02-11
19	☐ Estudar API selecionada	Wed 16-02-11	Thu 30-06-11
20	Estudar funcionalidades/exemplos	Wed 16-02-11	Thu 30-06-11
21	☐ Implementar Aplicação ProPEditor	Wed 16-02-11	Fri 01-04-11
22	Desenvolver aplicação	Wed 16-02-11	Thu 31-03-11
23	Testar/Analisar aplicação	Wed 16-02-11	Thu 31-03-11
24	Elaborar manual do utilizador	Fri 01-04-11	Fri 01-04-11
25	☐ Implementar Aplicação MassProp	Mon 04-04-11	Tue 05-04-11
26	Analisar/ Desenvolver/ Testar aplicação	Mon 04-04-11	Tue 05-04-11
27	☐ Implementar Aplicação ProCatalog	Wed 06-04-11	Thu 30-06-11
28	Desenvolver aplicação	Wed 06-04-11	Wed 29-06-11
29	Testar/Analisar aplicação	Wed 06-04-11	Wed 29-06-11
30	Elaborar manual do utilizador	Thu 30-06-11	Thu 30-06-11

Figura 4 - Calendário de objetivos gerais e específicos do estágio

1 Introdução

Para garantir uma gestão eficiente do tempo de Estágio e mensurar da melhor forma os objetivos a atingir foi elaborado, em conjunto com o Coorientador Engenheiro Carlos Pinheiro e sob orientação do Doutor Paulo Sousa, um calendário de trabalho. O calendário apresentado na *Figura 4* representa o plano de trabalho final.

Este calendário de trabalho contém os vários objetivos gerais e específicos a cumprir durante o estágio, com início no dia 2 de Novembro de 2010 e fim no dia 30 de Junho de 2011. Durante este tempo houve cerca de 9 feriados que são tidos em conta neste calendário, havendo outros períodos de falta por razões académicas (trabalhos, exames) ou pessoais que não são aqui registados. As datas de início e fim de cada objetivo, representadas na figura anterior, representam datas aproximadas.

Objetivo ***Organizar os dados dos acessórios de fixação no GlobalArt***: definir as características globais de cada acessório de fixação no GlobalArt, capacitando esta plataforma para responder aos requisitos da aplicação ProCatalog; tratar os dados destes acessórios num documento Excel e acompanhar a migração desses dados para o GlobalArt; e elaborar documentos com a classificação de acessórios de fixação para auxiliar os utilizadores na criação dos acessórios de fixação na plataforma GlobalArt.

Objetivo ***Estudar Framework da Intranet AMT***: estudar o *Framework* da Efacec e a linguagem de programação PHP, colocando em prática os conhecimentos adquiridos através do desenvolvimento de duas aplicações para a intranet AMT.

Objetivo ***Selecionar API do Pro/ENGINEER***: breve estudo a três APIs do Pro/ENGINEER Wildfire 3.0 e 4.0 (VB, J-Link e Pro/Web.Link) para que possa ser feita uma análise comparativa entre as mesmas. Esta análise comparativa tem como pretensão auxiliar o processo de seleção da API mais adequada para utilizar nas aplicações do programa Colombo e noutras aplicações que se revelem necessárias no futuro. De seguida devem ser enumerados os requisitos das aplicações ProPEditor e ProCatalog e identificados e descritos os seus casos de uso para selecionar uma API que satisfaça estes requisitos e que consiga implementar as funcionalidades do Pro/E requeridas. Cabe ao departamento de SI da Efacec decidir se a API sugerida pelo autor desta dissertação deve ser a escolhida para implementar as aplicações.

1.5 Método de Trabalho

Objetivo **Estudo da API selecionada**: no contexto deste estágio, uma vez que a prioridade é o desenvolvimento das aplicações ProPEditor e ProCatalog, o estudo das classes e funcionalidades da API selecionada, a API Pro/Web.Link, será feito em paralelo com a programação destas aplicações. Para o estudo pode recorrer-se à documentação e aos exemplos de aplicações fornecidos pela PTC. Numa fase posterior ao estágio, pretende-se completar este objetivo estudando esta API com mais pormenor com o propósito de auxiliar outros programadores através da complementação da literatura existente.

Objetivos **Implementar Aplicação ProPEditor** e **Implementar Aplicação ProCatalog**: implementação das duas aplicações **ProPEditor** (Formulário Para o Preenchimento das Legendas dos Desenhos no Pro/ENGINEER) e **ProCatalog** (Catálogo Web para a Localização de Artigos de Fixação). A análise das aplicações deve ser efetuada na fase de seleção da API e como tal deve iniciar-se esta etapa com o desenvolvimento das aplicações. Durante a programação das aplicações e de acordo com o *feedback* dos utilizadores no período de testes, podem surgir novas ideias e novos requisitos de forma a ir ao encontro das melhores soluções e por essa razão pode ser necessário efetuar nova análise às aplicações. No final da implementação de cada aplicação deve ser elaborado um manual de utilizador em Inglês, Português e Espanhol.

A tarefa **Implementar aplicação MassProp** não faz parte da lista de objetivos desta dissertação e como tal não é descrita na próxima secção. O departamento de SI decidiu que esta aplicação devia ser implementada uma vez que esta pode facilitar o trabalho dos engenheiros e contribuir para a exploração de novas funcionalidades da API Pro/Web.Link, não exigindo demasiado tempo para a programar.

1.5 Método de Trabalho

Para o desenvolvimento das aplicações foram seguidos alguns dos princípios do Manifesto Ágil, nomeadamente a “entrega rápida e contínua de *software* com valor”, com “períodos de entrega de poucas semanas”, podendo “aceitar alterações de requisitos mesmo numa fase tardia do ciclo de desenvolvimento” [Beck et al., 2001].

1.6 Contribuições

Durante os 8 meses de Estágio foram fornecidos os seguintes contributos:

- Estruturação dos acessórios de fixação na plataforma GlobalArt através da definição das suas características globais (Objetivo **Organizar Dados dos Acessórios de Fixação no GlobalArt**);
- Documento com a classificação dos acessórios de fixação para suportar os utilizadores na criação de artigos no sistema GlobalArt (Objetivo **Organizar Dados dos Acessórios de Fixação no GlobalArt**);
- Duas aplicações em PHP para integrarem a intranet da unidade AMT: uma para a geração de portais dos vários departamentos (Produção, Informática, etc.) de cada companhia, e outra para a edição desses portais (Objetivo **Estudar Framework da Intranet AMT (SynergyNet)**);
- Relatório sobre as APIs do Pro/E com informação essencial sobre as suas principais características para suportar o processo de seleção de uma API (Objetivo **Selecionar API do Pro/ENGINEER**);
- Documentos com a descrição dos casos de uso das aplicações ProPEditor e ProCatalog e com outras especificações (Objetivos **Selecionar API do Pro/ENGINEER**, **Implementar Aplicação ProPEditor** e **Implementar Aplicação ProCatalog**);
- Sugestão da API Pro/web.Link para o desenvolvimento das aplicações ProPEditor e ProCatalog (Objetivo **Selecionar API do Pro/ENGINEER**);
- Versão melhorada do módulo JavaScript da PTC, que contém as funções comuns a todas as aplicações que sejam desenvolvidas com a API Pro/Web.Link, no que diz respeito à organização do código e ao tratamento de erros (Objetivo **Estudar API Selecionada (Pro/Web.Link)**);
- O código PHP e JavaScript das aplicações ProPEditor e ProCatalog (Objetivos **Implementar Aplicação ProPEditor** e **Implementar Aplicação ProCatalog**);
- O código PHP e JavaScript da aplicação MassProp para integrar o Pro/E. Esta aplicação fornece as *mass properties* (volume, área, massa e densidade) dos componentes de um *assembly* (Objetivo **Implementar Aplicação MassProp**, definido após a implementação da aplicação ProPEditor);
- Dois manuais de utilizador do ProPEditor e do ProCatalog em Português, Inglês e Espanhol, com a identificação das vantagens, regras das aplicações e descrição das

1 Introdução

suas funcionalidades (Objetivos **Implementar Aplicação ProPEditor** e **Implementar Aplicação ProCatalog**).

Após o período de estágio foram fornecidos os seguintes contributos:

- Documentação técnica do ProPEditor e ProCatalog para auxiliar os programadores do departamento de SI com a apresentação do contexto do problema, o modelo de dados, descrição das funções criadas e dos métodos e propriedades utilizados da API Pro/Web.Link. A esta documentação foram anexados os documentos, desenvolvidos durante o estágio, com a descrição dos casos de uso e de outras especificações (Objetivos **Implementar Aplicação ProPEditor** e **Implementar Aplicação ProCatalog**);
- Documentação sobre a API Pro/Web.Link com a seguinte informação: arquitetura da API; os tipos de classes que a constituem; o processo de configuração do ambiente de desenvolvimento e o *deployment* de uma aplicação; o módulo JavaScript para ser utilizado em todas as aplicações programadas com a Pro/Web.Link; identificação de um método de trabalho adequado para desenvolver aplicações com a API Pro/Web.Link; e outros conceitos relevantes que estão contidos nesta dissertação. (Objetivo **Estudar API Seleccionada (Pro/Web.Link)**).

1.7 Apresentação da Empresa

O Grupo Efacec é atualmente o maior grupo Eléctrico Nacional de capitais portugueses e teve origem em 1905 com a inauguração de “A Moderna, Sociedade de Serração Mecânica de Madeiras”, começando a produzir em 1917 os primeiros motores eléctricos fabricados em Portugal. Sendo constituído em 1948 e tendo mais de 4.500 colaboradores e um volume de negócios que ultrapassou os 1000 milhões de euros, este aposta fortemente na internacionalização atuando em mais de 65 países em todo o mundo e exportando aproximadamente metade da sua produção [Efacec, 2012c].

Com uma presença mundial mais alargada, estando no sector da energia aos transportes e à engenharia, do ambiente aos serviços e às energias renováveis, o Grupo Efacec foca a sua atividade em oito regiões que constituem as suas Unidades de Mercado: Portugal, Estados Unidos da América; América Latina (Brasil, Argentina, Chile, Paraguai, Uruguai); Europa Central (Roménia, República da Bulgária, República Checa, Áustria, Grécia, Eslováquia e

1.7 Apresentação da Empresa

Hungria); Magrebe (Argélia, Marrocos e Tunísia); África Austral (Angola, África do Sul e Moçambique); Espanha e Índia [Efacec, 2012d].



Figura 6 - Mapa-mundo com 8 regiões de atividade do grupo Efacec [Efacec, 2012b]

O Grupo Efacec é constituído por 3 áreas de negócio e organiza-se em 10 Unidades de Negócio que fornecem soluções, serviços e produtos: Energias (Transformadores, Aparelhagem de Média e Alta Tensão, Servicing de Energia); Engenharia, Ambiente e Serviços (Engenharia, Automação, Manutenção, Ambiente, Renováveis) e Transportes e Logística (Transportes, Logística) [Efacec, 2012e]. Todas estas unidades possuem recursos comerciais, engenharia, I&D (Investigação e Desenvolvimento), produção e logística. A unidade de Aparelhagem de Média e Alta tensão está encarregue da construção de equipamentos de distribuição e transmissão de energia de média e alta tensão.

Por razões logísticas e financeiras algumas destas unidades representam uma companhia, onde por exemplo, a aparelhagem de Média e Alta Tensão representa a companhia 453. A unidade de Aparelhagem dirige as companhias da Argentina 704 (Efacec Argentina CA) e 721 (Efacec Argentina CC), a companhia da Índia 724 (Efacec C&S) e da Espanha 735 (Efacec Equipos) uma vez que esta possui a maioria ou todo o capital dessas empresas. A companhia 723 (C&S Efacec) é uma empresa da Índia com direção partilhada, sendo 50% da aparelhagem e 50% de um parceiro local.

1 Introdução

A unidade de Aparelhagem tem vários departamentos: Engenharia, Produção, Comercial, Aprovisionamento, Sistemas de Informação, entre outros. O departamento de Sistemas de Informação tem como funções o desenvolvimento de aplicações *Web* que automatizam, agilizam e eliminam erros nas principais atividades da unidade de negócio das energias e de outras unidades de negócio, tanto em Portugal como no estrangeiro.

1.8 Estrutura da Dissertação

Esta dissertação encontra-se organizada em nove capítulos. O primeiro capítulo **Introdução** contém uma visão geral dos objetivos que se pretendiam atingir com o estágio/dissertação, apresentando o contexto do problema, o calendário de trabalho final do estágio, o método de trabalho utilizado para desenvolver as aplicações propostas, as contribuições fornecidas durante e após o estágio e a descrição do negócio da Efacec.

Os restantes capítulos têm como finalidade dissertar sobre os objetivos cumpridos durante e após o estágio através da descrição e discussão do trabalho efetuado, da apresentação da informação necessária para a sua compreensão e dos resultados atingidos.

O segundo capítulo **Intranet AMT-SynergyNet** é constituído: pela descrição do *Framework* da intranet da unidade AMT (SynergyNet), apresentando algumas das suas funcionalidades e aplicações da intranet; pela descrição, discussão e apresentação dos objetivos cumpridos *Organizar Dados dos Acessórios de Fixação no GlobalArt e Estudar Framework da Intranet AMT (SynergyNet)*.

O terceiro capítulo **PDS – Product Development System** é constituído: pela apresentação do sistema de desenvolvimento de produto (PDS) utilizado na empresa que inclui o Pro/ENGINEER e Windchill; pela descrição, discussão e apresentação do objetivo cumprido *Selecionar API do Pro/ENGINEER*. Este objetivo permitiu resumir numa tabela as características principais das APIs J-Link, VB e Pro/Web.Link.

O quarto capítulo **Pro/Web.Link** apresenta parte do resultado do estudo efetuado à API Pro/Web.Link (objetivo *Estudar API Selecionada*). Este capítulo é constituído pela apresentação da arquitetura da API, pela identificação e descrição das suas classes e pela explicação de alguns conceitos relacionados essas classes, nomeadamente como estas podem ser instanciadas e/ou utilizadas, com a capacidade de herança e com a documentação da API.

1.8 Estrutura da Dissertação

O quinto capítulo **Integração Intranet – Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link** completa o estudo efetuado à API (objetivo *Estudar API Seleccionada*) abordando alguns aspetos relacionados com a interação entre as aplicações *Web* e o Pro/E. Este capítulo apresenta: como configurar o ambiente de desenvolvimento das aplicações que utilizam a API Pro/Web.Link para estas interagirem com o Pro/E e alguns conceitos inerentes a esse processo; o módulo JavaScript adaptado do módulo da PTC, que permite detetar erros de configuração do ambiente de desenvolvimento e efetuar algumas validações; a proposta de um método de trabalho adequado para auxiliar outros programadores a desenvolver aplicações com esta API; e alguns aspetos relacionados com as tecnologias utilizadas no desenvolvimento das aplicações ProPEditor, ProCatalog e MassProp, nomeadamente as nomeação das tecnologias utilizadas e dos componentes envolvidos.

Os capítulos **ProPEditor**, **ProCatalog** e **MassProp** apresentam uma contextualização das aplicações, a descrição dos casos de uso, apresentando a interface da aplicação final e as vantagens das aplicações.

O último capítulo **Conclusões** agrega todos os objetivos cumpridos e contribuições fornecidas durante o estágio, resumindo tudo aquilo que foi feito, avaliando o sucesso dos objetivos e finalizando com uma reflexão crítica ao trabalho.

1 Introdução

2 Intranet AMT - SynergyNet

Este capítulo encontra-se dividido em cinco secções. A primeira secção **2.1 Introdução** introduz a intranet da unidade AMT, fazendo uma breve referência às aplicações Profiler e Translator, que efetuam a gestão dos utilizadores e das traduções da intranet respetivamente. A segunda secção **2.2 Arquitetura da SynergyNet** expõe a arquitetura física e lógica da intranet. A terceira secção **2.3 Framework da SynergyNet** descreve o *Framework* de desenvolvimento de aplicações *Web* da intranet AMT que, como se verá no capítulo **5 Integração Intranet – Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link**, é usado para desenhar a interface das aplicações do programa Colombo, e descreve as aplicações, desenvolvidas para a Efacec durante o estágio, de geração e edição dos portais dos departamentos. A quarta secção **2.4 GlobalArt** corresponde à apresentação da aplicação GlobalArt, que permite gerir os artigos da Efacec e, como se verá no capítulo **7 ProCatalog**, é a fonte de informação da aplicação ProCatalog, dos seus conceitos, e à descrição das tarefas realizadas durante o estágio, nomeadamente a organização dos dados dos acessórios de fixação no GlobalArt e a escrita da documentação dos acessórios de fixação. A quinta secção **2.5 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

2.1 Introdução

A intranet da unidade AMT (SynergyNet) é uma rede de comunicação/informação interna da Efacec que disponibiliza várias aplicações e tem um sistema de *login* para restringir os acessos, que consiste num processo de identificação e autenticação do utilizador através da introdução do número do utilizador e da sua palavra-chave.

2 Intranet AMT - SynergyNet

A SynergyNet tem um público-alvo restrito, estando acessível às companhias logísticas de Portugal (companhias Servicing de Energia, AMT, e à companhia da Efacec Renováveis), e às companhias dirigidas pela unidade da Aparelhagem que estão situadas na Argentina, Índia e Espanha. As companhias das Renováveis e Servicing usam a SynergyNet numa ótica de partilha, não tendo ligação com a Aparelhagem. Os fornecedores e clientes comuns a estas companhias também têm acesso à SynergyNet pois muitas das suas aplicações são direcionadas para eles. Nestas situações a intranet é considerada uma extranet, onde se tem acesso a ela a partir da exterior via *Web*.

A SynergyNet tem uma aplicação designada Profiler, desenvolvida pelo departamento de SI, que permite gerir os dados de dois tipos de utilizadores: utilizadores geridos pelo Active Directory do domínio Windows da organização e utilizadores existentes apenas no Profiler (fornecedores, clientes). No caso dos primeiros, as informações base (*username, password, supervisor, e-mail*) são geridas exclusivamente pelo Active Directory e obtidas pelo Profiler a partir de um *web service*. No caso dos segundos, a informação é gerida por utilizadores com permissão no Profiler para tal. Outras informações mais específicas da SynergyNet, tais como a listagem das companhias a que cada utilizador tem acesso são geridas exclusivamente pelo Profiler.

Após o *login*, a SynergyNet disponibiliza uma barra com algumas informações (o nome do utilizador que está atualmente a utilizar a sessão, a companhia a que este pertence e o idioma em que a página é apresentada) e ligações (uma página de ajuda, um conjunto de *links* com todas as aplicações desenvolvidas pela unidade AMT para o utilizador aceder e uma opção *sair* para terminar a sessão da SynergyNet).

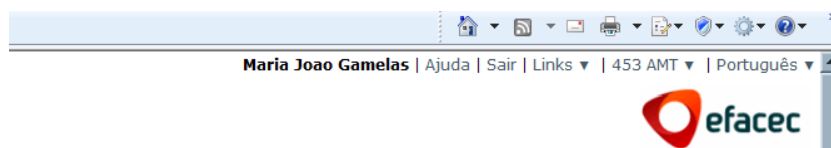


Figura 7 - Barra de ferramentas da SynergyNet

De momento, a SynergyNet funciona em ambiente multilingue e os idiomas disponíveis para tradução são o Português, Inglês e Espanhol que são as línguas mais faladas nas companhias

2.1 Introdução

que têm acesso à SynergyNet. Esta opção de tradução permite que todas as aplicações da SynergyNet possam ser traduzidas e visualizadas pelo utilizador em três línguas diferentes.

Como se pode verificar na figura seguinte (*Figura 8*), à medida que o utilizador vai alternando o idioma na aplicação ProCatalog a página é traduzida para o idioma respetivo.



a) Página do ProCatalog traduzida para Inglês

b) Página do ProCatalog traduzida para Castelhanos



c) Página do ProCatalog traduzida para Português

Figura 8 - Página do ProCatalog traduzida para (a) Inglês, (b) Castelhanos e (c) Português

A aplicação que permite a tradução do conteúdo das páginas da SynergyNet para um destes idiomas foi desenvolvida no departamento de SI da unidade da AMT e designa-se por Translator. Embora atualmente os idiomas de interesse sejam o Português, Inglês e Espanhol, o Translator está preparado para receber traduções em novas línguas, o que permitirá adicionar novos idiomas à SynergyNet em caso de necessidade. Para que a tradução possa ser feita é necessário criá-la previamente no Translator através da identificação de um código, do

resultado da tradução e do idioma para o qual se traduziu. Após este momento, a função de tradução do *Framework* pode ser utilizada para o código criado. Por exemplo, para traduzir a frase *Anilhas e Placas Oblíquas* para Inglês tem de ser criado no Tradutor o seguinte: um código (ex: *washersPlate*); o idioma destino (ex: Inglês); e o resultado (ex: *Washers and Oblique Plates*). Para permitir a tradução da mesma frase para Espanhol recorre-se ao mesmo processo anterior utilizando o mesmo código (*washersPlates*) mas desta vez com o idioma *Espanhol* e com o resultado *Arandelas e Placas Oblicuas*.

2.2 Arquitetura da SynergyNet

Em termos de arquitetura física, a SynergyNet é um sistema tradicional de três camadas (*three-tier*): cliente; servidor aplicacional, com Apache e PHP; e servidor de base de dados com Microsoft SQL Server.

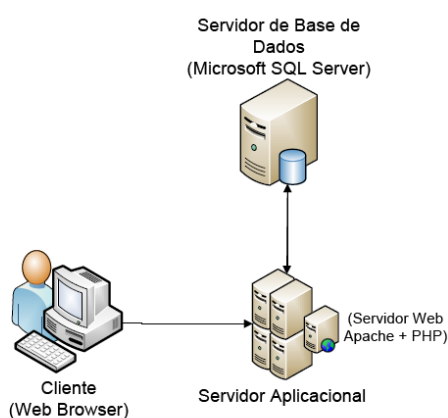


Figura 9 - Arquitetura física da SynergyNet (*Three-Tier*)

As aplicações *Web*, criadas em PHP, estão armazenadas no servidor aplicacional e sempre que estas necessitam de obter dados do Microsoft SQL Server o servidor aplicacional acede ao servidor de base de dados. O servidor aplicacional é constituído por um servidor *Web* Apache e por um *Framework* PHP, desenvolvido pelo departamento de SI, para auxiliar o desenvolvimento de aplicações *Web*. A próxima secção apresenta uma breve descrição deste *Framework*.

2.3 Framework da SynergyNet

Ao nível da arquitetura lógica, a SynergyNet é um sistema de duas camadas: apresentação; negócio + dados uma vez que é um requisito não funcional que as regras de negócio têm de estar em *stored procedures* da base de dados.

2.3 Framework da SynergyNet

O servidor aplicacional da Efacec possui um *Framework PHP* para o desenvolvimento de aplicações *Web*, criado no departamento de SI, que detém um conjunto de classes com variáveis e métodos. Estas classes podem ser visualizadas num documento *Web* e facilitam a obtenção de alguns dados relativos à sessão da intranet, como também o desenho da interface (tabelas, formulários, etc.), os pedidos às bases de dados, as traduções dos conteúdos, entre outras funcionalidades. Desta forma, evita-se o uso de código repetitivo tornando mais eficiente o processo de desenvolvimento e manutenção das aplicações, e auxiliando a comunicação entre os programadores do departamento SI que facilmente entendem o código pelo facto de recorrerem a funções *standard*.

“A SynergyNet implementa o padrão *FrontController*³ em que o cliente efetua o pedido sempre ao controlador da SynergyNet indicando o módulo ao qual pretende aceder e o controlador encarrega-se de validar se o utilizador possui uma sessão válida e se possui acesso ao módulo que pretende” [Pinheiro C., 2011].

As principais classes do controlador são a *CInterface* e *CMain*.

A classe *CInterface* é responsável pelo desenho do *header* e do menu da página da SynergyNet e pela adaptação da interface gráfica ao tipo de equipamento que acede à aplicação, que pode ser um computador, um PDA (*Personal Digital Assistant*).

³Front Controller, <http://www.martinfowler.com/eaCatalog/frontController.html>

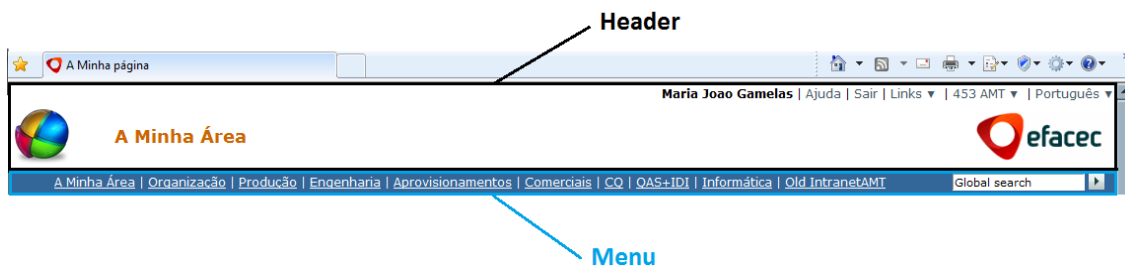


Figura 10 - Header e menu da SynergyNet

A classe CMain é uma subclasse da classe CInterface e possui os atributos e métodos principais de validação de acesso do utilizador à SynergyNet e ao módulo invocado [Pinheiro C., 2011].

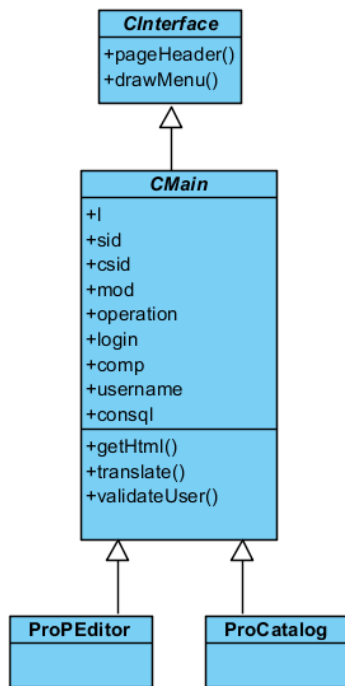


Figura 11 - Diagrama de classes (Adaptação do diagrama de [Pinheiro C., 2011])

A classe CMain também disponibiliza outros atributos e métodos que podem ser ou não usados pelas aplicações consoante a necessidade. As classes ProPEditor e ProCatalog, que estendem a classe CMain, representam os módulos que o cliente invoca.

2.3 Framework da SynergyNet

Os principais atributos da classe CMain são os seguintes (*Figura 11*):

- *l*: Idioma ativo na sessão;
- *sid*: *Session Id*;
- *csid*: sid guardado num *cookie*;
- *mod* e *operation*: como um módulo é uma aplicação complexa e realiza diversas operações, é necessário indicar o que se quer fazer ao invocar o módulo. Para tal são usados os parâmetros *mod* e *operation*, cujos valores ficam ao critério do programador. Habitualmente o *mod* indica a funcionalidade e o *operation* indica uma operação específica dentro da funcionalidade (exemplos: *mod*: editArticle, *operation*: getForm; *mod*: editArticle, *operation*: submitForm);
- *login*: código de utilizador. Exemplo: 3398;
- *comp*: companhia ativa na sessão;
- *username*: nome do utilizador. Exemplo: Maria João Gamelas;
- *consql*: objeto ADO (*ActiveX Data Object*) de acesso à base de dados.

Ao invocar um determinado módulo, o *FrontController* carrega o ficheiro correspondente ao módulo e invoca o seu construtor. Dado que o módulo é uma extensão da classe CMain é invocado também o construtor desta classe que verifica se o utilizador tem uma sessão válida com acesso ao módulo invocado. Caso o utilizador possua permissões para aceder ao módulo o *FrontController* instancia as classes ProPEditor ou ProCatalog (dependendo do módulo solicitado) e invoca o método *getHTML()* para obter os conteúdos do módulo.

Os módulos devem ter no seu endereço a indicação dos parâmetros *l*, *sid*, *mod*, *operation* e *comp* com os respetivos valores para que se consiga reconhecer qual o idioma, o utilizador, o módulo da aplicação, a operação e a companhia que estão ativos na sessão. Neste endereço também podem estar outros parâmetros, sendo uma forma do PHP transmitir variáveis.

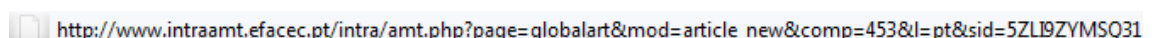
 http://www.intraamt.efacec.pt/intra/amt.php?page=globalart&mod=article_new&comp=453&l=pt&sid=5ZL9ZYMSQ31

Figura 12 - URL para o módulo de criação de um novo artigo da página GlobalArt

A *Figura 12* apresenta o exemplo do endereço da página da aplicação GlobalArt para criar um novo artigo. O endereço aponta para a página do GlobalArt (*page=globalArt*) e para o módulo *article_new* (*mod=article_new*), a companhia ativa da sessão da intranet é a companhia 453 (*comp=453*) da aparelhagem, o idioma ativo é o Português (*l=pt*) e o número de sessão é o 5ZLI9ZYMSQ31.

O *Framework* também contempla o módulo de traduções. Todas as traduções de cada módulo estão registadas numa base de dados e quando o objeto do módulo é criado estas são carregadas pelo construtor da classe *CMain* para uma coleção em memória. Sempre que é preciso traduzir um texto invoca-se o método *translate()* com o código da tradução e este método procura o código na *collection* e retorna o texto correspondente [Pinheiro C., 2011]. Este código deve ser previamente criado no Translator.

Tal como foi referido anteriormente, o *Framework* disponibiliza classes para facilitar a criação da interface gráfica: **CForm** para a criação de formulários; e **CTable** para a criação de tabelas. Todas as aplicações podem relacionar-se com estas classes. Apesar do uso destas classes durante o estágio para a programação de algumas aplicações *Web*, estas não são aprofundadas visto que não fazem parte do âmbito da presente dissertação.

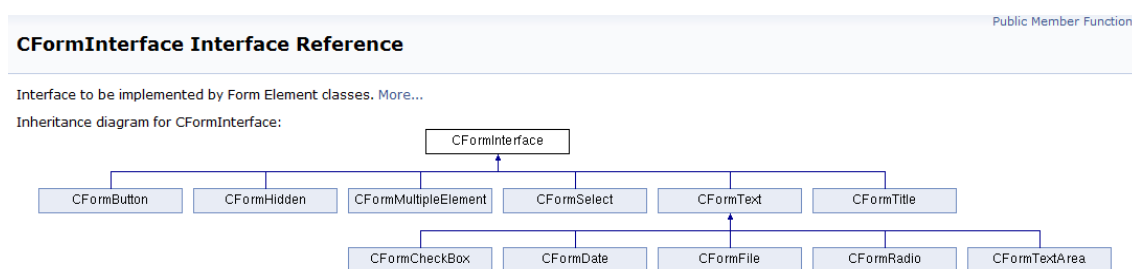


Figura 13 - Classe CForm e classes hereditárias

Como demonstra a *Figura 13*, outras classes herdam a classe *CForm* para a criação de elementos para o formulário, designadamente as classes:

- **CFormButton**: botão;
- **CFormHidden**: campo de texto oculto;
- **CFormMultipleElement**: campo de texto com múltiplos valores;

2.3 Framework da SynergyNet

- **CFormSelect:** campo de seleção de valores;
- **CFormText:** campo de texto. Desta classe herdam cinco classes que podem ser utilizadas:
 - **CFormCheckBox:** campos de seleção do tipo “caixa”;
 - **CFormDate:** campos que aceitam unicamente datas e permitem selecionar uma data a partir de um calendário;
 - **CFormFile:** campos que aceitam somente ficheiros e permitem selecionar um ficheiro a partir da sua localização;
 - **CFormRadio:** campos de seleção de exclusão mútua;
 - **CFormTextArea:** caixa de texto;
- **CFormTitle:** título do formulário.

2.3.1 Aplicações de Geração e Edição dos Portais dos Departamentos

Para desenvolver as aplicações sugeridas no programa Colombo é necessário utilizar o *Framework* da intranet da AMT. Para adquirir uma maior compreensão do *Framework* e, ao mesmo tempo, satisfazer uma necessidade imediata da Efacec, foi proposta a criação de duas aplicações: uma para gerar automaticamente os portais dos vários departamentos das companhias da Efacec e outra para editar os portais anteriormente criados. Para auxiliar este trabalho recorreu-se ao *phpdocs*, um documento *Web* que detém as variáveis e métodos do *Framework*. As aplicações foram implementadas na plataforma de desenvolvimento *NetBeans*.

Apenas algumas companhias têm permissão para aceder à intranet da unidade AMT. Cada uma destas companhias pode estar ligada a vários departamentos: Recursos Humanos, Informática, Controlo, Produção, Aprovisionamento, entre outros. Como se observa na figura seguinte (*Figura 14*), as antigas páginas destes departamentos restringiam-se a expôr atalhos para as restantes aplicações da intranet mais usadas.

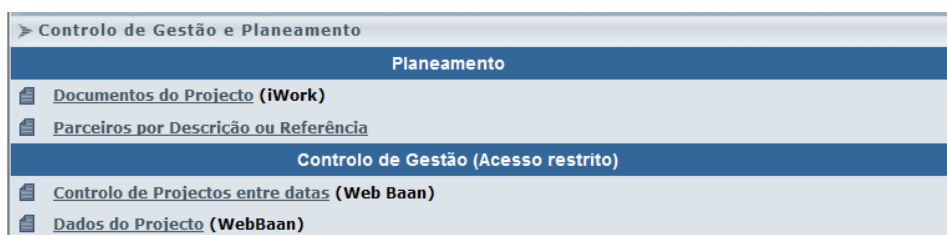


Figura 14 - Fragmento do antigo portal do departamento de Controlo

Para contornar esta limitação e disponibilizar campos de pesquisa, foi sugerido o desenvolvimento de uma aplicação que gera os portais dos diversos departamentos em tempo real (consultar [Apêndice A](#)). Desta forma, os utilizadores podem visualizar sempre a versão atualizada da página.

As antigas páginas dos departamentos também não eram flexíveis pois cada vez que os utilizadores pretendiam adicionar ou remover uma hiperligação tinham de o solicitar à equipa de SI. Para possibilitar a edição das páginas sugeriu-se também a implementação de uma aplicação que permita que os responsáveis pelos departamentos possam modificá-las consoante as suas necessidades, tornando a página mais dinâmica (consultar [Apêndice B](#)).

2.4 GlobalArt



Figura 15 - Plataforma GlobalArt

2.4 GlobalArt

O Grupo Efacec tem replicado o modelo de negócio presente em Portugal com a criação de várias empresas por todo o mundo. Esta crescente internacionalização exigiu o advento de um sistema centralizado, o GlobalArt, para que a criação e classificação de artigos fossem efetuadas de uma forma coerente e integrada.

O GlobalArt é uma aplicação *Web* inserida na intranet AMT que permite gerir todos os artigos das várias companhias. Estes artigos representam todos os materiais adquiridos ou fabricados pela Efacec. Concentrando todos os artigos numa única plataforma em ambiente multilingue consegue-se combater o problema da redundância dos dados.

O GlobalArt está totalmente integrado com o ERP (*Enterprise Resource Planning*) da Efacec, o Baan, que gere toda a informação interna e externa da organização englobando as compras, vendas, os dados dos artigos, entre outros elementos e processos de negócio. O GlobalArt tem uma função que permite que determinados utilizadores possam criar novos artigos no Baan. Normalmente o processo de sincronização automática entre o GlobalArt e o Baan é feito duas vezes por dia. Há uma versão *Web* do ERP Baan, também desenvolvida pelo departamento de SI da AMT, intitulado por WebBaan, onde o utilizador pode consultar os dados e utilizar as funcionalidades disponibilizadas pelo Baan, com a vantagem de ser em ambiente multilingue.

De acordo com o guia do GlobalArt para os utilizadores [Efacec Switchgear, 2009], o GlobalArt foi criado com o intuito de ultrapassar algumas das seguintes limitações do Baan:

- Evitar artigos repetidos, ou seja, evitar que em armazém estejam códigos diferentes para o mesmo material;
- Evitar que o mesmo código de artigo represente materiais diferentes em companhias diferentes;
- Centralizar e disponibilizar os dados dos artigos a outros sistemas;
- Permitir atribuir outras características aos artigos;
- Possibilitar a gestão multilingue dos artigos e das suas características para que o utilizador da intranet AMT possa ver toda a informação no seu próprio idioma;
- Pesquisar os artigos pelas suas características em ambiente multilingue.

O GlobalArt é também o sistema base de outras aplicações *Web* desenvolvidas na Efacec, tais como o GlobalTracking, que é um sistema *Web* para gerir todas as atividades relacionadas

com os materiais, e o AMTProPartnersLink, que gere os contactos entre a Efacec Aparelhagem e os seus fornecedores, controlando os processos de compras.

Como demonstra a *Figura 16* abaixo, cada artigo está inserido numa ou mais companhias e numa companhia principal (*master company*). O artigo pode ser comprado ou fabricado (tipo de artigo) e pertence a uma família corporativa, a uma classe de artigo, a uma família de artigos (Grupo) e a um armazém.

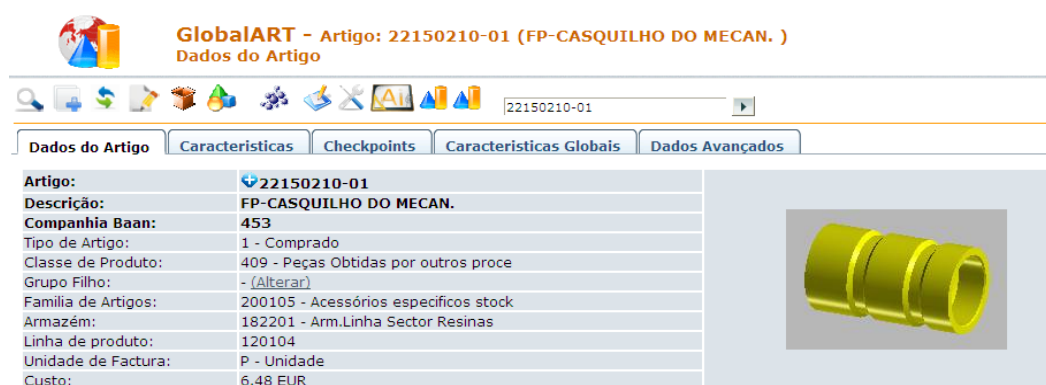


Figura 16 - Plataforma GlobalArt com dados de um artigo

Aproveitando o conceito do Baan de **família corporativa**, aplicado para categorizar os artigos da Efacec, o GlobalArt define um *standard* para o registo e qualificação de artigos. Este conceito pode ser expandido por **classes de artigo**, conceito criado no GlobalArt como forma de simplificar a associação entre as características e os artigos uma vez que dentro de uma família corporativa podem estar artigos com diferentes características. Embora na maioria dos casos a família corporativa seja igual à classe do artigo, esta também pode estar subdividida em mais do que uma classe de artigo.

Estas classes estão associadas a **características** que podem ser **globais** (aplicam-se a todas as companhias) ou **locais** (são específicas para uma determinada companhia) [Efacec Switchgear, 2009]. A característica é um atributo do artigo. Por exemplo, o artigo parafuso tem características como o comprimento, diâmetro, etc.

2.4 GlobalArt

Um determinado artigo pode pertencer a mais do que uma companhia, pelo que surgiu a necessidade de criar o conceito de **companhia mestre (Master Company)**, que é a companhia que detém a propriedade do artigo. Essa companhia é a única que pode alterar os valores das características globais dos artigos. As restantes características do ERP (dados de custo, dados de produção, dados de armazém, etc.) podem e devem também ser alteradas pelas restantes companhias pois variam de companhia para companhia.

A **família de artigos** ou grupo é uma característica que tem como propósito a criação de um artigo no Baan, funcionando como um modelo (*template*). Quando se cria um artigo no Baan e se indica um grupo, o novo artigo herda as características base desse grupo, tais como os dados de venda, dados de compra, dados de produção, dados de custo, dados de planeamento, dados de armazém, etc. Estas características podem ser alteradas durante a criação ou posteriormente.

2.4.1 Organização dos Dados dos Acessórios de Fixação no GlobalArt

A plataforma GlobalArt deve ser a fonte de informação da aplicação ProCatalog, e como tal é essencial que o GlobalArt tenha os dados dos artigos bem estruturados e com toda a informação necessária para se conseguir catalogá-los devidamente.









Embora o GlobalArt tivesse uma série de artigos, os artigos de fixação não estavam na estrutura ideal para suportar a aplicação do catálogo, ou seja, estes não estavam devidamente classificados pois não possuíam características globais, pelo que era fundamental atualizar a sua base de dados com a nova estrutura. Para esse fim, foi necessário definir as características globais de cada um dos tipos acessórios de fixação no GlobalArt, importar os dados de todos os acessórios a partir do Windchill para inferir os valores das características identificadas anteriormente e atualizar esta informação no GlobalArt através da migração dos dados.

2.4.1.1 Definição das Características Globais dos Acessórios de Fixação no GlobalArt

Os tipos de artigos de fixação que devem ser atualizados para incluir o catálogo ProCatalog são: parafusos, porcas, anilhas, rebites, cavilhas, golpilhas, *circlips* e freios.

A Tabela 1 expressa as características globais que foram identificadas por parte do autor e aprovadas por um dos membros responsáveis do departamento de SI após reunião com um engenheiro mecânico.

Tabela 1 - Características globais dos Acessórios de Fixação

Anilha	Cavilha	Golpilha	Circlip	Freio
				
Classe Qualidade Acessório Diâmetro Externo Diâmetro Interno Espessura Matéria-Prima Norma Proteção Tipo de Anilha	Comprimento Diâmetro Formato da Cavilha/ Golpilha Matéria-Prima Norma Proteção Tipo de Cavilha/ Golpilha		Diâmetro Matéria-Prima Norma Tipo de Circlip/ Freio	
Parafuso	Porca	Rebite		
				
Cabeça Classe qualidade Acessório Comprimento Comprimento da Rosca Diâmetro Matéria-Prima Norma Proteção Tipo de Chave Tipo de Rosca	Classe Qualidade Acessório Diâmetro Interno Formato da Porca Matéria-Prima Norma Revestimento Superior Tipo de Rosca	Comprimento Diâmetro Formato do Rebite Matéria – Prima Norma		

2.4 GlobalArt

A norma é uma característica global que está sempre presente nos artigos de fixação. A Efacec utiliza acessórios com normas internas (NA, ENA e INA), e com normas ISO (*International Standards Organization*) e DIN (*Deutsches Institut fur Normung*). Faz parte dos objetivos da Efacec substituir gradualmente todas as normas internas por normas ISO ou DIN equivalentes, uma vez que estas são conhecidas globalmente.

Após a identificação das características de cada tipo de artigo foi necessário associar cada uma delas às respetivas classes de artigo e famílias corporativas no GlobalArt. Cada um destes tipos de artigos está inserido numa classe e numa família corporativa. A configuração destas características veio facilitar a pesquisa dos acessórios de fixação no GlobalArt, que passou a ser feita também por características.

2.4.1.2 Tratamento dos Dados e Acompanhamento da sua Migração para o GlobalArt

Após a definição das características no GlobalArt foi necessário tratar os dados exportados do Windchill, que tinha a informação mais completa dos acessórios de fixação, e distribuir a informação obtida pelas várias características para que posteriormente fosse possível migrá-los para a base de dados do GlobalArt. Os dados exportados estavam na seguinte forma:

Number	Name	Context	Article	Local Description
FREIO_12_DIN6799.PRT	freio_12_DIN6799.prt	STANDARDS	9053314	FREIO A 12 DIN-6799

Figura 17 - Dados de um acessório de fixação – Freio exportado do Windchill

Visualizando os dados recolhidos do separador *Local Description* foi possível distribuí-los pelas várias características anteriormente identificadas.

Tipo de circlip/freio	Diâmetro	Matéria-prima	Norma
C	12	steel	DIN 6799

Figura 18 - Características de um acessório de fixação - Freio

O acessório da

Figura 18 é um freio e como tal as características globais são: tipo de *circlip*/freio, diâmetro, matéria-prima e norma. Segundo o livro das normas da Efacec, o tipo de *circlip*/freio pode ser: um freio; um *circlip* exterior; ou um *circlip* interior. Para representar estes tipos foram criados no GlobalArt os seguintes valores de entrada para a característica “tipo de *circlip*/freio”: C (Freio), CE (*Circlip Exterior*); e CI (*Circlip Interior*). No exemplo acima, o artigo é do tipo C (Freio), feito de aço (apontado pela letra “A” no campo “Local Description”), de diâmetro 12 e respeita a norma DIN 6799.

Para uma melhor compreensão dos dados e para a sua organização pelas várias características recorreu-se ao livro das normas da Efacec.

Visto que o volume de dados relativos aos artigos era grande, foram programadas algumas macros no Excel para colocar, de forma automática, alguns dados nos respetivos campos. Para auxiliar as macros foi construída uma tabela com os possíveis valores que cada uma das características podia ter. A função da macro era procurar por esses valores no campo *Local Description* e colocar os valores encontrados no campo da característica correspondente.

Após o tratamento de todos os dados acompanhou-se a migração de cerca de 2.500 artigos para a base de dados do GlobalArt. Estes artigos estão agora devidamente classificados para que possam ser utilizados pela aplicação ProCatalog.

Segue-se um exemplo de um freio caracterizado no GlobalArt:

The screenshot shows the GlobalArt interface for article 9053314 (FREIO A 12 DIN-6799). The top navigation bar includes 'Global search' and 'Pesquisar' fields. Below the navigation bar, there are tabs for 'Dados do Artigo', 'Características', 'Checkpoints', and 'Informações do fornecedor do artigo'. The 'Características' tab is active, displaying a table with the following data:

!	!	Ordem	Característica	Valor	Alterado por	Data modificação	Operações
<input type="checkbox"/>		0	Tipo de circlip/freio	Freio	Luis Gomes	2010-01-01 00:00	
<input type="checkbox"/>		0	Diâmetro [mm]	12	Luis Gomes	2010-01-01 00:00	
<input type="checkbox"/>		40	Norma adoptada	DIN 6799	Luis Gomes	2010-01-01 00:00	
<input type="checkbox"/>		98	Matéria-prima	Aço	Luis Gomes	2010-01-01 00:00	

Figura 19 - Artigo caracterizado no GlobalArt

2.4 GlobalArt

2.4.1.3 Elaboração de Documentos com a Classificação dos Acessórios de Fixação

O GlobalArt é uma plataforma onde utilizadores autorizados podem criar e classificar artigos. Para apoiar essa tarefa foi elaborado um documento para os artigos de fixação usados na Efacec (porcas, parafusos, anilhas, rebites, cavilhas, golpilhas, *circlips* e freios), com as diferentes normas usadas.

Segue-se o exemplo abaixo para os freios e *circlips*:



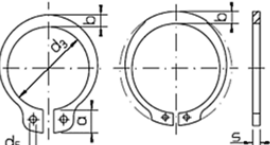

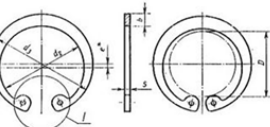

Acronym	Description	Standard		Image	EFACEC Acronym	EFACEC Description	EFACEC Norm	Ref.
C	Retaining Washer	DIN 6799			Freio	Freios de Localização	ENA 1039	http://catalog.lesjoforsab.com/pdf/en/c-hapters/145-145_EN.pdf
CI	External Retaining Rings	DIN 471			E	Circlips exteriores	NA 1040	http://www.springmasters.com/sp/standard-external-circlips.html
CE	Internal Retaining Rings	DIN 472			I	Circlips Interiores	NA 1041	http://www.springmasters.com/sp/standard-internal-circlips.html

Figura 20 - Documento de classificação dos freios e *circlips*

Em relação aos freios, a Efacec utiliza apenas os freios de localização de norma DIN 6799. Esta norma corresponde à norma da Efacec ENA 1039. Em termos de *circlips*, estes classificam-se em *circlips* exteriores Embora atualmente as normas da Efacec não sejam utilizadas, muitos dos artigos do GlobalArt ainda não estão associados a elas. Este documento de classificação, ao relacionar a antiga norma com a nova norma, facilitará o processo de atualização para as novas normas.

A primeira coluna de acrónimos da *Figura 20* apresenta os códigos internos utilizados no GlobalArt para cada um dos valores da característica “tipo de freio/*circlip*”. A segunda coluna tem as designações dos freios e *circlips* em inglês. As colunas 6 e 7 correspondem aos acrónimos e descrições aplicados no livro das normas da Efacec. A *Figura 20* também possui uma coluna de referências, onde é colocada a hiperligação (*link*) para o sítio onde foi retirada a informação para que mais tarde possa ser consultado. O livro das normas da Efacec e o catálogo de 2008 da Fabory⁴, empresa de venda de acessórios de fixação, ferramentas e produtos químicos, tiveram um papel importante para esta classificação.

Com esta documentação, que auxilia a nível visual ao recorrer a imagens, o utilizador pode identificar mais facilmente que tipo de acessório pretende criar no GlobalArt (por exemplo, se o utilizador quer criar um *circlip* exterior ou interior) e que norma utilizar, diminuindo as probabilidades de erro e contribuindo para uma boa classificação dos acessórios.

Para algumas das características globais foram criados no GlobalArt os valores de entrada possíveis para diminuir a ocorrência de erros na classificação de um acessório.

⁴ Fabory, <http://www.fabory.pt/>

2.5 Sumário

The screenshot shows the GlobalArt software interface for creating a new article in Baan. The interface includes a header with the GlobalArt logo and 'Novo Artigo' text, a navigation bar with menu items like 'A Minha Área', 'Organização', 'Produção', 'Engenharia', 'Logística', 'Comerciais', 'CQ', 'OAS+IDI', and 'Informática', and a search bar. The main area is divided into sections: 'Atualizar dados do Artigo' with fields for 'Fam. Corporativa' (120112) and 'Classe do Artigo' (Freios 120112); 'Características' with dropdowns for 'Tipo de circlip/freio' (Freio), 'Diâmetro [mm]' (12), 'Matéria-prima' (Aço), and a list for 'Norma adotada' with 'DIN 6799' selected; 'Criar Artigo no Baan' with fields for 'Descrição', 'Família' (200109 - Acessórios construação stock), 'Armazém', 'Unid.', 'Preço simulado', and 'Idioma' (Português); and a 'Texto' field containing technical specifications: 'Tipo de circlip/freio=Freio; Diâmetro [mm]=12; Matéria-prima=Aço; Norma adotada=DIN 6799;'.

Figura 21 - Página do GlobalArt para criação de um artigo no Baan

A *Figura 21* representa a funcionalidade do GlobalArt para a criação de um artigo. Os freios têm características definidas e, aquando da sua criação é necessário indicar os valores para essas características. No campo da “Norma adotada” aparece uma listagem de todas as normas utilizadas na Efacec e sem apoio um utilizador pouco experiente não sabe qual a norma correta. Com o apoio da documentação de classificação dos acessórios de fixação (*Figura 20*), um utilizador pode saber exatamente qual a norma a selecionar para criar, por exemplo, um freio: a norma DIN 6799. Dado que é possível associar a norma ao tipo de freio/circlip, a seleção automática da norma aquando da seleção do tipo de freio/circlip pode ser um automatismo do GlobalArt a implementar no futuro, facilitando ainda mais o trabalho de classificação.

2.5 Sumário

Neste capítulo foram alcançados os objetivos *Organizar Dados dos Acessórios de Fixação no GlobalArt* e *Estudar Framework da Intranet AMT (SynergyNet)*, sendo fornecidos três contributos.

Objetivos

- **Organizar Dados dos Acessórios de Fixação no GlobalArt:** Com a nova estruturação dos dados dos acessórios de fixação, através da definição das suas características globais (norma, matéria-prima, etc.), o GlobalArt fica apto para satisfazer os requisitos da aplicação ProCatalog, nomeadamente o agrupamento e a filtragem dos artigos do catálogo *Web* por características. Para que novos artigos de fixação possam ser corretamente classificados pelos utilizadores responsáveis pela criação de artigos, foram elaborados documentos de apoio;
- **Estudar *Framework* da Intranet AMT (SynergyNet):** O desenvolvimento das aplicações de geração e edição dos portais dos departamentos possibilitou a aprendizagem das regras da intranet da AMT e dos métodos do seu *Framework*. Desta forma, foi mais fácil programar as funcionalidades PHP das aplicações do programa Colombo, permitindo reservar para essa fase somente o estudo da API Pro/Web.Link nessa fase. Estas aplicações também trouxeram benefícios acrescentados para os utilizadores pois vieram melhorar e dinamizar o seu espaço de trabalho, quer com um aspeto renovado, quer com conteúdos mais dinâmicos.

Contributos

- Estruturação dos acessórios de fixação na plataforma GlobalArt através da definição das suas características globais (Objetivo **Organizar os Dados dos Acessórios de Fixação no GlobalArt**);
- Documento com a classificação dos acessórios de fixação para suportar os utilizadores na criação de artigos no sistema GlobalArt (Objetivo **Organizar Dados dos Acessórios de Fixação no GlobalArt**);
- Duas aplicações em PHP para integrarem a intranet da unidade AMT: uma para a geração de portais dos vários departamentos (Produção, Informática, etc.) de cada companhia, e outra para a edição desses portais. (Objetivo **Estudar Framework da Intranet AMT (SynergyNet)**).

3 PDS – *Product Development System*

Este capítulo encontra-se dividido em quatro secções. A primeira secção **3.1 Introdução** faz uma breve introdução ao PDS (*Product Development System*) utilizado na Efacec, que é constituído pelo Pro/ENGINEER Wildfire 3.0 e pelo Windchill PDMLink 9.0. A segunda secção **3.2 Pro/ENGINEER Wildfire 3.0** apresenta alguns dos tipos de modelos que o Pro/E permite desenvolver e as suas principais características (paramétrico, associativo, baseado em características). Nesta secção também é feita uma análise às três APIs das versões Wildfire 3.0 e 4.0 (J-Link, VB e Pro/Web.Link) para auxiliar o processo de seleção da API mais adequada para o desenvolvimento das aplicações ProPEditor e ProCatalog e de outras aplicações, sendo apresentada a API escolhida após o processo de seleção realizado durante o estágio. A terceira secção **3.3 Windchill** descreve a arquitetura do Windchill PDMLink 9.0 e aborda alguns conceitos relacionados com a integração do Windchill com o Pro/E. A quarta secção **3.4 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

3.1 Introdução

Um dos principais objetivos deste estágio consiste em analisar as APIs J-Link, VB e Pro/Web.Link do Pro/E para que o departamento de SI da Efacec possa selecionar a API mais apropriada para a programação das duas aplicações do programa Colombo (ProPEditor e ProCatalog). Estas aplicações, direcionadas para o *software* de CAD Pro/ENGINEER Wildfire 3.0, visam aumentar a produtividade dos projetistas das equipas de Engenharia de Produto, Engenharia Industrial, e de Inovação e Desenvolvimento de Portugal, Índia e Argentina que

são geridas pela unidade de AMT. Na Efacec, o Pro/ENGINEER está ligado ao *software* Windchill para permitir a gestão dos modelos e as aplicações ProPEditor e ProCatalog devem possuir funcionalidades relacionadas com o Windchill. Neste sentido, torna-se imprescindível estudar e compreender o sistema de desenvolvimento de produto (PDS) utilizado na empresa no qual está incluído o Pro/ENGINEER e o Windchill.

O PDS, fabricado pela PTC, reúne quatro famílias principais de produtos: Creo (abrange produtos de desenho que eram conhecidos como Pro/ENGINEER, CoCreate e ProductView), Windchill (*software* para gestão de conteúdos e processos), Arbortext (*software* para entrega dinâmica de informação) e Mathcad (*software* para efetuar cálculos de engenharia) [PTC, 2012d].

As aplicações deste sistema funcionam numa única plataforma, havendo interoperabilidade entre si e com outros sistemas (ERP, etc.), ou em outras palavras, comunicam entre si e com outros sistemas de forma transparente. Todas elas têm ligação à internet e podem funcionar de forma isolada ou como um sistema integrado [PTC, 2012c].

Dos produtos que constituem o PDS da PTC apenas dois são utilizados, de forma integrada, na Efacec: o Pro/ENGINEER Wildfire 3.0, pacote de *software* integrado 3D CAD/CAID/CAM/CAE que permite desenhar os modelos; e o Windchill, mais especificamente o módulo PDMLink 9.0, que armazena esses modelos numa base de dados garantindo a sua integridade e monitorizando, controlando e registando todas as modificações efetuadas [PTC, 2007b], e o módulo ProjectLink 9.0, que permite um desenvolvimento dos produtos de forma colaborativa.

As tecnologias CAD (*Computer Aided Design*) e CAM (*Computer Aided Manufacturing*) dependem da computação gráfica [Asthana and Sinha, 1996], tal como a tecnologia CAE (*Computer Aided Engineering*), e podem ser usadas em várias áreas como medicina, publicidade, criação de jogos, indústria, engenharia, etc. [Oliveira e Silveira, 2006]. Segundo a ISO, a computação gráfica pode ser definida como um conjunto de “métodos e técnicas de converter dados para um dispositivo gráfico, via computador” [Oliveira e Silveira, 2006].

O CAE engloba todas as utilizações do computador em aplicações de engenharia, sendo uma tecnologia utilizada para avaliar e melhorar o desempenho de um produto através da simulação, análise (análise de movimento, térmica, entre outras), diagnóstico e reparação [Garcia S., 2010]. O CAE possui vários ramos, entre os quais o CAD e o CAM [Daintith J., 2005].

3.2 Pro/ENGINEER Wildfire 3.0

De uma forma geral, o CAD refere-se ao uso da tecnologia do computador para o desenho dos produtos e o CAM diz respeito ao uso do computador para traduzir essas modelações às máquinas responsáveis pela manufatura dos mesmos [Daintith J., 2005]. Enquanto o CAD “implementa o rigor e a disciplina da engenharia” [Poter C., 2000], o CAID (*Computer Aided Industrial Design*) é uma adaptação do CAD para ser utilizado por “pessoas criativas que precisam de liberdade para trabalhar os contornos e as formas dos modelos” [Poter C., 2000], estando mais orientado para o *design* estético [Westin S., 1998].

De acordo com o supracitado, para a programação das aplicações *Web ProPEditor*, *ProCatalog* e *MassProp* é fundamental compreender os conceitos do Pro/E e Windchill que estão relacionados com estas aplicações. Enquanto os conceitos mais gerais como os tipos de objetos e características do Pro/E, arquitetura do Windchill e a sua integração com o Pro/E são abordados neste capítulo, os conceitos mais específicos de cada uma das aplicações serão explorados nos capítulos correspondentes às aplicações desenvolvidas (Capítulos 6 ProPEditor, 7 ProCatalog e 8 MassProp).

3.2 Pro/ENGINEER Wildfire 3.0

O Pro/ENGINEER, atualmente designado por Creo Parametric [PTC, 2012e], é um pacote de *software* integrado 3D CAD/CAID/CAM/CAE, desenvolvido pela empresa PTC, que permite desenhar esboços em 2D e produtos em 3D utilizando as melhores práticas de design, sendo compatível com os padrões utilizados na indústria e empresas [PTC, 2012a].

Em termos de arquitetura o Pro/ENGINEER Wildfire 3.0 é uma aplicação que executa no lado do cliente, tendo um *browser* embutido que pode ser ocultado, e é gerida por um *software* de licenças da empresa Flexera Software⁵, o FlexNet [PTC, 2006c].

3.2.1 Tipos de Modelos

No Pro/ENGINEER Wildfire 3.0 o utilizador pode criar ficheiros de diversos tipos: *part* (peça), *assembly* (montagem) ou *drawing* (desenho). Cada um destes tipos tem a extensão “.prt”, “.asm” e “.drw” respetivamente. Existem outros tipos de objetos que estão disponíveis, como por exemplo os *sketches* (esboços), que são usados para fazer esboços geométricos do objeto a

⁵ Flexera Software, <http://www.flexerasoftware.com>

modelar, contudo estes não são relevantes para esta dissertação uma vez que as aplicações desenvolvidas durante o estágio estão unicamente direcionadas para objetos do tipo *part*, *assembly* ou *drawing*.

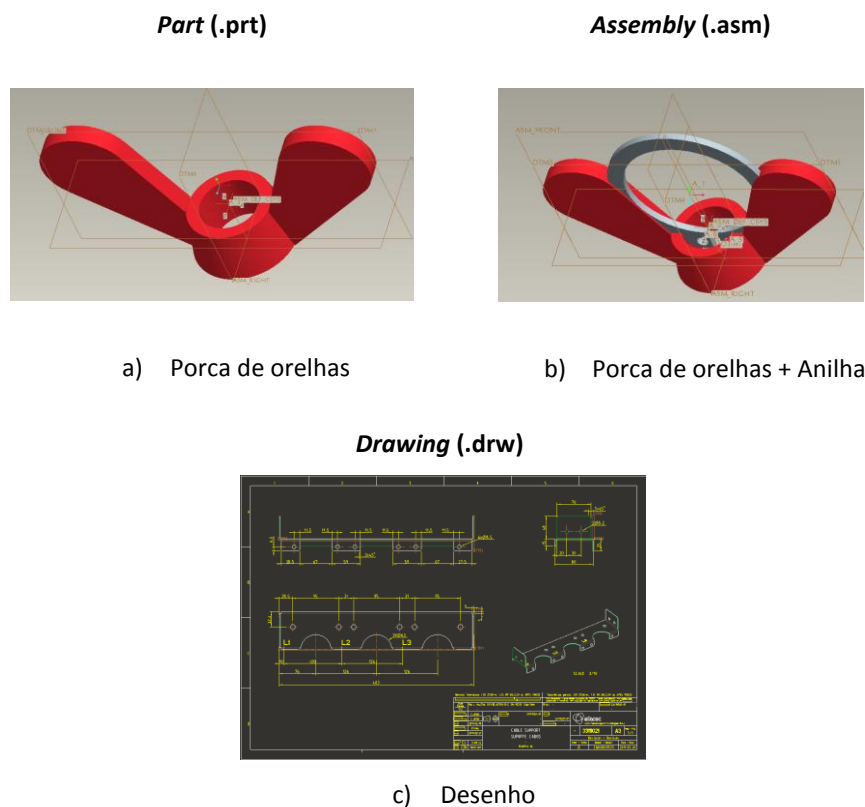


Figura 22 - Tipo de objetos do Pro/ENGINEER

Como indica a figura acima (*Figura 22*), enquanto um modelo *part* é um componente, o modelo *assembly* é uma montagem que une vários componentes. Os desenhos são representações 2D dos modelos 3D (*parts* e *assemblies*) e seguem as normas e convenções do desenho técnico mecânico estabelecidas pela ISO. Estes desenhos contêm informações dos modelos 3D, tais como as dimensões, entre outras notas.

O Pro/E permite abrir várias janelas ao mesmo tempo e cada janela contém um modelo, mas o utilizador apenas pode trabalhar numa janela de cada vez. Para isso é preciso ativá-la e a partir daí o modelo pode ser modificado.

3.2.2 Características

O Pro/E foi lançado pela primeira vez ao público em 1988, tendo sido o primeiro a introduzir no mercado o conceito de modelação paramétrica, modelação associativa, modelação baseada em características (*features*) e modelação de sólidos [PTC, 2012b].

A abordagem de modelação de sólidos do Pro/E assegura que um modelo tem propriedades de um objeto sólido, nomeadamente propriedades de massa como o volume, a área, etc.

3.2.2.1 Modelação Paramétrica

Os modelos são constituídos por características (furos, cortes, saliências, etc.) que estão inter-relacionadas e que têm atributos (dimensão, profundidade, etc.). A modelação paramétrica assegura que a modificação de um atributo de uma dessas características afeta todo o modelo [Kogent Learning Solutions, Inc.,2011]. Por exemplo, se um utilizador estabelecer que dois furos têm de estar alinhados a uma distância de 1 cm e se diminuir o tamanho de um dos furos essa relação mantém-se.

3.2.2.2 Modelação Associativa

No Pro/E os modelos *part*, *assembly* e *drawing* são associativos, isto é, qualquer modificação feita num objeto de um destes modelos reflete-se nos restantes modelos onde o objeto está presente, contribuindo para a sua consistência [Kogent Learning Solutions, Inc.,2011].

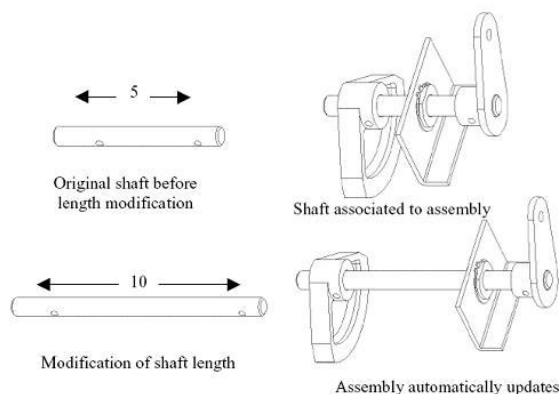


Figura 23 - Impacto da modificação do comprimento de uma peça no *Assembly*

Neste caso o utilizador alterou o comprimento da peça (*part*) e o *assembly* onde esta peça estava aplicada atualizou essa informação.

3.2.2.3 Modelação Baseada em Características

A modelação no Pro/E é baseada em características, ou seja, os modelos são construídos através da adição de características [Kogent Learning Solutions, Inc.,2011]. As várias características que compõem o modelo podem ser apresentadas, de forma ordenada e estruturada, na secção área *model tree* do Pro/E.

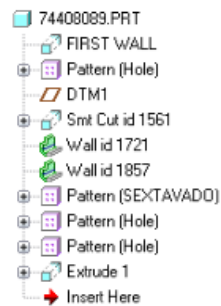


Figura 24 - Model Tree do Pro/ENGINEER

Se um modelo for um *assembly* também podem ser visualizados na *model tree* os vários modelos do tipo *part* ou *assembly* pelo qual este é composto. O utilizador pode manipular (editar, renomear, apagar, entre outras opções) as características ou os modelos e alterar a sua ordem a partir da *model tree*. Uma simples alteração da ordem das características pode modificar o aspeto de um objeto uma vez que estas estão relacionadas entre si.

3.2.3 APIs

A PTC disponibiliza com o Pro/ENGINEER Wildfire 3.0 três APIs proprietárias para estender, automatizar e personalizar as funcionalidades do Pro/E: Pro/Toolkit, J-Link, e Pro/Web.Link. A Pro/Toolkit é a única que requer uma licença adicional para além da licença base do Pro/E, possibilitando o desenvolvimento de programas com a linguagem C ou C++. Essa licença apenas é necessária para instalar no ambiente de desenvolvimento uma vez que a Pro/Toolkit

3.2 Pro/ENGINEER Wildfire 3.0

permite desbloquear a aplicação para ser distribuída pelas várias máquinas de trabalho dos colaboradores [PTC, 2006a]. Com o lançamento do Pro/ENGINEER Wildfire 4.0 surgiu uma nova API gratuita: VB [PTC, 2012f]. À exceção da Pro/Toolkit, todas as APIs são ferramentas de programação orientadas a objetos [Felco Solutions, Inc., 2011].

Uma API (*Application Programming Interface*) pode ser expressa como “uma interface bem-definida que define o serviço que um componente, módulo, ou aplicação fornece a outros elementos de *software*” [Daughtry et al., 2009]. Utilizando a linguagem adequada, uma aplicação pode invocar funções de uma API para interagir com o Pro/E.

Cada uma das APIs do Pro/ENGINEER disponibiliza uma ou, no caso da J-Link, duas bibliotecas com classes e funções que podem ser utilizadas por outras aplicações. As APIs incluem também a seguinte documentação: um documento de notas, que contém as mudanças efetuadas na API pela PTC, nomeadamente as classes, propriedades ou métodos que foram substituídos, alterados ou criados (ex: *Weblink_Wildfire30_RelNotes.pdf*); um manual de utilização com alguns conceitos da API, designadamente aspetos relacionados com a configuração da mesma, explicando ao leitor como obter determinados objetos utilizando as suas classes (ex: *weblinkug.pdf*); exemplos práticos de aplicações desenvolvidas com a API (ex: pasta *weblinkexamples*); e uma *API Wizard* (ex: *weblinkdoc\index.html*), para ser consultada a partir de um *browser*, com uma lista e descrição das classes e das suas propriedades, métodos e relações hierárquicas.

3.2.3.1 Tabela Comparativa

Na Tabela 2 estão sintetizadas as características principais das três APIs J-Link, VB e Pro/Web.Link do Pro/ENGINEER Wildfire 3.0 e 4.0. Apesar da API VB apenas estar disponível para a versão Wildfire 4.0, esta não foi descartada do estudo uma vez a Efacec estava disposta a efetuar um *upgrade* ao Pro/ENGINEER na hipótese desta API vir a ser selecionada. De notar que a API Pro/Toolkit não é comparada pois não é objeto de interesse para a Efacec devido aos seus custos de licenciamento. Esta tabela foi elaborada fundamentalmente para que o departamento de SI da Efacec conhecesse também as restantes APIs. Com esta tabela não se pretende auxiliar totalmente outros projetos futuros uma vez que, dependendo do contexto da aplicação a implementar, pode ser necessário identificar e analisar outro tipo de características.

Tabela 2 – Características das APIs J-Link, VB e Pro/Web.Link

Características	J-Link	VB	Pro/Web.Link
Licença Gratuita			
API do Pro/ENGINEER Wildfire 3.0			
Linguagens de Programação Suportadas	Java	VBA, Visual Basic 2005, JavaScript, VBScript, C++, C#	JavaScript
Aplicações Síncronas			
Aplicações Assíncronas			
Aplicações Web			
Browser	Interno/Externo	Interno/Externo	Interno
Aplicações Web síncronas no Browser Interno do Pro/E			
Aplicações Standalone			
Integração com Aplicações Síncronas J-Link			
Integração com Aplicações Web desenvolvidas com as APIs J-Link, VB e Pro/Web.Link			

a) Licença Gratuita e API do Pro/ENGINEER Wildfire 3.0

As APIs J-Link, VB e Pro/Web.Link são gratuitas [Felco Solutions, Inc., 2011] e a VB surgiu somente com a chegada do Pro/ENGINEER Wildfire 4.0, pelo que não é possível com esta API programar aplicações para versões anteriores ao Wildfire 4.0 [PTC, 2009b].

b) Linguagens de Programação Suportadas

A API J-Link é composta por duas bibliotecas JAR (*Java Archive*): *pfc.jar*, para compilar aplicações síncronas; e *pfcasync.jar*, para compilar aplicações assíncronas. As classes fornecidas por estas bibliotecas podem ser acedidas usando a linguagem Java [PTC, 2006e].

A API VB é composta por uma biblioteca implementada num servidor COM executável (*pfclscom.exe*) [PTC, 2009b]. Este servidor COM, objeto que disponibiliza serviços a clientes, é do tipo *out-process* [MSDN, 2012a]. Enquanto um servidor COM *out-process* executa num processo independente da aplicação cliente, isto é, da aplicação que acede aos seus serviços, um servidor COM *in-process* executa no mesmo processo da aplicação cliente (referência x). As classes fornecidas por esta biblioteca podem ser acedidas usando as linguagens VBA (*Visual Basic Application*), *Visual Basic.NET 2005* e outras linguagens com capacidade para usar servidores COM (*JavaScript*, *VB Script*, *C++*, *C#*) [PTC, 2009b].

A API Pro/Web.Link é composta por uma biblioteca implementada num servidor COM (*pfcscom.dll*). Este servidor COM é um controlador *ActiveX* do tipo *in-process* [MSDN, 2012a][PTC, 2006b]. Para aceder às classes desta biblioteca pode ser utilizada a linguagem *JavaScript* [PTC, 2006b].

c) Aplicações Síncronas e Aplicações Assíncronas

As aplicações podem comunicar de forma assíncrona ou síncrona com o Pro/E para obter informações (ex: qual o tipo de objeto que está ativo na sessão do Pro/E) ou para lhe fornecer instruções (ex: alteração de um valor de um parâmetro de um modelo). Neste contexto, enquanto as aplicações assíncronas podem realizar operações concorrentes e executam num processo independente do Pro/E, podendo conectar-se a um processo do Pro/E existente ou gerar um novo processo do Pro/E e conectar-se a ele, as aplicações síncronas não podem executar operações concorrentes e necessitam que o Pro/E esteja a ser executado para conseguirem iniciar [PTC, 2009b] [PTC, 2006e].

Entre as três APIs, a J-Link é a única que permite desenvolver aplicações síncronas e assíncronas. Como meios de comunicação entre as aplicações assíncronas e o Pro/E, a J-Link utiliza o *JNI (Java Native Interface)* e o *RPC (Remote Procedure Calls)*. Na construção destas aplicações síncronas deve haver um especial cuidado para não interferir com o *thread* principal do Pro/E porque todas os *threads* criados estão dentro do mesmo processo do Pro/E. Assim, as caixas de diálogo construídas em aplicações Java síncronas devem ser do tipo

modal (ou bloqueante) para bloquear possíveis interações do utilizador com o Pro/E até que esta seja fechada e impossibilitando assim interferências entre *threads*. Esta API também tem a particularidade de possibilitar a associação de uma aplicação síncrona a um determinado modelo *part* ou *assembly*. Desta forma, quando um determinado modelo *part* ou *assembly* são carregados para a sessão do Pro/E o método *start* é ativado e quando o modelo é eliminado da memória o método *stop* é ativado. As aplicações assíncronas criadas com esta API podem suportar o modo simples ou *full*. A diferença entre estes dois modos é que no modo *full* a aplicação pode implementar um *loop* para escutar e responder quando determinados eventos são acionados no Pro/E. Nas aplicações assíncronas apenas um *thread* pode comunicar com o Pro/E e por isso não é permitido ter várias ligações à sessão do Pro/E. A sessão do Pro/E é o objeto de mais alto nível, ou seja, para comunicar com o Pro/E é necessário que a aplicação se ligue primeiro à sua sessão [PTC, 2006e].

A API VB suporta exclusivamente aplicações assíncronas, necessitando de conectar-se primeiro ao Pro/E para conseguir interagir com ele. Tal como a API J-Link, a VB também suporta o modo assíncrono simples ou *full*, e como apenas uma *thread* pode comunicar com o Pro/E, as aplicações assíncronas apenas devem conectar-se a uma sessão do Pro/E se não houver outra conexão ativa uma vez que quando se efetua novamente uma conexão à sessão do Pro/E a partir da aplicação, a última ligação é perdida e é realizada uma nova conexão [PTC, 2009b].

Ao contrário da VB, a API Pro/Web.Link suporta apenas aplicações síncronas e para que estas aplicações consigam comunicar com o Pro/ENGINEER é necessário que este esteja a ser executado [PTC, 2006b].

d) Aplicações Web

Todas estas APIs possibilitam a construção de aplicações *Web* uma vez que suportam linguagens que assim o permitem: a linguagem Java, suportada pela API J-Link, possui a classe Servlet que permite gerar código HTML (*HyperText Markup Language*) [Conover S., 2005] [PTC, 2006e]. A API VB suporta a linguagem VBScript e JavaScript [PTC, 2009b]; e a API Pro/Web.Link, construída exclusivamente para o desenvolvimento de aplicações *Web*, suporta o JavaScript que tal como a VBScript é uma linguagem de script que pode ser embebida no código HTML [PTC, 2006b].

e) Browser e Aplicações Web Síncronas no browser interno do Pro/E

À exceção das aplicações *Web* criadas com a API Pro/Web.Link, que apenas conseguem executar dentro do *browser* interno do Pro/E, possuindo mecanismos de restrição implementados pela PTC [PTC, 2006b], as aplicações *Web* criadas com a API J-Link e VB também podem ser executadas num *browser* externo desde que sejam assíncronas pois têm a capacidade de gerar um processo do Pro/E ou de se ligar a um processo existente do Pro/E [PTC, 2009b] [PTC, 2006e]. Apenas as APIs J-Link e Pro/Web.Link permitem desenvolver aplicações *Web* síncronas que executem no *browser* interno do Pro/E. Esta última característica está relacionada com os requisitos das aplicações ProPEditor e ProCatalog.

f) Aplicações Standalone

A API Pro/Web.Link é a única que não permite desenvolver aplicações *standalone* uma vez não possui mecanismos para ligar uma aplicação *standalone* ao Pro/E, sendo obrigatório que a aplicação seja executada dentro do *browser* interno do Pro/E [PTC, 2006b].

g) Integração com Aplicações Síncronas J-Link

As APIs VB e Pro/Web.Link possuem nas suas bibliotecas um método para executar aplicações síncronas desenvolvidas com a API J-Link: *StartJLinkApplication()*. Para disponibilizar estas aplicações é necessário registá-las utilizando uma funcionalidade da API J-Link. A API J-Link também possui este método, o que torna possível a integração de aplicações assíncronas com aplicações síncronas [PTC, 2006b][PTC, 2009b] [PTC, 2006e].

h) Integração com Aplicações Web desenvolvidas com as APIs J-Link, VB e Pro/Web.Link

As APIs J-Link, VB e Pro/Web.Link possuem nas suas bibliotecas um método para executar aplicações *Web* a partir do *browser* interno do Pro/E: *SetURL()*. Como tal, cada uma destas APIs pode invocar servlets desenvolvidos com a API J-Link, ou páginas *Web* desenvolvidas em VBScript ou JavaScript com a API VB ou em JavaScript com a Pro/Web.Link. Para invocar aplicações desenvolvidas com a API VB é necessário utilizar o Pro/ENGINEER Wildfire 4.0 uma vez que esta API apenas surgiu com esta versão [PTC, 2006b][PTC, 2009b] [PTC, 2006e].

A Tabela 2 reúne as principais características de cada uma das APIs. Outros fatores distinguem as APIs tais como o nível de dificuldade da aprendizagem, de deployment e o tempo de desenvolvimento. Contudo, estes fatores não são avaliados e apresentados nesta tabela devido à sua subjetividade uma vez estão relacionados não só com a complexidade das APIs,

como também com os conhecimentos do programador em relação às linguagens de programação suportadas.

Através da manipulação de pequenos exercícios exemplo da PTC por parte do autor, constatou-se que a Pro/Web.Link é uma API bastante acessível em termos de linguagem e das funções que disponibiliza. O tempo de desenvolvimento de pequenas aplicações com a Pro/Web.Link foi menor aquando comparado com a utilização da VB ou da J-Link. Comparando a API VB e a J-Link, considerou-se que a J-Link não é tão acessível como a VB dado que tem um processo de deployment um pouco mais complexo. A API VB não fica aquém da Pro/Web.Link. Embora estas observações sejam partilhadas por outros programadores, inclusive pelo Christopher Lewis, um Consultor Sénior da PTC Global Services [Lewis C., 2009], cada API tem características próprias e a sua seleção deve ser feita com base nos objetivos que se pretendem atingir e não com o grau de dificuldade da mesma.

3.2.3.2 Processo de Seleção e Decisão Final

Para desenvolver as aplicações ProPEditor e ProCatalog foi necessário selecionar uma das três APIs do Pro/ENGINEER Wildfire 3.0 e Wildfire 4.0. Para tal, iniciou-se o processo de seleção e sugeriu-se a API mais adequada. No final do processo, o departamento de SI decidiu se a API sugerida era uma boa escolha.

O processo de seleção foi dividido em cinco fases. Na primeira fase foram nomeados todos os requisitos da API com base nos requisitos das aplicações ProPEditor e ProCatalog. A segunda fase consistiu na seleção temporária da API, com o auxílio da *Tabela 2*, que satisfazia os requisitos identificados anteriormente e que, ao mesmo tempo, possuía vantagens em relação a outras APIs. É importante nesta fase identificar e analisar outros elementos relevantes, não menosprezando os conhecimentos do programador em relação às linguagens de programação. Na terceira fase foram definidos e descritos os casos de uso das aplicações ProPEditor e ProCatalog para identificar as funcionalidades do Pro/E que devem ser implementadas com a API. Na quarta fase foi realizado um estudo à documentação da API, previamente selecionada na segunda fase, para verificar se esta suporta as funcionalidades identificadas na alínea anterior. Na quinta fase sugeriu-se ao departamento de SI, com base nas fases anteriores, a API mais adequada para implementar as aplicações ProPEditor e ProCatalog. Caso a API selecionada na segunda fase não permitisse implementar todas as

funcionalidades do Pro/E identificadas, seria necessário ou repetir o processo de seleção a partir dessa fase, selecionando outra API que permitisse igualmente satisfazer todos os requisitos, ou ponderar a possibilidade de desenvolver uma determinada aplicação com diferentes APIs uma vez que existe interoperabilidade entre elas. Por exemplo, a VB API e Pro/Web.Link contêm métodos para executar aplicações síncronas do J-Link [PTC, 2006b][PTC, 2009b].

Fases do processo de seleção:

a) Nomeação de todos os requisitos da API

A API deve permitir desenvolver aplicações com as seguintes características:

- Aplicações *Web-based* síncronas;
- Aplicações que comuniquem com o Pro/E somente a partir do *browser* interno, por questões de segurança;
- Aplicações que utilizem uma linguagem de programação que possa ser integrada com a linguagem PHP. Isto porque as aplicações devem integrar a intranet, utilizando o seu sistema de *login* e de traduções, e o acesso às bases de dados e o desenho das interfaces deve ser feito com recurso ao *Framework* da Efacec, o que só é possível utilizando a linguagem PHP.

b) Seleção, com base na tabela, da(s) API(s) que satisfaz(em) os requisitos identificados na alínea a

Com base nos requisitos anteriores e na linguagem de programação, selecionou-se a API Pro/Web.Link. Esta API foi desenvolvida exclusivamente pela PTC para a construção de páginas *Web*, permitindo desenvolver páginas *Web* síncronas que utilizam a linguagem JavaScript. Suportando esta linguagem as aplicações podem ser facilmente integradas com a linguagem PHP. A API VB foi imediatamente descartada pois não permite o desenvolvimento de aplicações síncronas. A API J-Link foi colocada temporariamente de parte uma vez que a API Pro/Web.Link satisfaz todos os requisitos e suporta uma linguagem mais acessível (JavaScript) tanto para o autor como para os restantes programadores do departamento de SI, o que facilitará mais tarde o processo de atualização das aplicações. Caso numa fase posterior se verificasse que a Pro/Web.Link não conseguia implementar todas as funcionalidades requeridas das aplicações então seria necessário analisar a API J-Link e verificar se é possível integrar o PHP com o Java, mas isso não aconteceu e como tal não foi necessário fazer esse estudo.

c) Identificação das funcionalidades das aplicações a implementar com a API

Nesta etapa, foram identificados e descritos os casos de uso das aplicações ProPEditor e ProCatalog para detetar todas as funcionalidades do Pro/E que eram necessárias implementar. Algumas das funcionalidades identificadas: obter o modelo genérico a partir de uma instância; obter e editar os valores dos parâmetros de um modelo; obter e editar os valores dos campos de um desenho; verificar se um determinado modelo está *checked out*. A descrição final dos casos de uso pode ser consultada nas secções Apêndice E - Caso de Uso Editar Legenda e Apêndice F - Caso de Uso Adicionar Artigo.

d) Estudo da documentação da(s) API(s), previamente seleccionadas na alínea a, para verificar se esta(s) suporta(m) as funcionalidades identificadas na alínea anterior

Nesta etapa, através do estudo da documentação da API Pro/Web.Link, a API seleccionada na alínea anterior, verificou-se que esta permite implementar todas as funcionalidades do Pro/E requeridas.

É de referir que cada uma das três APIs é limitada em relação à API Pro/Toolkit, que possui cerca de 80/90% das funcionalidades do Pro/E: a J-Link tem cerca de 60%; a Pro/Web.Link com 50%; e a VB possui perto de 40% das funcionalidades [Felco Solutions, Inc., 2011].

e) Sugestão da API

A API sugerida pelo autor foi a Pro/Web.Link uma vez que satisfaz todos os requisitos das aplicações do programa Colombo, permitindo implementar todas as funcionalidades do Pro/E requeridas, e suporta uma linguagem mais acessível (JavaScript) para o autor e para os membros do departamento de SI, não constituindo uma barreira para futuras atualizações. Perante estas razões, o departamento de SI concordou que a **API Pro/Web.Link** é a mais adequada para este projeto.

3.3 Windchill

O Windchill é um PLM, desenvolvido pela PTC, que faz a gestão do ciclo de vida dos produtos [PTC, 2012f]. A gestão do ciclo de vida dos produtos envolve uma gestão integrada dos processos de negócio e das informações relacionadas com os produtos, desde o seu planeamento e produção até à sua utilização, manutenção e descarte [Zancul E., 2009].

3.3 Windchill

Segundo Hélio Samora, diretor da PTC da América Latina, “o Windchill aumenta a eficiência da equipe de engenheiros na medida em que elimina os erros associados a dados duplicados ou incompletos, por dispor um repositório único e seguro para armazenar todo o conteúdo relativo ao produto” [Civa G, 2010].

A PTC disponibiliza vários componentes do Windchill para fazer a gestão dos conteúdos, dos dados dos produtos e dos projetos. Atualmente a Efacec utiliza os componentes PDMLink 9.0 e o ProjectLink 9.0 que funcionam de forma integrada sob a forma de aplicação *Web* e possuem um sistema de *login*.

3.3.1 Windchill PDMLink 9.0 (Product Data Management)

“O Windchill PDMLink é um sistema de gestão de dados do produto que gere toda a informação relacionada com o desenvolvimento do produto, otimizando os processos de negócio ao longo da sua vida” [PTC, 2007a].

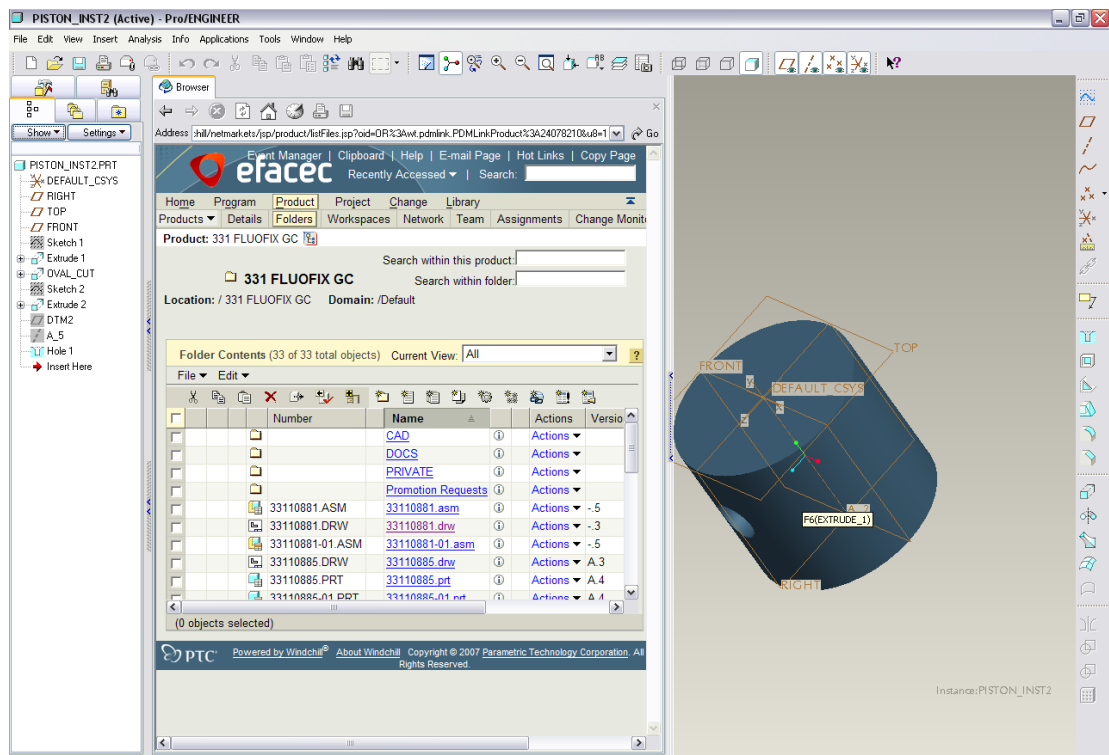


Figura 25 - Módulos PDMLink 9.0 e ProjectLink 9.0 no navegador embutido do Pro/E

O Windchill PDMLink pode interagir com o Pro/ENGINEER para efetuar a gestão dos modelos, controlando os acessos aos mesmos. Esta interação do Windchill com o Pro/E é suportada se o Windchill PDMLink estiver a ser executado no *browser* embutido do Pro/E [PTC, 2007b] .

Os objetos do Windchill podem ser, entre outros, documentos CAD [PTC, 2007a] que, tal como no Pro/E, podem ser do tipo “*part*”, “*assembly*” e “*drawing*”. Estes objetos podem estar num dos seguintes estados: *Design* (em projeto/desenho); *Prototype* (em protótipo); *Under Review* (à espera de aprovação); *Production Change* (em alteração); *Released* (aprovado e pronto para ser utilizado em produção); *Obsolete* (não é/não deve ser utilizado).

Estes objetos do tipo *part*, *assembly* ou *drawing* podem ser controlados por uma versão, permitindo ao utilizador rever todas as modificações feitas a esse mesmo objeto. A versão é constituída pela revisão e iteração.

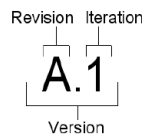


Figura 26 - Identificação dos elementos da versão de um objeto [PTC, 2007a]

Embora as revisões possam ser personalizadas, por omissão estas são representadas por uma letra do alfabeto (A, B, C) e são criadas sempre que for detetado e corrigido algum erro no objeto ou for necessário realizar uma alteração para o adequar a um novo produto. As iterações são compostas por números inteiros e quando o utilizador faz *check in* ao objeto a iteração é automaticamente incrementada [PTC, 2007a].

3.3.1.1 Arquitetura do Windchill PDMLink

Ao nível da arquitetura, o Windchill PDMLink é uma aplicação multicamada baseada na Web e escrita totalmente em linguagem Java utilizando a plataforma J2EE (**Java2 Platform Enterprise Edition**).

3.3 Windchill

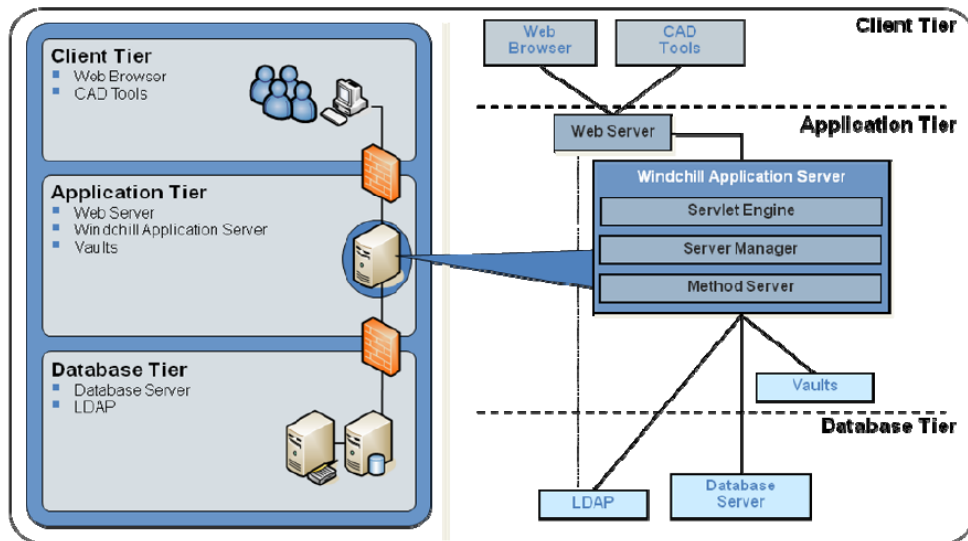


Figura 27 - Arquitetura multicamada do Windchill PDMLink [VMware, 2008]

O Windchill PDMLink pode ser implementado num único servidor ou, para responder a pedidos mais complexos, em múltiplos servidores. Habitualmente a sua arquitetura física é representada por três camadas (*three-tier*) como demonstra a *Figura 27* [VMware, 2008][Cisco, 2009].

As três camadas que compõem o Windchill PDMLink são:

- **Camada Cliente/Apresentação:** Onde o utilizador interage, através de um *browser*, com o servidor *Web* do Windchill [Melodie S., 2009]. O protocolo de comunicação entre o *browser* e o servidor *Web* da camada de aplicação é principalmente o HTTP (*HyperText Transfer Protocol*) ou HTTPS (*HyperText Transfer Protocol Secure*). Os protocolos RMI (*Remote Method Invocation*), RMI sobre HTTP ou RMI sobre HTTPS também podem ser utilizados. O Pro/E está presente nesta camada e, através do seu *browser* embutido, pode interagir com o servidor *Web* da camada aplicação do Windchill PDMLink utilizando o protocolo SOAP (*Simple Object Access Protocol*) sobre HTTP (HTTPS) [Cisco, 2009];
- **Camada Aplicação:** Composta pelo servidor *Web* (Apache) e pelo servidor aplicacional, que contém a lógica de negócio e a interface para integração com outros sistemas (ERP) [Cisco, 2009].

O servidor aplicacional é formado pelos seguintes componentes:

- *Servlet engine* (Tomcat) : faz a gestão de servlets e de JSP (*Java Server Pages*) e filtra os pedidos do Apache para o *method server* para processamento [VMware, 2008];
- *Server manager*: responsável por gerir o *method server*, as credenciais da sessão do utilizador, etc.;
- *Method server*: responsável pelo processamento das transações do cliente, bases de dados, etc. [Melodie S., 2009]

Para ligar a camada da aplicação à camada de dados é utilizado a API JDBC [Cisco, 2009].

- **Camada Dados:** Os dados são guardados num Sistema de Gestão de Bases de Dados Relacional (Oracle ou SQL Server).

3.3.1.2 Integração do Windchill PDMLink com o Pro/ENGINEER

Como se verificou na subsecção anterior, o Pro/E e o servidor *Web* da camada aplicação do Windchill PDMLink podem interagir entre si fazendo pedidos SOAP sob HTTP ou HTTPS através do *browser* embutido do Pro/E. Estes pedidos são feitos para fazer, por exemplo, *upload* de modelos, obter uma página HTML do Windchill, realizar operações às bases de dados do Windchill [PTC, 2007b].

Para que o Pro/E e o Windchill PDMLink consigam interagir entre si é necessário estabelecer uma ligação entre eles através do registo e ativação do servidor do Windchill (*PDM Server*) no Pro/E e da indicação do *workspace* ativo do lado do servidor (*Figura 28*). Neste contexto, o PDM Server contém a camada aplicação e a camada dados. O *workspace* do lado do servidor (*server-side workspace*) é um espaço do Windchill definido pelo utilizador para armazenar os modelos a editar no Pro/E.

Após a identificação do servidor e da escolha do *workspace*, através da opção “Server Registry” do menu “Tools” do Pro/ENGINEER (*Figura 28*), estes ficam ativos no Pro/E e a partir

3.3 Windchill

deste instante o utilizador pode efetuar várias operações aos modelos CAD, tais como a abertura dos modelos a partir do *workspace* do Windchill, o *check-out*, o *check-in*, etc.

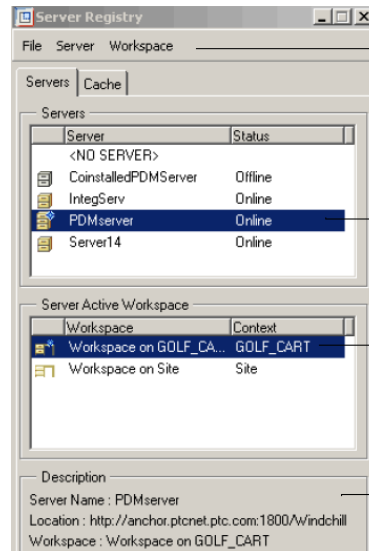


Figura 28 - Opção *Server Registry* do menu *Tools* do Pro/ENGINEER [PTC, 2007b]

A opção *Server Registry* do Pro/E (Figura 28) também permite saber qual o nome, o estado (*online* ou *offline*) e a localização do servidor ativo no Pro/E, e qual o *workspace* ativo, podendo alterar essas definições quando entenderem.

O Windchill tem de estar preparado para gerir grandes quantidades de dados e manter a integridade destes controlando as operações concorrentes. Assim, quando o Windchill PDMLink está integrado com o Pro/E quatro locais de armazenamento distintos compõem o *PDM Server* do Windchill e o lado cliente onde o Pro/E está a ser executado.

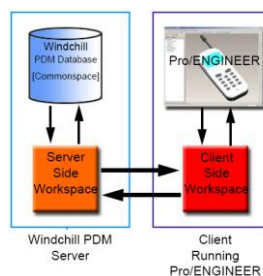


Figura 29 - Fluxo da comunicação entre o Windchill e o Pro/ENGINEER [PTC, 2007b]

Enquanto o *Windchill PDM Server* é composto pelo *commonsplace* do *Windchill* e pelo *server-side Workspace*, o cliente é composto pelo *cliente-side workspace cache* e pela sessão do Pro/E [PTC, 2007b]:

- **Shared folder (commonsplace):** espaço do *Windchill*, comum a múltiplos utilizadores que tenham privilégios de acesso, que contém todos os modelos CAD da empresa (*Figura 25*);
- **Workspace (Server-side workspace e client-side workspace):** espaço pessoal de cada utilizador criado especificamente para manter cópias dos modelos que cada um pretende editar. Cada utilizador pode criar vários *workspaces*. Cada vez que é criado um *workspace* é criado um *server-side workspace* e um *client-side workspace*.

O *Workspace* divide-se em dois lugares de armazenamento:

- **Server-side workspace:** Está situado no lado do servidor e pode conter por exemplo: modelos que foram enviados do Pro/E através da opção “save & upload”; modelos da área de partilha que foram adicionados a esta área, através da funcionalidade “Add To Workspace”, cujas permissões são apenas de leitura até o utilizador fazer check out.
- **Client-side workspace cache:** Local situado no lado do cliente com cópias locais dos objetos que são editados. Esta cópia serve para aprimorar o desempenho do Pro/E e para guardar localmente as modificações que são feitas ao objeto e que são confirmadas através da opção *save* do Pro/E. Esta área interage com o *server-side workspace* para guardar cópias dos modelos no *Windchill*.
- **Sessão do Pro/ENGINEER:** Memória do Pro/E que mantém uma cópia dos modelos que são abertos através de diversas operações (*check-out*, etc.). Se o utilizador fechar os modelos no Pro/E e não apagar a memória (através de uma opção do Pro/E), estes continuam em sessão. Quando o utilizador carrega um modelo *assembly* para o Pro/E, este é guardado em sessão juntamente com os modelos que o constituem (*part e/ou assembly*), ficando apenas visível para o utilizador o modelo *assembly*.

3.3 Windchill

Para gerir os modelos CAD, os utilizadores podem realizar várias operações PDM a partir do Windchill e do Pro/E: *check-out*, *check-in*, etc. Se o utilizador aceder ao *server-side workspace* a partir de um *browser* externo ao Pro/E, este apenas tem acesso a esse *workspace* e ao *commonsplace*, não podendo interagir com o *cliente-side workspace* e sessão do Pro/E. Quando esta interação não é possível, as funções PDM que permitem interagir com o Pro/E não podem ser utilizadas. A figura seguinte demonstra quais os espaços de armazenamento envolvidos nas diferentes operações.

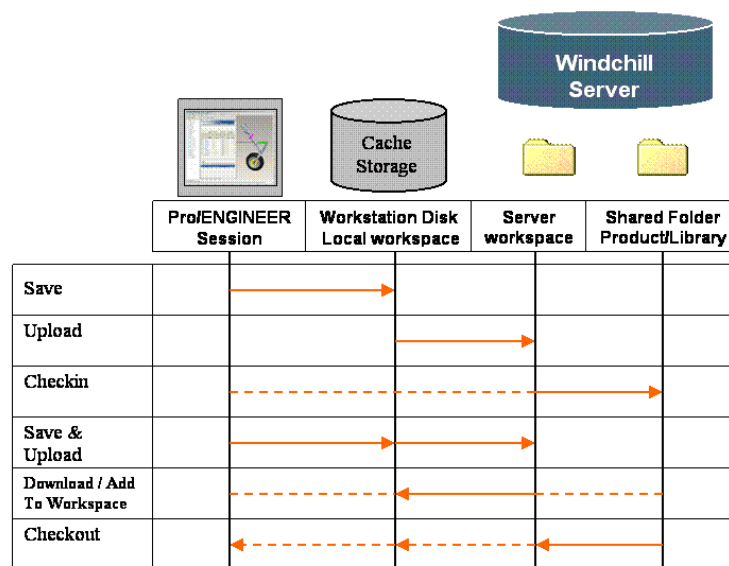


Figura 30 - Operações PDM e locais de armazenamento [PTC, 2009a]

Destacando as operações de *check-out* e *check-in*:

- **Check-Out:** O *check-out* permite editar um objeto no Pro/E, bloqueando-o aos outros utilizadores que ficam somente com permissões de leitura sobre ele. Apenas os utilizadores que têm permissões administrativas podem desfazer a operação de *check-out* que outro utilizador fez. Para que um utilizador consiga fazer *check out* ao objeto tem de alterar primeiro o estado do objeto para “Production Change”.

A operação de *check-out* pode ser efetuada a partir do Pro/E, do *workspace* do lado do servidor ou do *commonsplace*.

Se for efetuado um *check-out* a um modelo que esteja no *commonsplace*, este é bloqueado aos restantes utilizadores e é copiado para um *workspace* do lado do servidor selecionado pelo utilizador. Se o utilizador desejar copiar o modelo para o **workspace** do lado do cliente e abri-lo no Pro/E então deve escolher também a opção *download* durante a operação de *check-out*.

Para libertar o objeto, o utilizador detentor do mesmo pode desfazer a operação de *check-out*, onde as modificações efetuadas ao objeto não são aplicadas e uma nova iteração não é criada, ou fazer *check in* ao objeto, sendo este guardado no Windchill com a nova informação e com uma nova iteração.

- **Check-in:** Quando o utilizador efetuar todas as alterações ao objeto ou este estiver pronto para ser divulgado na área partilhada do Windchill, deve fazer *check in* ao objeto. Após a operação de *check-in*, o objeto é guardado na área partilhada (*commonsplace*), sendo-lhe atribuída uma nova iteração. No *check in*, o utilizador pode optar por manter o *check-out* e nesse caso o objeto permanece bloqueado, de outro modo este é desbloqueado, permitindo que os utilizadores o possam modificar.

O *check-in* pode ser feito a partir do Pro/E, sendo feito um *save* e um *upload* automático juntamente com o *check-in*, ou a partir do *server-side workspace*, que implica que o utilizador tenha feito um *save* e *upload* manual do modelo para esta área [PTC, 2007a].

3.4 Sumário

Neste capítulo foi alcançado o objetivo *Selecionar API do Pro/ENGINEER*, sendo fornecidos dois contributos.

Objetivo

- **Selecionar a API do Pro/ENGINEER:** resumo das principais características das APIs J-Link, VB e Pro/Web.Link para auxiliar o processo de seleção da API mais adequada para o desenvolvimento das aplicações ProPEditor e ProCatalog e de outras aplicações que sejam necessárias implementar no futuro. No processo de seleção foi sugerida, por parte do autor desta dissertação, a API Pro/Web.Link com base nos critérios e funcionalidades das aplicações ProPEditor e ProCatalog. O departamento de SI

3.4 Sumário

aceitou a sugestão uma vez que esta cumpre todos os requisitos das aplicações do programa Colombo, permite implementar todas as funcionalidades do Pro/E e suporta uma linguagem que, quer o autor, quer os membros do departamento de SI, dominam.

Contributos

- Relatório sobre as APIs do Pro/E com informação essencial sobre as suas principais características para suportar o processo de seleção de uma API;
- Documentos com a descrição dos casos de uso das aplicações ProPEditor e ProCatalog e com outras especificações. Até ao final do estágio, estas descrições foram alvo de alterações durante as fases de implementação das aplicações ProPEditor e ProCatalog. As descrições finais são apresentadas nas secções Apêndice E - Caso de Uso Editar Legenda e Apêndice F - Caso de Uso Adicionar Artigo.
- Sugestão da API Pro/Web.Link para o desenvolvimento das aplicações ProPEditor e ProCatalog.

4 Pro/Web.Link

Este capítulo encontra-se dividido em seis secções. A primeira secção **4.1 Introdução** introduz a API Pro/Web.Link, apresentando os conhecimentos que um programador deve ter para desenvolver aplicações com esta API. A segunda secção **4.2 Arquitetura da API Pro/Web.Link** descreve a arquitetura da API Pro/Web.Link de uma forma muito breve. A terceira secção **4.3 Classes** identifica os tipos de classes que compõem a API Pro/Web.Link, sendo fornecidos alguns exemplos para explicar como podem ser utilizadas estas classes. A quarta secção **4.4 Herança** fala sobre a capacidade de herança dos objetos da API Pro/Web.Link. A quinta secção **4.5 Pro/Web.Link API Wizard** fornece algumas noções sobre a API *Wizard*, uma documentação que identifica as classes, métodos e propriedades da API Pro/Web.Link. A sexta secção **4.6 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

4.1 Introdução

A Pro/Web.Link é uma API, desenvolvida pela PTC, que permite que uma aplicação *Web* comunique com o Pro/ENGINEER a partir do seu *browser* interno [PTC, 2006b]. Como o uso do Pro/E está orientado para o sector empresarial e institucional, as aplicações *Web* podem estar inseridas numa intranet.

Para utilizar a API Pro/Web.Link no desenvolvimento das aplicações requer-se que o programador tenha conhecimentos sobre:

- **JavaScript:** para invocar os métodos e propriedades da biblioteca da API Pro/Web.Link;
- **HTML:** para o desenho da interface. Podem ser utilizadas outras linguagens de programação como o PHP uma vez que existe facilidade de integração entre estas duas linguagens;
- **Pro/ENGINEER:** A PTC escreveu o manual de utilizador da Pro/Web.Link pressupondo que o programador conhece as funções do Pro/ENGINEER [PTC, 2006a]. Portanto, é aconselhável que o programador compreenda os conceitos do Pro/E que estão relacionados com as aplicações Web a desenvolver para encontrar mais facilmente os métodos e propriedades capazes de responder adequadamente aos requisitos da aplicação e para, possivelmente, participar na discussão de soluções. Se houver integração do Pro/E com o Windchill (sistema PLM) também é essencial compreender os conceitos abrangidos por este.

4.2 Arquitetura da API Pro/Web.Link

A API Pro/Web.Link é constituída pelo controlador ActiveX *pfcscom.dll* [PTC, 2006b]. O controlador ActiveX é um tipo de servidor de Automatização (*Automation server in-process*) [MSDN, 2012a] [MSDN, 2012b]. Um servidor de Automatização suporta a tecnologia de Automatização, tecnologia baseada no COM (*Component Object Model*) que permite que as aplicações exponham os seus objetos (*Automation objects* ou *ActiveX objects*) a outras aplicações e que estas manipulem esses objetos. O COM é um sistema independente da plataforma de utilização e orientado a objetos que permite desenvolver componentes reutilizáveis, nomeadamente componentes binários que podem interagir com outros objetos [MSDN, 2012c]. Sendo o controlador ActiveX um servidor *in-process* é implementado num ficheiro DLL e é carregado e no mesmo processo do seu cliente de Automatização, isto é, da aplicação que acede aos seus serviços. Neste contexto, o cliente de Automatização é o *browser* interno do Pro/ENGINEER Wildfire 3.0. Embora um *browser* externo também consiga comunicar com o controlador ActiveX, por motivos de restrição da PTC, este não tem permissões para invocar os métodos ou propriedades dos seus objetos.

4.3 Classes

Este controlador ActiveX deve ser registado no Windows para ser carregado, suportando o auto registo e a anulação do mesmo no Windows através da implementação das funções *DllRegisterServer* e *DllUnregisterServer* [MSDN, 2012d].

Este controlador ActiveX foi classificado pela PTC como sendo não seguro [PTC, 2006b] .

Enquanto o mecanismo utilizado no Windows é o ActiveX, no UNIX utiliza-se o XPCOM (*Cross Platform Component Object Model*) que é semelhante ao modelo COM da Microsoft e foi criada pela Fundação Mozilla [Mozilla Developer Network, 2012]. O XPCOM não é utilizado no âmbito desta dissertação uma vez que se pretende estudar apenas a API Pro/Web.Link no contexto do Windows.

4.3 Classes

A API Pro/Web.Link do Pro/ENGINEER Wildfire 3.0 é constituída por classes designadas por PFC (*Parametric Foundation Classes*), que contêm métodos e propriedades. Antes do aparecimento destas classes, a API Pro/Web.Link disponibilizava outros métodos a que a PTC refere como *old 'PWL' style*. Embora a PTC não aconselhe o uso destes métodos em novas aplicações, a API Pro/Web.Link ainda os suporta, comportando-os na classe *pfcScript*, por questões de compatibilidade com antigas aplicações. Para aceder a esta classe é preciso criar um objeto da classe *MpfcCOMGlobal* e utilizar o seu método *GetScript()* [PTC, 2006b].

As classes PFC desta API abrangem 7 tipos de classes principais. À exceção das classes relacionadas com o Pro/ENGINEER, todas as classes podem ser instanciadas com um objeto do JavaScript, o *ActiveXObject*. Uma vez que este objeto apenas pode ser utilizado no *browser* IE e só terá utilidade se for usado no browser interno do Pro/E, pois a PTC implementou uma restrição que não permite o acesso aos métodos e/ou propriedades de um objeto retornado a partir de um *browser* externo ao Pro/E, devem ser feitas validações restringindo a execução do objeto *ActiveXObject* ao *browser* interno do Pro/ENGINEER (IE). As validações podem ser feitas numa função a que a PTC designa por *pfcCreate(className)*, mas o programador pode optar por manter ou não essa denominação. Esta função pode ser consultada na secção *Apêndice D - Módulo JavaScript*.

4.3.1 Pro/ENGINEER – Related

Classes que permitem manipular diretamente os objetos do Pro/ENGINEER [PTC, 2006b], nomeadamente a sua sessão, os seus modelos, parâmetros, etc. As classes *pfcSession*, *pfcModel* e *pfcParameter* são exemplos deste tipo de classes.

Estes objetos não podem ser instanciados diretamente utilizando o objeto *ActiveXObject* do JavaScript mas podem ser retornados através do uso de métodos e propriedades adequadas da API Pro/Web.Link.

O objeto sessão (classe *pfcSession*) é considerado o objeto de mais alto nível dado que é necessário obtê-lo para conseguir interagir com os restantes objetos do Pro/E. Para obter o objeto sessão utiliza-se o método *GetProESession()* sobre o objeto *MpfcCOMGlobal*. Mesmo que este método seja utilizado várias vezes na mesma aplicação, a sessão retornada será sempre a mesma [PTC, 2006b].

4.3.2 Module – Level

Classes que contêm métodos estáticos para criar ou aceder a outras classes. Este tipo de classes não contêm quaisquer atributos acessíveis e o seu nome começa sempre por “M” seguido do nome do módulo [PTC, 2006b].

O objeto *MpfcCOMGlobal* é um exemplo deste tipo de classes. Como foi referido, este objeto é necessário para obter a sessão do Pro/E.

```
//Instanciar um objeto da classe MpfcCOMGlobal
var mGlob = pfcCreate("MpfcCOMGlobal");
//Obter a sessão do Pro/E
var session = mGlob.GetProESession();
```

Código 1 - Extrato de código

Ao contrário das classes relacionadas com o Pro/E, estas classes podem e devem ser instanciadas utilizando a função *pfcCreate(className)* [PTC, 2006a].

4.3.3 Compact Data

Classes criadas para ser usadas como argumentos em alguns métodos do Pro/E. Diferentemente dos objetos do Pro/E, estas classes podem e devem ser instanciadas [PTC, 2006b].

Para instanciar estas classes utiliza-se a função estática *pfcCreate* e em seguida, se aplicável, invoca-se o método construtor definido para a classe em questão que pode ser *Create()* ou pode ter a palavra *create* no seu nome [Lewis C., 2008].

A classe *pfcCheckoutOptions* representa uma classes de dados compactos e esta é, por exemplo, um dos argumentos do método *CheckoutObjects(pfcModel Mdl, string File, boolean Checkout, pfcCheckoutOptions Options)*. Para utilizá-la neste método é necessário criá-la através do código abaixo e definir as propriedades requeridas caso estas possam ser modificadas.

```
//Instanciar um objeto da classe pfcCheckOutOptions
var checkoutOptions = pfcCreate("pfcCheckOutOptions");
//Invocar o método construtor Create()
checkoutOptions = checkoutOptions.Create();
```

Código 2 - Extrato de código

4.3.4 Enumeration

Cada classe de enumeração define uma lista de valores/propriedades para representar determinados tipos de objetos (tipos de parâmetros, tipos de valores de parâmetros, tipos de modelos, etc.) [PTC, 2006b]. Cada uma destas propriedades retorna um número inteiro de 0 até (x-1), onde o x é o número de propriedades da classe. A API Wizard enumera estas propriedades por ordem crescente, onde a primeira propriedade retorna sempre o número "0", a segunda propriedade retorna o número "1", etc.

A classe **pfcParamValueType** é um exemplo de uma classe de enumeração e apresenta sete valores/propriedades que dizem respeito aos tipos que os valores dos parâmetros podem assumir. A primeira propriedade da lista é a PARAM_STRING, que representa os valores dos parâmetros do tipo "string" e retorna o número inteiro "0". A propriedade

`ParamValueType_nil` é a última propriedade indicada e representa o valor “null”, retornando o número inteiro “5”. Esta propriedade serve para indicar que o valor do parâmetro não tem tipo associado e pode ser utilizada em argumentos opcionais que aceitem objetos da classe **`pfcParamValueType`**.

Estas classes podem ser retornadas por propriedades ou métodos ou podem ser instanciadas através da função `pfcCreate(className)`, classe referida anteriormente na secção das classes de dados compactos. Caso a classe seja instanciada é necessário atribuir o valor pretendido à classe invocando a propriedade adequada.

No código abaixo é instanciado um objeto da classe **`pfcParamValueType`** que represente todos os valores de parâmetros do tipo *string* e outro da classe **`pfcModelType`** que represente todos os modelos do tipo *assembly*.

```
//Instanciar um objeto da classe pfcParamValueType e associá-lo ao tipo
string
var assembly = pfcCreate("pfcParamValueType").PARAM_STRING;
//Instanciar um objeto da classe pfcModelType e associá-lo ao tipo assembly
var assembly = pfcCreate("pfcModelType").MDL_ASSEMBLY;
```

Código 3 - Extrato de código

4.3.5 Union

Classes com potencial para assumirem um de vários tipos de dados diferentes [PTC, 2006b].

A classe **`pfcParamValue`** pertence a este tipo de classes e a sua função é fornecer o valor de um determinado parâmetro de um modelo. Se tiver um objeto parâmetro (**`pfcParameter`**) e utilizar o método *Value* para obter o seu valor obtém-se um objeto da classe **`pfcParamValue`**, contudo isto não é suficiente para obter o valor do parâmetro. Como o valor deste parâmetro pode ser do tipo booleano, *string*, *double*, *integer* ou *note* é preciso também indicar, através do uso da propriedade adequada (`BoolValue`, `DoubleValue`, `IntValue`, `Noteld`, `StringValue`), qual o tipo do valor que se pretende ler.

Estas classes possuem sempre a propriedade *discr* para saber qual o tipo de dados que estas detêm [PTC, 2006b]. Ao utilizar esta propriedade sobre o objeto **`pfcParamValue`** obtém-se a

4.3 Classes

classe enumerada **pfcParamValueType** e de acordo com o seu resultado utiliza-se sobre o objeto **pfcParamValue** a propriedade que combina com o seu tipo. Por exemplo, se o tipo retornado pela propriedade *discr* for “booleano” então a propriedade a utilizar sobre o objeto **pfcParamValue** deve ser a “BoolValue”.

```
//Obter um parâmetro
var param = model.GetParam("nome_parametro");
//Ler o valor do parâmetro
if(param.Value.dscr == pfcCreate("pfcParamValue").PARAM_BOOLEAN)
//equivalente a "if(param.Value.dscr ==2)"
    alert(param.Value.BoolValue);
else if(param.Value.dscr == pfcCreate("pfcParamValue").PARAM_STRING)
    alert(param.Value.StringValue);
    (...)
```

Código 4 - Extrato de código

4.3.6 Sequence

As classes de sequência representam arrays de dimensão expansível. Estas classes possuem sempre a propriedade *Count*, que retorna o número de itens que fazem parte da sequência, e possuem sete métodos para ler ou editar o array (*Append*, *Clear*, *Insert*, *InsertSeq*, *Item*, *Remove*, *Set*) [PTC, 2006b].

Estas classes podem ser retornadas por propriedades ou métodos ou podem ser instanciadas através da classe estática *pfcCreate(className)*. Caso a classe seja instanciada é necessário adicionar itens ao *array* porque este está vazio. A instanciação destas classes é útil na criação de uma lista de objetos (por exemplo uma lista de modelos) para usar em algum método ou para comparar com outros objetos.

A classe **pfcModels** pertence a este tipo de classe e é usada para especificar uma lista de modelos. Os seus métodos são os seguintes:

- Método *Append(pfcModel Item)*: Adiciona um novo item ao final da sequência;
- Método *Clear()*: Apaga todos os itens da sequência;
- Método *Insert(Integer AtIndex, pfcModel Item)*: Insere um novo item da classe **pfcModel** numa determinada posição/índice da sequência. Se um item estiver na posição indicada neste método este é substituído;

- Método *InsertSeq(Integer AtIndex, pfcModels ModelsSequence)*: Insere uma nova sequência de modelos da classe **pfcModel** a partir de uma determinada posição/índice da sequência. Se estiverem itens na posição “0” e “1” e inserir uma sequência de 2 modelos a partir da posição 1, o item da posição 1 é substituído pelo primeiro modelo da nova sequência e o segundo modelo é armazenado na posição “2”, aumentando assim o tamanho original do array;
- Método *Item(Integer Index)*: Accede ao item que está localizado na posição indicada como argumento;
- Método *Remove(Integer FromIndex, Integer ToIndex)*: Apaga os itens que estão localizados entre as posições indicadas nos argumentos, inclusive o item da posição indicada no primeiro argumento (FromIndex) e excluindo o item da posição indicada no segundo argumento (ToIndex). Exemplos: o método Remove (0,0) não tem efeito; o método Remove(0,1) elimina somente o item que está na posição “0” e desloca para a esquerda todos os restantes itens das posições seguintes e por isso o item que estiver na posição “1” passa a estar na posição “0” e o mesmo acontece com os itens das posições seguintes, diminuindo assim o tamanho do vetor;
- Método *Set(Integer Index, pfcModel Item)*: Atribui um item da classe **pfcModel** a uma determinada posição da sequência.

4.3.7 Array

Classes que representam *arrays* de dimensão limitada. Uma das características destas classes é que os *arrays* não sabem quantas dimensões têm, sendo retornado um erro quando se utiliza o método “length” do JavaScript sobre estas classes [PTC, 2006b]. Estas classes possuem sempre 2 métodos para ler e editar o *array* (*Item*, *Set*).

A classe **pfcPoint3D** pertence a este tipo de classes, sendo um *array* unidimensional limitado a 3 elementos. Esta classe é usada para armazenar um ponto tridimensional, ou seja, um ponto de 3 coordenadas. A posição “0” corresponde à coordenada “x”, a posição “1” corresponde à coordenada “y” e a posição “2” corresponde à coordenada “z”.

Os métodos da classe **pfcPoint3D** são os seguintes:

4.4 Herança

- Método *Item(Integer Index)*: Acede à coordenada que está localizada na posição indicada como argumento;
- Método *Set(Integer Index, Number Item)*: Atribui um valor à coordenada que está localizada na posição indicada como argumento.

No caso da classe **pfcPoint3D** se indicar índices acima do índice “2” é retornado um erro uma vez que este array não pode ter mais do que 3 elementos.

Para instanciar este tipo de classes utiliza-se a função estática *pfcCreate (className)*, sendo atribuído por omissão o valor “0” a todas as posições do *array*.

```
//Instanciar um objeto da classe pfcPoint3D
var pfcPoint3D = pfcCreate("pfcPoint3D");
//Ler o valor da primeira posição/coordenada
alert(pfcPoint3D.Item(0);
//Ler o //Ler o valor da segunda posição/coordenada
alert(pfcPoint3D.Item(1);
//Ler o valor da terceira posição/coordenada
alert(pfcPoint3D.Item(2);
//Ler o valor da quarta posição/coordenada
alert(pfcPoint3D.Item(3); //erro
//Substituir o valor "0" da primeira coordenada pelo valor "1"
pfcPoint3D.Set(0,1.0);
//Ler novamente o valor da primeira posição/coordenada
pfcPoint3D.Item(0); //Apresenta o valor "1"
```

Código 5 - Extrato de código

4.4 Herança

As únicas classes que herdam métodos e propriedades de outras classes são as classes relacionadas com o Pro/ENGINEER e as classes de dados compactos [PTC, 2006b]. As classes que estendem os métodos e propriedades de outras classes designam-se por *subclasses* e as classes que são estendidas designam-se por *superclasses*. As subclasses podem utilizar as propriedades e métodos das superclasses acima destas [Lewis C., 2008].

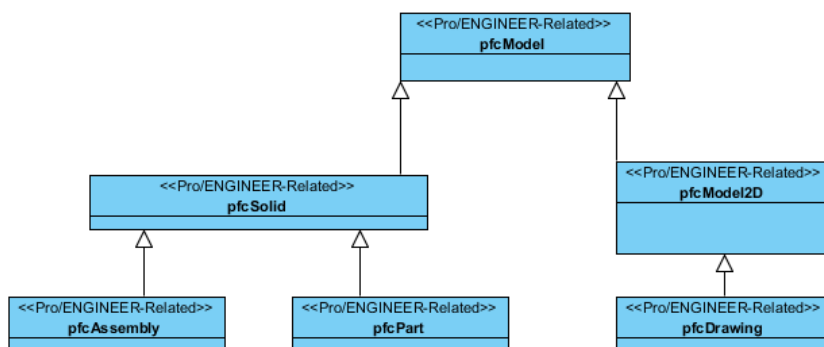


Figura 31 - Diagrama de classes Pro/Web.Link 1

A classe **pfcAssembly** é uma subclasse da classe **pfcSolid** e esta por sua vez é uma superclasse das classes **pfcAssembly** e **pfcPart**. Estas duas classes podem utilizar os métodos e propriedades da classe **pfcSolid**, da classe **pfcModel** e de todas as superclasses acima (superclasses da classe **pfcModel**, superclasses destas superclasses, etc.).

4.5 Pro/Web.Link API Wizard

Se na instalação do Pro/ENGINEER for também instalado o suplemento “Pro/Web.Link”, a documentação das classes e dos seus métodos e propriedades da API pode ser consultada a partir de um *browser* que possua o JFC (*Java Foundation Classes*) ativo. A PTC designa esta documentação de “APIWizard” [PTC, 2006b] e esta encontra-se no seguinte diretório: <diretório do Pro/ENGINEER>\Weblink\Weblinkdoc\index.html.

Nesta “APIWizard” o utilizador tem acesso, a partir do menu do lado esquerdo, aos vários módulos, classes, classes de exceção, tipos enumerados e ao manual da API (“Pro/Web.Link User's Guide”) que tem alguns exemplos práticos. Esses exemplos práticos podem ser testados uma vez que estão colocados em ficheiros JavaScript e HTML no diretório “<diretório do Pro/ENGINEER>\Weblink\weblinkexamples”. O manual também pode ser consultado a partir do diretório “<diretório do Pro/ENGINEER>\Weblink\weblinkug.pdf”. Cada versão do Pro/ENGINEER contém uma versão atualizada da API Pro/Web.Link. Com estas atualizações algumas das suas propriedades ou métodos podem vir a ser alterados ou a ficar obsoletos. Por esta razão, quando uma empresa pretende atualizar a versão do Pro/E deve averiguar, através da leitura de um documento fornecido pela PTC relativamente à nova versão da API (ex:

4.5 Pro/Web.Link API Wizard

“Weblink_Wildfire30_RelNotes.pdf”), se é necessário efetuar ajustes às aplicações desenvolvidas.

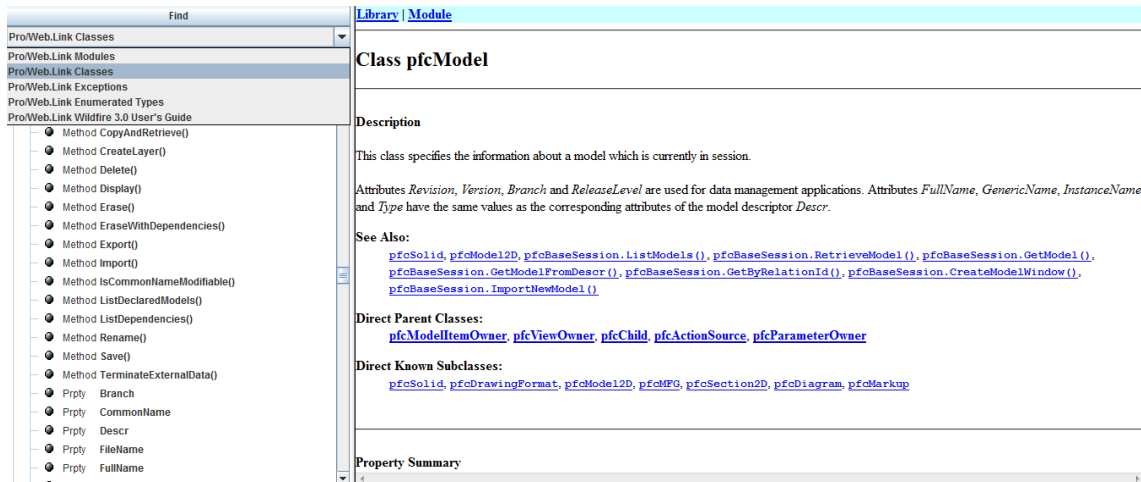


Figura 32 - API Wizard da API Pro/Web.Link

Os módulos agrupam as classes por temas e desta forma facilitam a procura da classe pretendida. Por exemplo, o módulo **pfcModel** contém todas as classes relacionadas com os modelos ou operações a modelos.

Quando uma determinada classe é selecionada podem ser observadas as seguintes informações: descrição da classe (**título “Description”**); identificação de métodos que estão relacionados de algum modo com a classe em questão (**título “See Also”**); identificação das classes pai diretas (**título “Direct Parent Classes”**) e classes filho diretas (**título “Direct Known Subclasses”**); identificação e descrição dos métodos (**título “Method Summary”**) e propriedades (**título “Property Summary”**) da classe, explicitando os argumentos necessários e o tipos de objetos que são retornados (pfcModel, String, etc.) (referência da API Wizard). Existem determinadas propriedades que não podem ser modificadas pela API Pro/Web.Link, tendo esta somente permissões de leitura [PTC, 2006b].

Em algumas propriedades e métodos a PTC não expõe de forma explícita qual a classe do objeto retornado ou de um determinado argumento e por isso o programador deve interpretar a informação fornecida. Por exemplo, quando um objeto retornado pode ser da

classe **pfcPart** ou **pfcAssembly**, a PTC indica que o objeto retornado é da classe **pfcSolid** pois esta classe representa os modelos do tipo *part* ou *assembly*. Esta é uma forma de generalizar o resultado pois o objeto pode ser de uma classe ou de outra. O mesmo acontece com alguns argumentos pois quando um argumento suporta objetos da classe **pfcSolid** significa que aceita objetos da classe **pfcPart** ou **pfcAssembly**.

A API Wizard também possui uma funcionalidade de pesquisa através da seleção do botão “Find”, situado acima do menu do lado esquerdo, que permite pesquisar pelos nomes das classes (opção “API Names”), pelos conteúdos das classes (opção “API definitions”) e/ou pela informação do manual.

4.6 Sumário

Neste capítulo foi alcançado parte do objetivo *Estudo da API Seleccionada*, sendo fornecido um contributo. Os aspetos relacionados com a integração entre a Pro/Web.Link e o Pro/E, que visam completar o estudo deste objetivo, são discutidos no próximo capítulo **5 Integração Intranet – Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link.**

Objetivo

- **Estudo da API Seleccionada:** o departamento de Sistemas de Informação escolheu a API Pro/Web.Link para alvo de estudo e para o desenvolvimento das aplicações do programa Colombo. Este estudo permitiu compreender sobretudo o que é a API Pro/Web.Link (um controlador ActiveX *in-process*), quais são os tipos de classes que a constituem (classes relacionadas com o Pro/ENGINEER, classes *Module - Level*, classes de dados compactos, classes de enumeração, classes união, classes de sequência, classes *Array*) e como interpretar a documentação da API (API *Wizard*).

Contributo

- Documentação sobre a API Pro/Web.Link com a seguinte informação: arquitetura da API; os tipos de classes que a constituem; algumas noções sobre o conceito de herança e sobre a documentação da API *Wizard*.

5 Integração Intranet – Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link

Este capítulo encontra-se dividido em sete secções. A primeira secção **5.1 Introdução** faz uma breve introdução à API Pro/Web.Link e possibilidade de integrar aplicações *Web* com o Pro/ENGINEER utilizando esta API. A segunda secção **5.2 Arquitetura** apresenta uma proposta da arquitetura que possibilita a interação entre uma aplicação *Web* e o Pro/ENGINEER Wildfire 3.0. A terceira secção **5.3 Configuração do Ambiente de Desenvolvimento/Execução** resume os passos que compõem o processo de configuração do ambiente de desenvolvimento/execução de uma aplicação *Web* desenvolvida com a API Pro/Web.Link, explicando alguns conceitos relacionados com a interação entre uma aplicação *Web* e o Pro/ENGINEER Wildfire 3.0. A quarta secção **5.4 Módulo JavaScript** apresenta de uma forma resumida as classes e as acções executadas pelo módulo JavaScript. A quinta secção **5.5 Método de Trabalho Proposto** apresenta a proposta de um método de trabalho para desenvolver uma aplicação com a API Pro/Web.Link. A quinta secção **5.6 Aplicações Desenvolvidas** identifica as tecnologias utilizadas no desenvolvimento das aplicações ProPEditor, ProCatalog e MassProp e a sua arquitetura. A sexta secção **5.7 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

5.1 Introdução

O Pro/ENGINEER Wildfire 3.0 possui um *browser* interno que permite executar aplicações *Web*. Estas aplicações podem comunicar com o Pro/E através dos objetos e funcionalidades da API Pro/Web.Link, permitindo assim a integração de uma intranet de uma empresa com o

software de CAD, o Pro/ENGINEER. Esta integração é útil para criar aplicações que satisfaçam necessidades específicas de uma empresa.

Segundo Christopher Lewis, para aumentar a eficiência das empresas deve-se apostar na reengenharia das tarefas diárias dos seus colaboradores pois a maior parte do tempo das organizações é gasto nestas funções. Com esta observação ele introduz o conceito *Simple Automation*, que consiste em programar pequenas aplicações com a API Pro/Web.Link em poucos dias para resolver pequenos problemas do dia-a-dia dos colaboradores [Lewis C., 2008].

Neste capítulo pretende-se introduzir os temas de configuração do ambiente de desenvolvimento, de execução e *deployment* das aplicações programadas com a API Pro/Web.Link para que o leitor possa compreender um pouco como funciona a interação entre uma aplicação *Web* e o Pro/E. Pretende-se também propor um método de trabalho adequado para implementar uma aplicação.

5.2 Arquitetura

Teoricamente a API não comunica diretamente com o Pro/E mas permite que o *browser* interno do Pro/E comunique com este através da invocação dos métodos ou propriedades desta API.

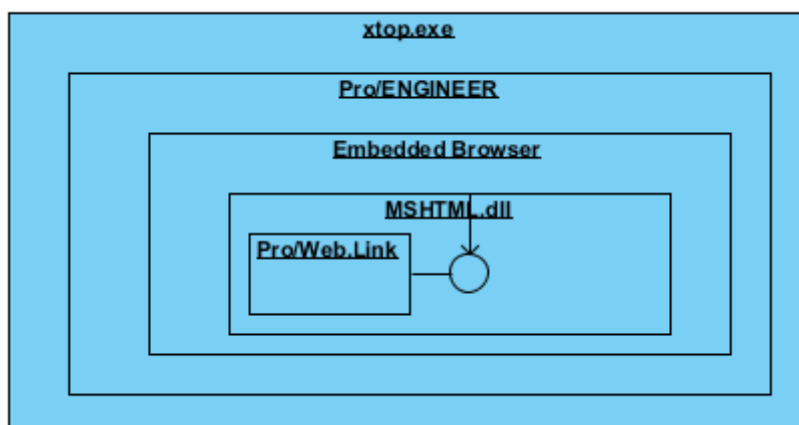


Figura 33 - Browser interno - Pro/Web.Link - Pro/ENGINEER Wildfire 3.0

5.3 Configuração do Ambiente de Desenvolvimento/Execução

Verificou-se com o programa Process Explorer que o processo do Pro/ENGINEER Wildfire 3.0 é o xtop.exe e que este contém threads que executam o Pro/ENGINEER e o *browser* interno. Teoricamente e em condições ideais (após a configuração do ambiente de desenvolvimento/execução), a API Pro/Web.Link é carregada para o componente MSHTML do *browser* e este componente invoca os objetos da API. Uma vez obtidos os objetos este comunica com o Pro/E através da invocação dos seus métodos ou propriedades. Se a opção *web_enable_javascript* do Pro/E não estiver ativa, o browser consegue invocar os objetos, contudo não consegue comunicar com o Pro/E.

Não existe qualquer documentação que chegue a este nível de detalhe, contudo com base na arquitetura da API Pro/Web.Link, nos conteúdos escritos neste e no capítulo anterior com referência à documentação da PTC, e no que foi observado no programa Process Explorer, desenhou-se a *Figura 33*. Christopher Lewis, Consultor Sénior da PTC Global Services concordou com esta figura. Esta interação é explicada, dentro dos possíveis, com um pouco mais de detalhe na próxima secção.

5.3 Configuração do Ambiente de Desenvolvimento/Execução

Para programar e testar as aplicações construídas em Pro/Web.Link é necessário configurar primeiro o ambiente de desenvolvimento e de execução. Para mais detalhes acerca desta configuração do ambiente de desenvolvimento, execução e da distribuição de uma aplicação pelos utilizadores (*deployment*) consulte a secção *Apêndice C*.

Em resumo, a configuração do ambiente de desenvolvimento e de execução é constituída por três etapas.

5.3.1. Configuração das Definições de Segurança no Internet Explorer

Esta etapa consiste em ativar, nas definições de segurança do IE, a opção “Processamento de scripts ativo” para permitir a execução das aplicações e ativar a opção “Inicializar e efetuar o script de controlos ActiveX não assinalados como seguros para o processamento de scripts” para permitir a execução das funcionalidades da API Pro/Web.Link uma vez que esta é um controlador ActiveX não seguro [PTC, 2006b].

No Windows, o Pro/ENGINEER possui um *browser* interno que foi construído com componentes do Internet Explorer [PTC, 2002]. Um dos componentes é o “MSHTML.dll” que tem como função fazer o parsing e rendering do código HTML e alojar os controladores ActiveX [MSDN, 2012e]. Quando o *browser* do Pro/E tenta executar scripts e controladores ActiveX, este interage com o *browser* Internet explorer instalado na máquina do utilizador para verificar se as definições de segurança do mesmo permitem a sua execução [Lewis C., 2008].

5.3.2. Registo da API Pro/Web.Link no Windows

Esta etapa consiste no registo do controlador ActiveX “pfcscm.dll” nos registos do Windows. Após este registo, a API fica visível para o Pro/ENGINEER [PTC, 2006b] e cada vez que uma aplicação for executada no *browser* interno do Pro/E ou no IE, a API é carregada para o componente MSHTML.dll. Esta última observação foi feita com o programa Process Explorer⁶ da Microsoft, que permite verificar os processos que são carregados e as respetivas threads.

O programa Process Explorer também permitiu observar que o componente MSHTML utiliza a função `DLLGetClassObject()`, que tem como objetivo comunicar com a interface da API [MSDN, 2012f]. Desta forma, o *browser* IE externo e o *browser* do Pro/E podem comunicar com a API para instanciar uma das classes, à exceção das classes relacionadas com o Pro/E, com o objeto JavaScript *ActiveXObject*. A partir de um *browser* externo não é possível interagir com o Pro/E, por exemplo através da utilização da função `GetProESession()` da classe **MpfcCOMGlobal**, uma vez que não há qualquer ligação com o Pro/E e como tal a API Pro/Web.Link lança a exceção “PfcXNotConnectedToProe” [PTC, 2006b]. A partir do *browser* do Pro/E é possível interagir com o Pro/E pois estes estão sempre ligados entre si, mas para a comunicação ser permitida é necessário ativar a opção `web_enable_javascript` no ficheiro de configuração do Pro/E [PTC, 2006b].

O *browser* do Pro/E acede ao Pro/ENGINEER utilizando o protocolo interno “proep” que possui acessos restritos [University of Cambridge, 2012].

⁶ Process Explorer: <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>.

5.3.3. Ativação da Opção *web_enable_javascript* no ficheiro *config.pro*

Ao iniciar o Pro/E, este verifica no seu ficheiro de configuração se a opção “*web_enable_javascript*” está ativa e em caso afirmativo o Pro/E carrega de imediato o controlador ActiveX para o componente MSHTML [PTC, 2006b]. Estando esta opção ativa o *browser* interno do Pro/E tem acesso completo à API Pro/Web.Link e consegue comunicar com o Pro/E.

5.4 Módulo JavaScript

Todas as aplicações em Pro/Web.Link devem utilizar um conjunto de funções comuns. Havendo um módulo JavaScript com estas funções, o programador pode preocupar-se apenas com as funções específicas da sua aplicação. Desta forma, todas as aplicações podem reutilizar as funções do módulo JavaScript. Como estas funções não são específicas das aplicações ProPEditor, ProCatalog e MassProp, estas não foram consideradas durante o processo de modelação. Para saber mais detalhes acerca do módulo JavaScript consulte a secção Apêndice D.

As funções do módulo JavaScript não foram ideia do autor uma vez que a PTC as disponibiliza também num módulo próprio (*pfcutils.js*). Contudo, foram efetuadas algumas melhorias relacionadas com a organização do código e com a apresentação das mensagens de erros para que fosse possível indicar ao utilizador exatamente qual ou quais os motivos do erro, no caso de este existir, relacionados com o processo de configuração do ambiente de desenvolvimento ou *deployment*. A identificação dos erros foi feita com base na análise do comportamento da API Pro/Web.Link.

O módulo JavaScript contém as seguintes funções que são comuns a todas as aplicações: **pfcCreate(className)**, para permitir a criação de um objeto da API Pro/Web.Link à exceção dos objetos das classes relacionadas com o Pro/ENGINEER que não podem ser instanciadas [PTC, 2006b]; e a função **pfcCreateSession()** para obter a sessão do Pro/ENGINEER.

Se o módulo for executado com sucesso então as configurações de desenvolvimento e execução estão corretas e a nova aplicação está pronta para executar.

A função **pfcCreate(className)** contém as funções **isProEEmbeddedBrowser()**, para verificar se o browser está imbutido no Pro/E, e a função **pfcIsWindows()** para verificar se o browser é da Microsoft e, em caso afirmativo, criar o objeto ActiveX. A função **pfcCreateSession()** utiliza a função **pfcCreate(className)** para obter um objeto da classe **MpfcCOMGlobal** e posteriormente obter a sessão do Pro/ENGINEER através do método **GetProESession()** dessa classe.

A seguinte figura apresenta de uma forma resumida as acções executadas pelo módulo JavaScript:

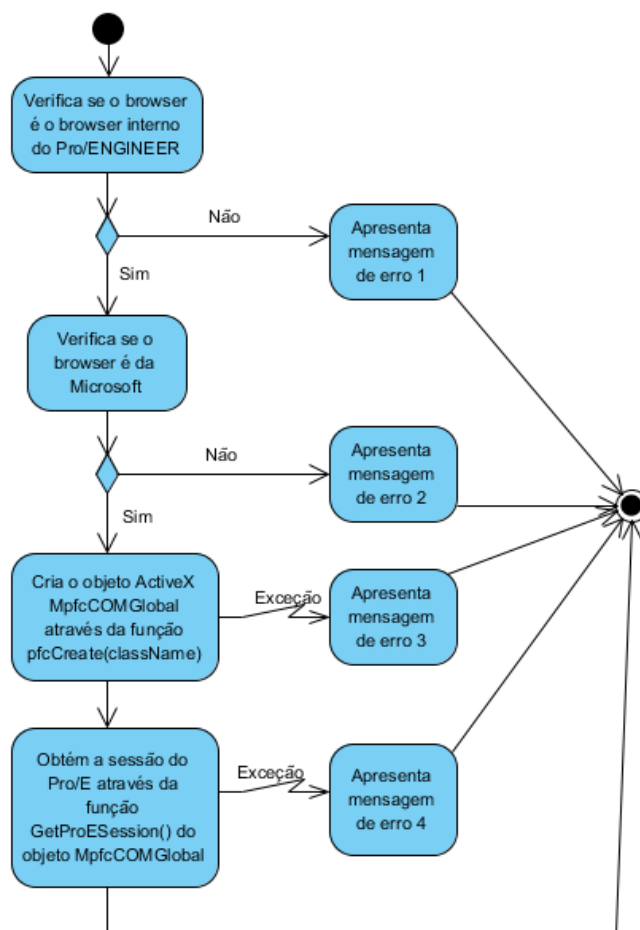


Figura 34 - Ações executadas pelo módulo JavaScript

Apesar da figura anterior referir que o módulo apresenta as mensagens de erro, na verdade este somente as armazena, ficando a apresentação das mensagens de erro ao cargo do programador através da criação de uma função. O módulo JavaScript não permite a execução

5.5 Método de Trabalho Proposto

de aplicações na versão UNIX do Pro/ENGINEER uma vez que não faz parte do contexto desta dissertação, havendo unicamente preocupação com a plataforma Windows.

Para saber mais detalhes sobre as funções e mensagens de erro do módulo JavaScript consulte a secção Apêndice D.

5.5 Método de Trabalho Proposto

A seguinte proposta do método de trabalho pretende facilitar o processo de desenvolvimento de uma aplicação com a API Pro/Web.Link. Supõe-se que o programador possui conhecimentos em JavaScript, tem uma descrição funcional da aplicação e possui conhecimentos suficientes sobre os conceitos relacionados com o Pro/ENGINEER.

Para implementar uma aplicação com a API Pro/Web.Link propõe-se o seguinte método de trabalho:

- 1. Instalação da documentação da API Pro/Web.Link a partir do setup de instalação do Pro/ENGINEER;**
- 2. Configuração do ambiente de desenvolvimento/execução da aplicação: este processo está descrito na secção 5.3;**
- 3. Testar se o ambiente de desenvolvimento foi bem configurado no passo anterior.**

Nesta etapa, para testar se o ambiente de desenvolvimento foi bem configurado deve-se utilizar o módulo JavaScript referido na secção anterior (5.4). Uma vez que é necessário programar então pode-se iniciar nesta fase o processo de programação da aplicação utilizando, por exemplo, o IDE Netbeans. Aconselha-se a criar três ficheiros: um ficheiro com o html ou php para desenhar a interface da aplicação (ex: aplicacao.html); outro ficheiro com o módulo JavaScript fornecido na secção 5.4 (ex: moduloJavascript.js); e outro ficheiro com o código JavaScript da aplicação a implementar com a API Pro/Web.Link (ex: minhaAplicacao.js). No ficheiro html deve ser feita uma referência aos ficheiros JavaScript.

```
<html>
  <head>
    <script type="text/javascript" src="moduloJavaScript.js"></script>
    <script type="text/javascript" src="minhaAplicacao.js"></script>
  </head>
  <body onload= "if(pfcCreateSession()){funcaoPrincipal()} else
  tratamentoErros(">
  </body>
</html>
```

Código 5 - Exemplo do código do ficheiro aplicacao.html

Como é indicado no código acima, para invocar a função principal do ficheiro *minhaAplicacao.js* deve ser invocada primeiro a função *pfcCreateSession()* do módulo *moduloJavaScript.js*. Se esta função for executada com sucesso significa que o ambiente de desenvolvimento/execução da aplicação foi bem configurado e a função principal da aplicação pode ser executada de seguida (ex: *aplicacaoPrincipal()*). Por restrições da PTC, os métodos e propriedades da API Pro/Web.Link só podem ser utilizados a partir do *browser* interno do Pro/E por isso a função deve ser executada neste *browser*, caso contrário dá um erro. Se não for executada com sucesso invoca a função *tratamentoErros()* que pode ser implementada no módulo da aplicação (*minhaAplicacao.js*) e que retorna a mensagem do erro que ocorreu efetuando um *alert* à variável *msg*, que é a variável que identifica o erro. No passo 5 é apresentado o exemplo do código da aplicação com a função *tratamentoErros()*.

Embora no caso do código acima, a aplicação principal esteja a ser invocada assim que a aplicação é carregada, isso não é imperativo pois pode ser necessário utilizar os métodos da API Pro/Web.Link apenas quando o utilizador selecionar, por exemplo, algum botão.

4. Ler a informação do capítulo 4 desta dissertação sobre a API Pro/Web.Link

Esta etapa consiste em ler alguns conceitos sobre a API Pro/Web.Link, nomeadamente que tipos de classes existem e como estas podem ser instanciadas e/ou utilizadas para se começar a programar o código da aplicação (*minhaAplicacao.js*). É importante perceber também a capacidade de herança das classes para saber interpretar a documentação da API *Wizard*, identificando qual ou quais os tipos de objetos que podem ser retornados quando se utiliza um dos seus métodos. Por exemplo, quando a API *Wizard* refere que retorna um objeto da classe *pfcModel*, na verdade está a retornar um objeto que pode ser da classe *pfcAssembly*, *pfcPart* ou *pfcDrawing*. Compreender os tipos de classes também é fundamental para saber como instanciar uma determinada classe da API uma vez que a documentação API *Wizard* não

5.5 Método de Trabalho Proposto

identifica, por exemplo, qual o tipo de classe da classe *pfcModel*. Após este estudo é mais fácil identificar que a classe *pfcModel* é um tipo de classe *Pro/ENGINEER – related* e como tal não pode ser instanciada. Esta etapa pode ser realizada antes da etapa 2.

5. Desenvolver e testar a Aplicação

É relevante nesta fase perceber que qualquer comunicação com o Pro/E é feita através da sessão do Pro/E e como tal é obrigatório instanciar o objeto *MpfcCOMGlobal* e obter a sessão com o seu método *GetProESession()*. Para obter esta sessão pode utilizar-se a função já desenvolvida *pfcCreateSession()* do módulo JavaScript (*moduloJavascrip.js*). A partir daqui podem ser utilizados todos os métodos e propriedades do Pro/E que se desejarem. Para instanciar um objeto pode ser utilizada a função *pfcCreate()* do módulo Javascript (*moduloJavascript.js*) que já tem as validações necessárias para o Windows. Contudo, é necessário que o programador verifique com o seu código se ocorreu algum erro, pois esta função retorna *false* e armazena a mensagem de erro no caso de insucesso na variável global *msg*.

```
function funcaoPrincipal()
{
    var session = pfcCreateSesion();
    if(session)
        Continua a executar..
    else
        tratamentoErros()
}

function tratamentoErros()
{
    alert(msg); //apresenta a mensagem de erro;
}
```

Código 6 – Exemplo do código do ficheiro *minhaAplicacao.js*

O tratamento dos erros dos métodos e propriedades da API Pro/Web.Link fica ao critério do utilizador.

6. Deployment da aplicação

Esta etapa encontra-se descrita no apêndice C.

5.6 Aplicações Desenvolvidas

Após a escolha da API Pro/Web.Link, foram desenvolvidas as duas aplicações do programa Colombo com esta API do Pro/E, nomeadamente as aplicações ProPEditor e ProCatalog. Estas aplicações têm como finalidade facilitar o trabalho das equipas de Engenharia do Produto, Engenharia Industrial e de Inovação e Desenvolvimento (I&D) das companhias de Portugal, Argentina e Índia. Também foi desenvolvida a aplicação MassProp que não fazia parte do plano inicial. Dado que contribuía positivamente para o trabalho dos engenheiros e para a exploração de novas funcionalidades da API Pro/Web.Link, não exigindo demasiado tempo de desenvolvimento, o departamento de SI decidiu que esta devia ser implementada.

Antes da definição dos requisitos das aplicações, o departamento de SI decidiu que as aplicações solicitadas devem ser aplicações *Web* devido às suas vantagens aquando comparadas com as aplicações *standalone*, nomeadamente:

- O processo de distribuição e instalação é mais eficiente pois o utilizador apenas necessita de saber o URL para executar as aplicações a partir de um *browser*, o que facilita o acesso aos colaboradores do Grupo Efacec que estão espalhados pelo mundo;
- O processo de atualização da aplicação é transparente para o utilizador;
- A existência de um *Framework*, desenvolvido pelo departamento de SI, para auxiliar no desenho da interface das aplicações *Web*, nas ligações às bases de dados, etc.;
- A possibilidade de integrar a intranet utilizando alguns dos seus mecanismos implementados em PHP como por exemplo o sistema de login, tornando possível a deteção do nome do utilizador que está atualmente a utilizar a sessão, da companhia a que este pertence e do idioma em que a página é apresentada para posterior utilização do sistema de traduções da intranet (Translator).

Numa fase anterior à análise das aplicações ProPEditor e do ProCatalog, o departamento de SI da unidade AMT forneceu dois enunciados com uma descrição breve das especificações destas aplicações e dos seus requisitos ([Anexo 1](#) e [Anexo 2](#)).

Após a escolha da API Pro/Web.Link, foram adicionados alguns requisitos não funcionais ao enunciado do ProPEditor e ProCatalog, nomeadamente:

5.6 Aplicações Desenvolvidas

- A aplicação deve usar a API Pro/Web.Link do Pro/ENGINEER;
- Todas as funcionalidades da Pro/Web.Link devem ficar num módulo JavaScript para possibilitar a sua reutilização noutras aplicações.

A análise efetuada a cada uma destas aplicações reflete essas especificações, identificando e descrevendo os casos de uso, isto é, os cenários de interação das aplicações com o utilizador. As análises podem ser consultadas no [Apêndice E](#) e [Apêndice F](#). Para compreender o processo de funcionamento das aplicações, foi fundamental assimilar a teoria envolvida em torno do Pro/ENGINEER e do Windchill. Após esse trabalho foi mais fácil encontrar as funções da API Pro/Web.Link, através da sua API *Wizard*, necessárias para implementar as funcionalidades requeridas

5.6.1 Tecnologias Utilizadas

As tecnologias utilizadas no processo de desenvolvimento das aplicações ProPEditor, ProCatalog e MassProP foram as seguintes:

Sistema operativo

- Microsoft Windows XP Professional x64 Edition

Web server

- Servidor Apache com suporte PHP

IDE

- Netbeans IDE

Servidor de bases de dados

- Microsoft SQL Server 2005

Bibliotecas

- API Pro/Web.Link do Pro/ENGINEER Wildfire 3.0 para implementar as funcionalidades relacionadas com o Pro/E, e o *Framework* da SynergyNet para o desenho da interface gráfica das aplicações ProPEditor e ProCatalog.

Linguagens de programação

- **PHP:** Esta linguagem foi usada para a integração das páginas ProPEditor e ProCatalog na intranet da unidade AMT através da utilização da classe de domínio CMain (ver secção 2.3 Framework da SynergyNet), e para o desenho da interface gráfica utilizando as classes CForm e CTable para a construção de formulários e tabelas. Esta também foi utilizada para interagir com a base de dados do SQL Server 2005.
- **HTML:** Este pode ser gerado pelo PHP. Para o desenho da interface da aplicação MassProp é utilizado o HTML, não se recorrendo a classes da SynergyNet visto que não é necessário utilizar sistema de *login* e que a tabela pode ser mais simples que a construída pela classe CTable.
- **JavaScript:** Para utilizar os atributos e métodos da API Pro/Web.Link.
- **AJAX (Asynchronous JavaScript and XML):** Para realizar pedidos assíncronos ao servidor PHP após o código JavaScript ter sido interpretado no lado do cliente e para permitir a atualização de um determinado conteúdo sem ter de recarregar toda a página. Somente as aplicações ProPEditor e ProCatalog utilizam esta tecnologia.

5.6.2 Arquitetura

A *Figura 35* apresenta os componentes envolvidos na execução das aplicações ProPEditor e ProCatalog da intranet e a integração dessas aplicações com o Pro/E através da API Pro/Web.Link. A aplicação MassProp é a única que não interage com o servidor de bases de dados e que foi desenvolvida apenas em HTML e JavaScript não recorrendo ao AJAX.

Quando um utilizador pretende visualizar o ProPEditor ou ProCatalog num *browser*, este faz um pedido ao servidor *Web* (Apache) para retornar a aplicação que este está a guardar. Nesta intranet, o servidor *Web* suporta PHP e interpreta o código PHP da aplicação. Quando é efetuado algum pedido à base de dados, o servidor *Web* envia o pedido ao servidor de aplicação e este por sua vez redireciona-o ao servidor de base de dados que o traduz e retorna o resultado. O servidor aplicacional transmite o resultado ao servidor *Web* em formato HTML. A página contém código PHP e JavaScript, pelo que o servidor *Web* envia para o cliente a página formatada em HTML, gerado a partir do PHP, e JavaScript.

5.6 Aplicações Desenvolvidas

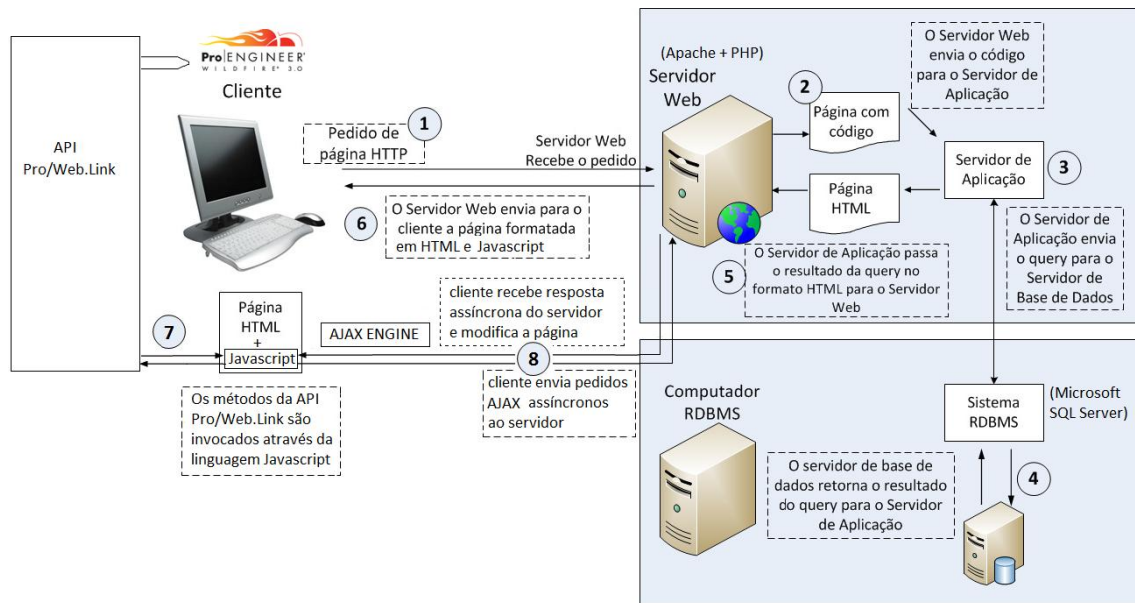


Figura 35 - Arquitetura ProPEditor, ProCatalog, MassProp (adaptação [Alexandre L., 2011])

No caso da aplicação MassProp, o servidor *Web* não necessita de interpretar o código, uma vez que esta é escrita em HTML e JavaScript, nem de enviar qualquer pedido ao servidor de bases de dados, retornando de imediato a página assim que a aplicação MassProp é invocada.

Após o retorno da aplicação em formato HTML e JavaScript, o *browser* IE ou o *browser* interno do Pro/E verifica se a opção de segurança “*processamento de scripts ativo*” está ativa e se sim invoca e interpreta dois módulos JavaScript, um que contém as funções comuns a todas as aplicações (*Apêndice D*) e outro que contém as funções específicas da aplicação invocada. Caso a opção não esteja ativa, o *browser* apresenta apenas os conteúdos HTML e não executa o JavaScript.

Quando se pretende instanciar objetos ActiveX da API Pro/Web.Link, o *browser* verifica se a opção “inicializar e efetuar o script de controlos ActiveX não assinalados como seguros para o processamento de scripts” está ativa e se o controlador ActiveX *pfcom.dll* está em memória. Em caso afirmativo os objetos são instanciados e as funcionalidades da API são invocadas pelo *browser*. Devido a restrições implementadas pela PTC, para invocar estas funcionalidades com

sucesso é necessário que a opção *web_enable_javascript* do ficheiro de configuração do Pro/E esteja ativa e que a aplicação seja executada dentro do *browser* interno do Pro/E [PTC, 2006b].

Para não complicar a interpretação da *Figura 35*, esta apresenta de uma forma muito genérica a interação entre as aplicações Web e o Pro/E através da API Pro/Web.Link. Para visualizar a arquitetura mais correta consulte a secção deste capítulo.

Nas aplicações ProPEditor e Procatalog são feitos pedidos AJAX para efetuar novos pedidos ao servidor PHP após a interação com o Pro/E. Por exemplo, na aplicação ProPEditor, após esta verificar se os modelos estão *checked out*, através de uma funcionalidade da Pro/Web.Link, pretende-se solicitar um novo pedido ao servidor PHP para sugerir os valores para os parâmetros e isso é possível através de um pedido assíncrono usando o AJAX.

5.7 Sumário

Neste capítulo completou-se o objetivo *Estudo da API Seleccionada*, sendo fornecidos dois contributos.

Objetivo

- **Estudo da API Seleccionada:** Após o estudo dos tipos de classes da API Pro/Web e de outros conceitos apresentados no capítulo 4, foi realizado um estudo à API Pro/Web.Link na perspetiva de integrar uma aplicação Web com o Pro/ENGINEER e de perceber como funciona essa integração. Este estudo permitiu não só identificar os passos necessários para configurar o processo de desenvolvimento/execução de uma aplicação e para distribuir uma aplicação pelos utilizadores, como também compreender um pouco mais sobre o processo de interação entre uma aplicação *Web* e o Pro/ENGINEER Wildfire 3.0. Para além disso foi possível melhorar o módulo JavaScript da PTC, que contém as funções comuns a qualquer aplicação *Web* desenvolvida com a API Pro/Web.Link, no que diz respeito à apresentação das mensagens de erro, permitindo que o programador identifique facilmente um erro no caso de este existir e que foque essencialmente nos métodos específicos da aplicação. Neste capítulo também foram identificados alguns componentes que envolvem o processo de interação entre uma página *Web* e o Pro/ENGINEER Wildfire 3.0.

Contributo

- Versão melhorada do módulo JavaScript da PTC, que contém as funções comuns a todas as aplicações que sejam desenvolvidas com a API Pro/Web.Link, no que diz respeito à organização do código e ao tratamento de erros;
- Documentação sobre a API Pro/Web.Link com a seguinte informação: o processo de configuração do ambiente de desenvolvimento e o *deployment* de uma aplicação e outros conceitos relacionados com este processo; o módulo JavaScript para ser utilizado em todas as aplicações programadas com a Pro/Web.Link.

6 ProPEditor

Este capítulo encontra-se dividido em cinco secções. A primeira secção **6.1 Contextualização** introduz a importância da necessidade da aplicação ProPEditor, explicando alguns conceitos relacionados com o processo de edição da legenda de um desenho efetuado na Efacec. A segunda secção **6.2 Modelação do Novo Sistema** apresenta o formato casual e completo do caso de uso *Editar Legenda* da aplicação ProPEditor. A terceira secção **6.3 Aplicação Final** apresenta a aplicação ProPEditor final, sendo identificadas algumas das suas funcionalidades. A quarta secção **6.4 Vantagens** descreve as vantagens atingidas com o ProPEditor. A quinta secção **6.5 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

O enunciado da aplicação ProPEditor é apresentado na secção **Anexo 1 - Enunciado do ProPEditor**.

As classes e respetivos métodos e propriedades utilizados na aplicação ProPEditor são apresentados na secção **Apêndice G - ProPEditor: Classes Pro/Web.Link**.

6.1 Contextualização

O ProPEditor, *software* para edição de parâmetros no Pro/ENGINEER surgiu de uma ideia que o colaborador Rui Marinho expôs no Colombo. Tem como objetivo principal agilizar a tarefa de preenchimento da legenda dos desenhos proporcionando vários automatismos.

Cada desenho do Pro/E tem como objetivo representar em 2D um modelo 3D (*part* e *assembly*) e o Pro/E permite que um desenho tenha zero ou mais modelos associados. Estes

modelos são apresentados na *model tree* do Pro/E. É uma regra interna da empresa que um desenho deve representar somente um modelo, isto é, deve ter um único modelo associado, e possuir o mesmo nome desse modelo. Se um desenho projetar um modelo nomeado *EMT294354.prt*, o desenho deve ter o nome *EMT294354.drw*. Em algumas situações é essencial fugir à regra e adicionar vistas de outros modelos, que podem estar relacionados com o modelo/desenho em causa, para melhor clarificar o desenho, pelo que podem estar associados outros modelos ao desenho.

Na Efacec, todos os modelos são criados a partir de *templates*, e esses *templates* têm os parâmetros, ou legenda no caso dos desenhos, definidos.

Cada modelo pode ter um conjunto de parâmetros que servem para manter alguma informação acerca deles: o parâmetro *MATERIAL*, que especifica qual o material que o constitui; o parâmetro *PROJETISTA*, que identifica quem o projetou; entre outros. O Pro/E aceita parâmetros dos tipos *integer* (números inteiros), *string* (sequência de caracteres), *real number* (constituído por parte inteira e parte fracionária – exemplo: 3.1714) e *boolean* (*true* ou *false*). Se houver integração do Pro/E com o Windchill, os parâmetros do modelo devem ter um visto na opção *designate* para quando o modelo for guardado no Windchill (através da operação *upload* ou *check-in*), o Windchill consiga reconhecê-los e atualizar os seus valores nos atributos respetivos [PTC, 2007b] para mais tarde apresentar ao utilizador a partir do *commonsplace* ou *workspace* do Windchill PDMLink (Figura 25).

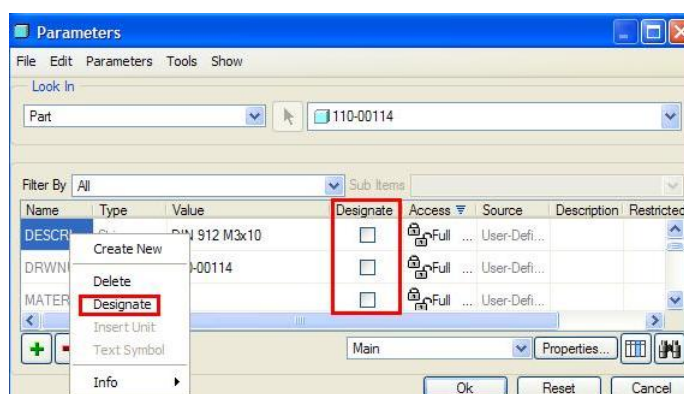
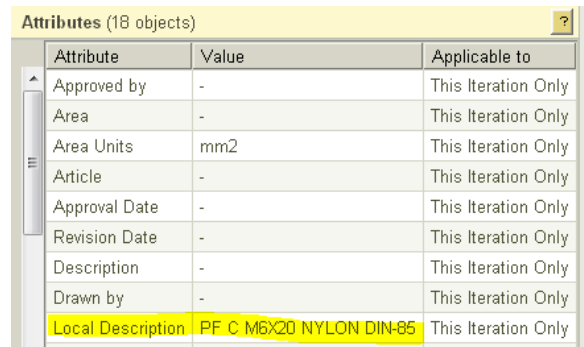


Figura 36 - Janela dos parâmetros de um modelo (repetição)

6.1 Contextualização

A opção *designate* é importante pois permite a visualização dos valores dos parâmetros devidamente atualizados no Windchill, sem ser necessário abrir o modelo.



Attribute	Value	Applicable to
Approved by	-	This Iteration Only
Area	-	This Iteration Only
Area Units	mm2	This Iteration Only
Article	-	This Iteration Only
Approval Date	-	This Iteration Only
Revision Date	-	This Iteration Only
Description	-	This Iteration Only
Drawn by	-	This Iteration Only
Local Description	PF. C. M6X20 NYLON DIN-85	This Iteration Only

Figura 37 - Atributos de um modelo apresentado no *commonsplace* do Windchill PDMLink

Um modelo do tipo *part* ou *assembly* pode ser um genérico ou uma instância. Enquanto uma instância tem obrigatoriamente um modelo genérico associado, um modelo genérico nem sempre tem instâncias associadas. Quando se abre uma instância no Pro/E é garantido que o seu genérico está em sessão, o mesmo não acontece quando se abre um modelo genérico pois não é garantido que as suas instâncias estejam em sessão. Pode-se dizer que um modelo genérico é o modelo original, o modelo pai, e a partir deste podem criar-se novos modelos similares (instâncias) na *family table*.

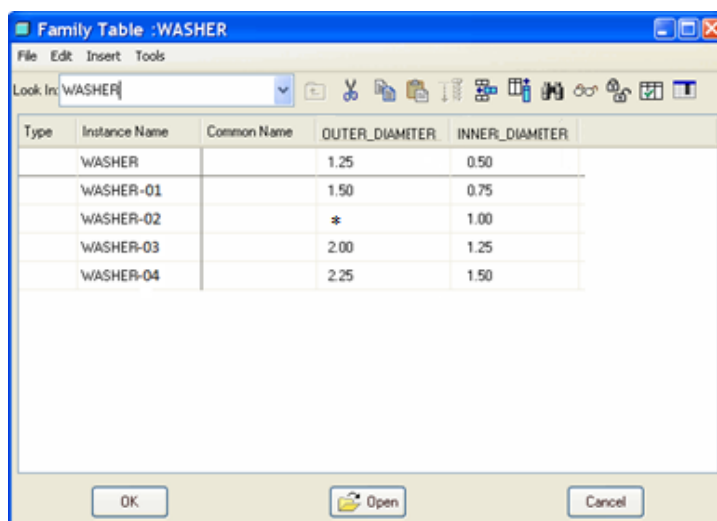
Desta forma, o Pro/E introduz o conceito de *family table*, uma tabela que pertence a um determinado modelo genérico e que apresenta uma listagem de todas as instâncias criadas a partir dele. Quando as instâncias são criadas estas herdam automaticamente todos os valores dos parâmetros do genérico. Se o utilizador pretender que as instâncias possuam valores diferentes para estes parâmetros, tem de adicioná-los à *family table*. Não é possível adicionar à *family table* outros parâmetros que não estejam no genérico.

Os parâmetros do genérico podem ser classificados de três formas [PTC, 2007b]:

- **Generic-Driven:** Não faz parte da *family table*, tendo o mesmo valor quer no genérico quer na instância. Como a modificação do valor deste parâmetro afeta o genérico e as

instâncias, é necessário fazer *check out* ao genérico e a todos os membros da *family table* (instâncias) para que a alteração seja efetuada com sucesso.

- **Independent Table-Driven:** Foi adicionado à *family table* e assume um valor diferente nas instâncias. Para editar o valor do parâmetro é necessário fazer *check out* ao modelo que pretende modificar (genérico ou instância).
- **Inherited Table-Driven:** Foi adicionado à *family table* e assume o mesmo valor do genérico através da colocação de um asterisco. Se este valor for editado na instância, retirando o asterisco e colocando um valor específico, é necessário fazer *check out* apenas a essa instância pois o modelo genérico não se modifica com a alteração deste valor. Contudo, se este atributo for modificado no genérico, é necessário fazer *check out* ao genérico e às instâncias que possuem * no valor deste parâmetro, pois uma modificação no genérico vai refletir-se também nessas instâncias.



Type	Instance Name	Common Name	OUTER_DIAMETER	INNER_DIAMETER
	WASHER		1.25	0.50
	WASHER-01		1.50	0.75
	WASHER-02		*	1.00
	WASHER-03		2.00	1.25
	WASHER-04		2.25	1.50

Figura 38 - Janela da Family Table de uma anilha (adaptação de [PTC, 2009c])

Após a criação das instâncias, estas devem ser validadas através da opção *Verify* do menu *Tools* da janela da *Family Table* (Figura 38) do Pro/E. Esta validação permite que o Pro/E verifique se estas instâncias são regeneradas com sucesso. Se todas as instâncias são verificadas com sucesso, o utilizador está apto para gravá-las no *workspace* do Windchill ou

6.1 Contextualização

noutro local caso não haja ligação com o Windchill. Se esta verificação correr mal, então é fundamental corrigir a instância em questão.

Em suma, através da criação de uma *family table*, o Pro/E permite poupar tempo e criar vários modelos semelhantes a partir de uma única modelação e através da variação de alguns itens. Estes itens são os parâmetros (dimensões, material, entre outros.) que, geralmente, diferenciam as instâncias do seu genérico.

Os desenhos podem ter mais do que uma página e são constituídos por uma legenda estandardizada definida pela Efacec. Esta legenda deve estar localizada no canto inferior direito da folha do desenho e pode estar repetida pelas várias páginas.



Tolerâncias gerais: ISO 2768-m, I.O. MT.80.2.01 ou APES 98025		General Tolerances: ISO 2768-m, I.O. MT.80.2.01 or APES 98025			
This document is the sole property of EFACEC. Its confidential, reproduction, conveyance of information to third parties and other unauthorized use are not permitted except in exact accordance with prior express permission.		Este documento é propriedade exclusiva da EFACEC. Sendo confidencial, não poderá ser comunicado a terceiros, nem utilizado ou reproduzido sem prévia autorização expressa.			
Scale Escala	Mat.: Cu-ETP HB ISO 6958 32x5mm	Prot.: -	Similar Item / Item Similar: 33109516-01		
1/1	-	-	-		
Projected by Projectado por	C.Pinheiro	Used by Usado em:	33110624-01		
Drawn by Desenhado por		Article Artigo:	33111031-01		
Date Data	2011-01-17	Quality Level Nível de Qual.:	-		
Approved by Aprovado por	M.Carvalho	 U.N. Aparelhagem / Switchgear B.U.			
Date Data	2011-05-20				
MAIN BUSBAR BARRAMENTO GERAL		-	3311031	A3	Page / Pág. 1/1
		Revision / Revisão			
Weight Peso (kg)	2.136	Index / Índice	Number / Número	Date / Data	
Area Área (mm ²)	89046		CN00000190	2011-05-06	
		FLUOFIX GC			

Figura 39 - Legenda de um desenho no Pro/ENGINEER Wildfire 3.0 2

Todo o texto da legenda da Efacec está representado por um único objeto a que o Pro/E designa por símbolo e pode ser movido como uma única entidade. Os símbolos podem conter elementos gráficos (linhas, arcos, etc.), elementos textuais variáveis e, uma vez que podem ter notas associadas, elementos textuais estáticos [Lewis C., 2008].

Projected by Projectado por	C.Pinheiro
--------------------------------	------------

Figura 40 - Texto variável e estático de uma legenda

Na *Figura 40*, o texto “*Projected by* Projectado por” é estático, e “C.Pinheiro” é texto variável. O Pro/E designa os textos estáticos de *notas* e estas têm apenas carácter informativo. Os textos variáveis são característicos dos símbolos e correspondem a campos que o utilizador pode editar e que podem estar relacionados com parâmetros do modelo (*part* ou *assembly* – genérico ou instância) que o desenho está a projetar.

Para associar um símbolo ao desenho este tem de estar definido. O símbolo utilizado nos desenhos da Efacec foi definido com o nome *TEXTO-LEGENDA*. Neste símbolo são identificados os nomes dos campos da legenda a editar (*Variable Text*), e definidas algumas das suas características tais como a altura, etc. O nome de cada campo tem de ser inserido com uma barra ao início e outra no final (exemplo: \nome\). Após a definição do símbolo este é adicionado manualmente à galeria de símbolos dos desenhos e posteriormente é necessário atribuí-lo ao desenho indicando qual o nome do símbolo e outras características tais como a cor da legenda, os valores dos campos, etc. No momento em que o símbolo é associado ao desenho, o Pro/E cria uma instância desse símbolo. Neste caso se o desenho tiver duas páginas e cada página apresentar uma legenda da Efacec significa que cada página contém uma instância diferente do símbolo *TEXTO-LEGENDA*.

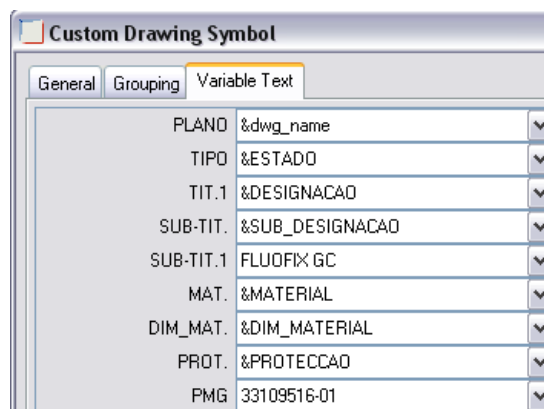


Figura 41 - Campos de texto variável

A relação entre os parâmetros e os campos *Variable Text* é indicada no Pro/E através da inserção do & seguido do nome do parâmetro no respetivo campo do texto variável (*Figura 41*). Quando se insere esse parâmetro, este fica associado ao modelo que está ativo nesse

6.1 Contextualização

momento. Os utilizadores não têm de se preocupar com esta questão porque os desenhos são criados a partir de templates e quando associam um modelo ao desenho este cria automaticamente o símbolo e as notas que constituem a legenda, definindo automaticamente os campos de texto variáveis e a sua relação com os parâmetros. De seguida, o Pro/E retorna os valores desses parâmetros para os campos da legenda do desenho.

O Pro/E possibilita que o preenchimento da legenda seja feito diretamente no desenho, onde o utilizador seleciona o campo que pretende alterar e modifica o seu valor. Contudo, no caso dos campos que estão associados a parâmetros, o utilizador pode também selecionar o modelo representado pelo desenho e alterar os parâmetros em questão através da janela de parâmetros do Pro/E (*Figura 36*).

Esta opção de edição dos parâmetros possibilita a apresentação de listas de valores pré-definidos para determinados parâmetros através da criação de ficheiros de texto com a especificação desses mesmos valores. Contudo, para cada uma destas listas não é possível apresentar valores provindos de outras aplicações ou fontes (ex: bases de dados) ou sugerir o valor mais correto, como por exemplo sugerir qual o utilizador que está a editar a legenda, e em algumas situações os preenchimentos de alguns campos tornam-se um pouco repetitivos, pelo que houve necessidade de um mecanismo de sugestão de valores. *Com esta necessidade, surge a ideia de desenvolver o ProPEditor, uma aplicação que tem como principais objetivos permitir o preenchimento automático de alguns campos da legenda e apresentar listas dinâmicas com base em informação armazenada em bases de dados.*

Dentro da empresa, alguns colaboradores são responsáveis pela aprovação dos objetos criados pelos projetistas/desenhadores, pelo que cada projetista está associado a um aprovador. Essa informação está definida no Profiler, uma aplicação programada pelo departamento de Sistemas de Informação da AMT que detém os perfis dos utilizadores da unidade AMT.

Outro dos campos da legenda diz respeito ao código da *change notice* (boletim ou número de modificação). Este campo está associado ao parâmetro *NUM_MOD*. O conceito de *change notice* surgiu com o Visualizer, que tal como o Profiler também é uma aplicação desenvolvida dentro da AMT. O Visualizer agrupa funcionalidades relacionadas com a documentação do produto, maioritariamente desenhos, permitindo a visualização de desenhos que sejam protótipos ou estejam acabados, como também a gestão das *change notices* por parte dos

utilizadores. As *change notices* agregam as modificações feitas aos objetos. Estas possuem um código de identificação e relacionam-se com um determinado produto, estando ligadas a uma categoria (Redução de custos, problemas de *design*, melhoria do produto, etc.) e a uma revisão do objeto. Estas *change notices* possuem diferentes prioridades (baixa, média, alta, de emergência) e podem estar no estado “em uso”, “arquivado” ou “removido”. Para o utilizador preencher o campo da *change notice* que está associada à revisão do objeto que está a editar tem de ir ao Visualizer e criar essa *change notice*.

6.2 Modelação do Novo Sistema

Inicialmente pretendia-se que o ProPEditor apresentasse um formulário e permitisse preencher unicamente os campos da legenda que estão ligados aos parâmetros dos modelos de CAD (*part* ou *assembly*). A razão para tal objetivo prendia-se com o facto de os automatismos estarem relacionados apenas com estes parâmetros, pois somente os valores dos parâmetros deviam ser sugeridos. Com o decorrer do desenvolvimento da aplicação e dado o tempo ainda disponível, decidiu-se melhorar a aplicação de forma a tornar possível a edição de toda a legenda através do formulário.

Desta forma, para além dos campos relatados no enunciado do ProPEditor, foi preciso acrescentar os seguintes campos: SUB-TIT.1 (subtítulo 1), DT.DES (data do desenho), MAT1. (material 1), PROT1. (proteção 1), UTIL.1 (utilização 1), UTIL.2 (utilização 2), UTIL.3 (utilização 3), P.FIN (artigo), PMG (desenho similar) e PMG1 (desenho similar 2).

Esta atualização trouxe bastantes benefícios pois com a primeira versão da aplicação os utilizadores tinham de fazer dois esforços para conseguir editar a legenda de um desenho: preencher a legenda no formulário e preencher os restantes campos diretamente no desenho. Com a nova aplicação o utilizador pode editar toda a legenda através do formulário de uma forma mais simples, rápida e apelativa, com a vantagem de ter alguns campos preenchidos, o que diminuí o tempo de realização desta tarefa. Apesar de esta atualização ter surgido numa fase tardia do processo de desenvolvimento do ProPEditor, esta é considerada nesta modelação.

Na modelação desta aplicação foi identificado o caso de uso *Editar Legenda*.

6.2.1 Caso de Uso: Editar Legenda

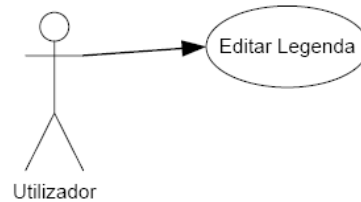


Figura 42 - Diagrama de casos de uso do ProPEditor

O ator deste caso de uso é o utilizador que interage com o programa Pro/E.

Formato Casual

O Pro/ENGINEER invoca a aplicação no seu navegador (*browser*) interno e se na sessão corrente do Pro/E estiver algum modelo ativo, a aplicação verifica qual o tipo do modelo. Se o tipo de modelo for um *drawing*, a aplicação obtém através de uma listagem os nomes dos campos da legenda que pretende e constrói o formulário. Para retornar os valores dos campos que estão associados a parâmetros, a aplicação obtém o primeiro modelo que tem o nome do desenho no seu prefixo e que está associado ao desenho e caso este modelo seja uma instância obtém o seu genérico. A aplicação lê os valores dos parâmetros requeridos a partir do modelo genérico obtido anteriormente. Os valores dos restantes campos que não estão associados a parâmetros são obtidos diretamente a partir da legenda do desenho.

A aplicação apresenta ao utilizador um formulário com o nome e valor atual dos campos a editar e, nos casos aplicáveis, a lista de valores que o respetivo campo pode assumir. A aplicação verifica se o desenho ou o modelo genérico ou as instâncias associadas ao genérico estão no Windchill e em caso afirmativo verifica se estão *checked out*. Caso algum destes modelos não esteja *checked out*, a aplicação não sugere valores para os parâmetros e não permite que o utilizador edite a legenda do desenho, bloqueando o formulário. Nas restantes situações, a aplicação sugere sempre o valor para o parâmetro *NUM_MOD* e determina se o objeto é novo. Se o objeto for novo, a aplicação sugere valores para os restantes parâmetros.

O utilizador edita os campos e no fim confirma a sua alteração. A aplicação efetua novamente as validações do Windchill e, caso haja permissão, atualiza os valores dos campos que não

estão associados a parâmetros e no caso dos campos que estão associados a parâmetros verifica se esses parâmetros estão no modelo. Caso não estejam, estes são criados com o valor inserido pelo utilizador; caso estejam, a aplicação verifica se estes foram alterados e se sim então este atualiza os seus valores.

O formato completo e as especificações suplementares deste caso de uso encontram-se na secção Apêndice E.

6.3 Aplicação Final

Caso a aplicação seja executada com sucesso, após o utilizador efetuar o *login* na sessão da intranet é apresentado o seguinte formulário com os campos da legenda para o utilizador modificar:

The screenshot shows the ProPEditor application interface. At the top left is a gear icon, and at the top right is the efacec logo. The main area contains a form with the following fields:

Modelo	33111031.PRT	-.1	RELEASED
Descrição Local	BARRAMENTO GERAL		PT
Descrição	MAIN BUSBAR		EN
Sub-designação	-	FLUOFIX GC	
Projectista	C.Jorge	C.Jorge	
Data Desenho	2011-01-17		
Aprovador			
Data Aprovação	2011-07-01		
Num. Modificação	CN00000190		
Data Change Notice	2011-06-17		
Material	Cu-ETP HB ISO 6958	-	
Dim. Material	32x5mm		
Protecção	-	-	
Utilização	33110624-01	-	-
Artigo	33111031-01		
Nível Qualidade	-		
Des. Similar	33109516-01	-	

At the bottom left of the form are 'Ok' and 'Reset' buttons.

Figura 43 - Interface da aplicação ProPEditor 1

6.3 Aplicação Final

Nesta aplicação, o parâmetro *Descrição Local* pode ser traduzido para inglês, espanhol ou português. Para traduzir é necessário selecionar a língua deste parâmetro (*pt*, *en* ou *es*) e do resultado da tradução (*pt*, *en* ou *es*). O resultado da tradução é apresentado pelo parâmetro *Descrição*. Esta funcionalidade foi implementada com a API Translate⁷ da Google.

Quando a aplicação sugere valores para os parâmetros, esta sugestão é feita da seguinte forma:

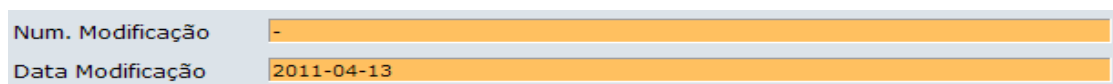


Figura 44 - Interface da aplicação ProPEditor 2

Após a modificação e seleção do botão *ok*, o ProPEditor pode apresentar os seguintes avisos:

- Sucesso: A aplicação coloca com fundo verde os campos que foram modificados com sucesso e, na ausência de erros, apresenta uma mensagem de sucesso. Nesta altura o utilizador pode salvar o seu desenho recorrendo ao menu do Pro/ENGINEER.

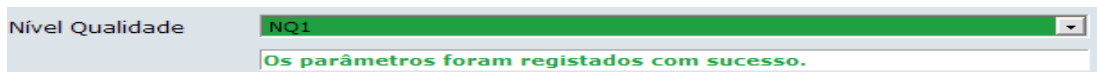


Figura 45 - Interface da aplicação ProPEditor 3

Como se pode verificar na figura anterior (*Figura 45*), o utilizador alterou com sucesso o campo *Nível Qualidade* e isso reflete-se na legenda do desenho.

⁷ Google Translate API: <https://developers.google.com/translate/?hl=pt-PT>

- Insucesso: Se pelo menos um valor de um campo não for modificado com sucesso, a aplicação coloca-o com fundo a vermelho e avisa o utilizador através de uma mensagem com o erro: “o(s) modelo(s) x não está(ão) *checked out*”; “o(s) campo(s) não aceita(m) valores desse tipo”; “ de momento o Pro/E não conseguiu modificar o valor do(s) campo(s)”.

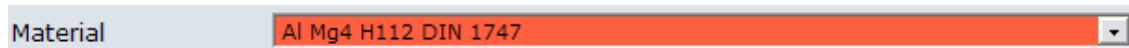


Figura 46 - Interface da aplicação ProPEditor 4

A qualquer momento o utilizador pode reiniciar a aplicação através da seleção do botão *reset*. Esta é uma funcionalidade que o próprio Pro/E disponibiliza aquando da edição dos parâmetros e funciona se o utilizador não validar as alterações através do botão “ok”.

6.4 Vantagens

Os principais benefícios do ProPEditor são:

- O suporte multilingue característico da intranet AMT (disponível em Português, Inglês e Espanhol) que torna os parâmetros facilmente perceptíveis para os utilizadores. Com a tradução dos campos da legenda, os utilizadores não têm de perceber algumas abreviaturas (como por exemplo, “Prot” que significa “Protecção” ou “Protection”) ou saber qual a informação a inserir em determinados campos que antes desta aplicação não possuíam título. Para além disso, os utilizadores da Argentina podem ter acesso à aplicação na sua língua oficial, o Espanhol;
- Sugestão de valores para determinados parâmetros. Com a sugestão automática da Change Notice associada à revisão do desenho, o utilizador não tem de consultar a aplicação Visualizer;
- Possibilidade de tradução instantânea do campo *descrição* para o campo *Description* do desenho através do uso da API Translate da Google.

6.5 Sumário

O ProPEditor foi disponibilizado aos colaboradores em Abril de 2011, tendo sido alvo de notícia no boletim informativo (*newsletter*) da Efacec AMT desse mês. Esta *newsletter* pode ser consultada na secção *Anexo 3*.

O ProPEditor permite um ganho superior a 1500 € por ano considerando que são preenchidas anualmente 7.500 legendas e que o ganho é de um minuto por cada legenda. Desta forma ganham-se 7.500 minutos de trabalho que correspondem a 125 horas. Se o valor da mão-de-obra for 12€ à hora serão poupados 1500€.

6.5 Sumário

Neste capítulo foi alcançado o objetivo *Implementar Aplicação ProPEditor*, sendo fornecidos três contributos.

Objetivo

- **Implementar Aplicação ProPEditor:** a aplicação ProPEditor, aplicação *Web* que possibilita o preenchimento da legenda de um desenho do Pro/ENGINEER através de um formulário, foi lançada para produção em Abril, sendo recebida com satisfação pelos colaboradores da Efacec.

Contributos

- Código PHP e JavaScript do ProPEditor;
- Manual de utilizador do ProPEditor em Português, Inglês e Espanhol, com a identificação das vantagens, regras das aplicações e descrição das suas funcionalidades;
- Documentação técnica do ProPEditor, para auxiliar os programadores do departamento de SI, com a apresentação do contexto do problema, o modelo de dados e descrição das funções implementadas. A esta documentação foi anexado o documento com a descrição dos casos de uso e de outras especificações do ProPEditor, desenvolvido na etapa da seleção da API e finalizado nesta fase.

7 ProCatalog

Este capítulo encontra-se dividido em cinco secções. A primeira secção **7.1 Contextualização** introduz a importância da necessidade da aplicação ProCatalog, explicando alguns conceitos relacionados com o processo de incorporação de um acessório num *assembly* do Pro/E efetuado na Efacec. A segunda secção **7.2 Modelação do Novo Sistema** apresenta o formato casual e completo do caso de uso *Adicionar Artigo* da aplicação ProCatalog. A terceira secção **7.3 Aplicação Final** apresenta a aplicação ProCatalog final, sendo identificadas as suas funcionalidades. A quarta secção **7.4 Vantagens** descreve as vantagens atingidas com o ProCatalog. A quinta secção **7.5 Sumário** identifica quais os objetivos atingidos e contributos fornecidos com as tarefas descritas neste capítulo.

O enunciado da aplicação ProCatalog é apresentado na secção **Anexo 2 - Enunciado do ProCatalog**.

As classes e respetivos métodos e propriedades utilizados na aplicação ProCatalog são apresentados na secção **Apêndice H**.

7.1 Contextualização

No âmbito do programa “Piloto Colombo Efacec”, o colaborador Rui Marinho sugeriu a implementação de uma aplicação para integrar com o Pro/E: um catálogo *Web* com alguns acessórios de fixação (porcas, parafusos, rebites, anilhas, cavilhas, golpilhas, *circlips* e freios) que possam ser consultados e incorporados nos modelos de CAD. Embora os artigos

pretendidos sejam os acessórios de fixação, a aplicação tem de estar preparada para suportar outro tipo de artigos.

O ProCatalog deve apresentar a informação dos acessórios que está armazenada na base de dados do GlobalArt. Contudo, a informação do GlobalArt não estava adequadamente estruturada para responder aos requisitos desta aplicação, uma vez que não possuía as características globais de cada acessório, e por isso foi necessário definir características e identificar os valores dessas características para cada um dos acessórios para migrá-los para o GlobalArt.

As equipas de Engenharia de Produto, Engenharia Industrial e de Inovação e Desenvolvimento de Portugal, Argentina e Índia utilizam o *software* Windchill, que está integrado com o Pro/E, para armazenar e gerir todos os objetos criados no Pro/E.

Cada objeto do Windchill corresponde a um artigo da aplicação GlobalArt, aplicação *Web* desenvolvida pelo departamento de SI da unidade AMT. Quando um novo objeto é criado no Pro/E e guardado no Windchill, o utilizador responsável pela sua modelação regista-o no ERP através da aplicação GlobalArt, sendo atribuído um código de artigo ao objeto. O GlobalArt tem, entre outras vantagens, a grande vantagem de evitar que códigos diferentes estejam associados ao mesmo artigo.

O Pro/E permite a incorporação de um acessório de fixação num *assembly* através da seleção da opção *Add componente to the assembly*, onde o utilizador deve escolher o acessório que pretende adicionar. Na Efacec os acessórios de fixação, componentes físicos que ligam dois ou mais objetos, estão reunidos no sistema Windchill. Contudo, a seleção do acessório não é feita de forma tão automática como se deseja e muitas vezes os utilizadores demoram horas para encontrar o artigo pretendido.

Para descobrir um artigo, os utilizadores podem utilizar o sistema de pesquisa do Windchill ou do GlobalArt. No sistema do Windchill, se o utilizador não se recordar do código do artigo, é necessário realizar uma pesquisa pela descrição do artigo, identificá-lo e pedir ao Pro/E para este adicionar ao *assembly*. Até recentemente a pesquisa do GlobalArt era feita da mesma forma do Windchill, através do código do artigo ou da descrição do mesmo. Devido à tarefa, realizada pelo autor, de definição das características globais dos acessórios de fixação do GlobalArt mais utilizados (consultar secção [2.4.1.1 Definição das Características Globais dos](#)

7.1 Contextualização

Acessórios de Fixação no GlobalArt), atualmente é possível pesquisar um artigo também por essas características. Esta evolução do GlobalArt diminuiu significativamente o tempo de pesquisa. Contudo, ainda eram necessárias outras melhorias pois o GlobalArt não apresenta uma interface gráfica tão apelativa como se deseja e não interage com o Pro/E no sentido de incorporar o artigo no Pro/E. Após encontrar o artigo pretendido no GlobalArt, ainda é necessário identificá-lo posteriormente no Windchill para que o Pro/E reconheça o objeto e o adicione ao *assembly*.

Quando o utilizador solicita ao Pro/E a incorporação de um determinado componente num *assembly*, o Pro/E coloca o componente num determinado ponto do *assembly* e adiciona-o temporariamente ao final da sequência de componentes da *model tree*.

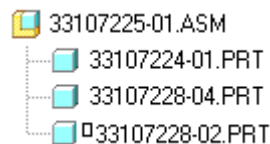





Figura 47 - Model tree

No exemplo da figura acima (*Figura 47*), o modelo “33107228-02.PRT” foi o novo modelo adicionado temporariamente à *model tree* como o último da sequência dos modelos.

O Pro/E apresenta uma caixa de diálogo para que este defina quais as restrições (juntar, alinhar) entre o novo modelo adicionado e os restantes componentes. Estas restrições permitem a montagem dos componentes através do estabelecimento das relações entre os vários componentes e das suas posições.

Na caixa de diálogo o utilizador pode confirmar as restrições efetuadas através da seleção do ícone  ou rejeitá-las com a seleção do ícone . Se o utilizador decidir anular as mesmas através da seleção do ícone , o componente é automaticamente eliminado da *model tree* e desaparece da janela do *assembly*. Se as restrições forem confirmadas o componente torna-se membro do *assembly*. A partir daí o utilizador pode fixar definitivamente o componente definitivamente no *assembly*, redefinir restrições ou até eliminar o componente.

7.2 Modelação do Novo Sistema

Na modelação desta aplicação foi identificado o caso de uso *Adicionar Artigo*.

7.2.1 Caso de Uso: Adicionar Artigo

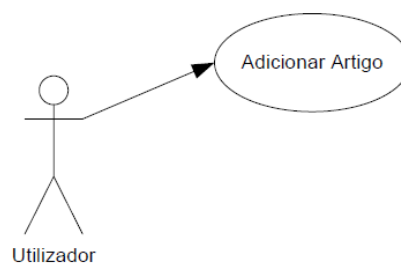


Figura 48 - Diagrama de casos de uso do ProCatalog

O ator deste caso de uso é o utilizador que interage com o programa Pro/E.

Formato Casual

O Pro/ENGINEER invoca a aplicação ProCatalog e esta apresenta um catálogo de artigos e o utilizador seleciona a classe do artigo. Após a seleção de uma classe todos os artigos dessa mesma classe são agrupados por uma determinada característica e após a escolha do grupo pretendido, é apresentada uma interface com a listagem de todos os artigos e das suas características globais. Nessa página, o utilizador pode filtrar os artigos pelas características de forma a encontrar o artigo desejado. Por fim, o utilizador solicita à aplicação que esta importe o modelo relacionado com o artigo selecionado para o seu modelo *assembly* do Pro/E.

O formato completo e as especificações suplementares deste caso de uso encontram-se na secção Apêndice F.

7.3 Aplicação Final

Na perspetiva do utilizador, a aplicação ProCatalog está dividida em quatro páginas. A primeira página apresenta as classes dos acessórios de fixação e ao selecionar uma delas o utilizador é remetido para a segunda página. A segunda página apresenta as normas que estão associadas aos artigos da classe anteriormente selecionada e ao escolher uma norma surge uma terceira página. A terceira página lista todos os acessórios de fixação, da classe e norma previamente selecionadas na primeira e segunda página, e as suas características globais, permitindo filtrar esses acessórios pelos valores dessas características. Qualquer uma destas páginas permite efetuar uma pesquisa global a todos os acessórios de fixação do GlobalArt. A quarta página apresenta os resultados desta pesquisa.

7.3.1 Primeira Página - Classes

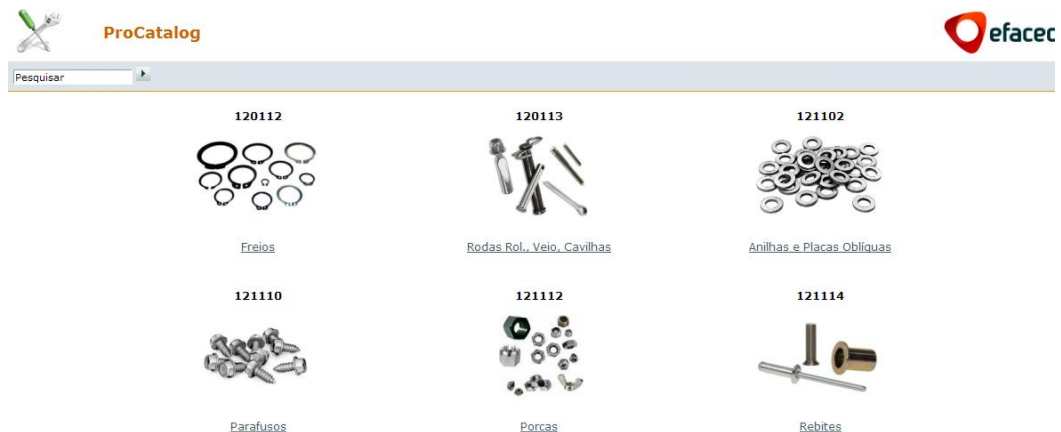


Figura 49 - Interface da aplicação ProCatalog 1

Após o *login* na sessão da intranet, é apresentada esta página com seis classes cujo código e nome são: 120112 – freios; 120113 – Rodas, Rol., Veio, Cavilhas; 121102 – Anilhas e Placas Obliquas; 121110 – Parafusos; 121112 – Porcas; e 121114 – Rebites. A aplicação tem flexibilidade para considerar mais ou menos classes consoante a necessidade dos utilizadores.

7.3.2 Segunda Página - Normas

Após o clique na imagem ou no título da classe, o utilizador é remetido para a página apresentada pela *Figura 50*, que agrupa a respetiva classe pelas várias normas que estão associadas aos artigos dessa classe no GlobalArt. Embora todas as classes estejam a ser agrupadas pela norma, a aplicação possibilita que cada classe seja agrupada por características diferentes, onde por exemplo os parafusos podem estar organizados pela norma e as anilhas podem estar organizadas pelo tipo de anilha.

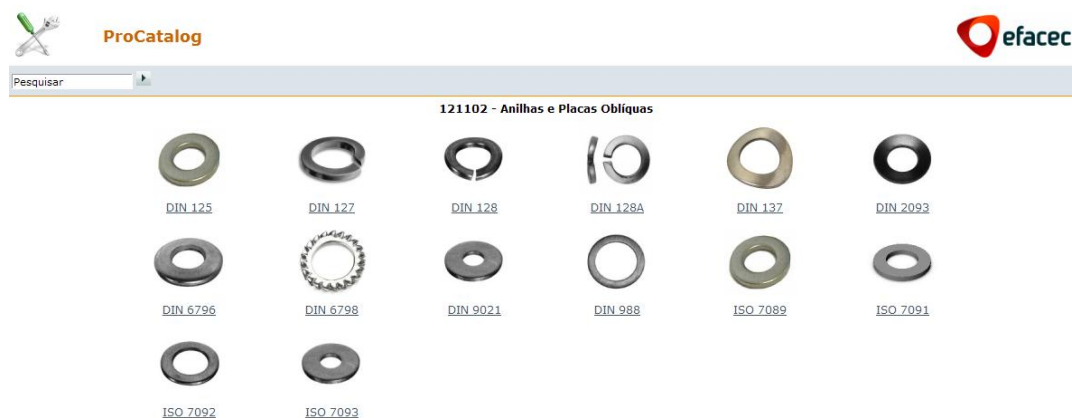




Figura 50 - Interface da aplicação ProCatalog 2

7.3.3 Terceira Página - Artigos

Ao clicar numa das normas é apresentada uma página com as características globais da classe e com os seus valores, e é mostrada uma listagem dos artigos dessa norma. Se um código de artigo estiver em várias companhias, a aplicação apresenta somente o artigo relacionado com uma das companhias, dando sempre prioridade à companhia ativa da sessão da intranet, não sendo apresentados artigos repetidos. Por questões de desempenho são listados apenas os primeiros 100 resultados.

7.3 Aplicação Final

Maria Joao Gamelas | Ajuda | Links | 453 Aparelhagem | Português | Sai

 **ProCatalog** 

Pesquisar

121102 - Anilhas e Placas Oblíquas DIN 6798

Diâmetro interior [mm] Tipo de anilha Matéria-prima

Revestimento sup.

Resultados da Pesquisa(11 registo(s))

















Companhia	Artigo		Descrição	Diâmetro interior [mm]	Diâmetro exterior [mm]	Espessura [mm]	Tipo de anilha	Classe qualidade acessório	Matéria-prima	Revestimento sup.	Norma adoptada
453	9041605		 ANILHA DEA 4 DIN 6798A	4			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041606		 ANILHA DEA 4 DIN 6798/V	4			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041607		 ANILHA DEA 5 DIN 6798A	5			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041608		 ANILHA DEA 5 DIN 6798/V	5			ITLW		Aço		DIN 6798
453	9041609		 ANILHA DEA 6 DIN 6798A	6			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041611		 ANILHA DEA 8 DIN 6798A	8			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041613		 ANILHA DEA 10 DIN 6798A	10			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798
453	9041619		 ANILHA DEA 16 DIN 6798/V	16			Anilha serrilhada (ext)		Aço	ZN20EC	DIN 6798

Figura 51 - Interface da aplicação ProCatalog 3

Os valores das características são apresentados sob a forma de uma lista de seleção, cuja função é filtrar a lista de artigos. Cada lista de seleção pertence a uma característica global e esta é apresentada ao utilizador se pelo menos um artigo daquela classe e norma possuir valores para essa característica. Neste exemplo, as características diâmetro exterior, espessura, classe qualidade acessório são filtros ocultos, o que significa que não há artigos com valores preenchidos para estas características. A característica pela qual a classe é agrupada (neste caso, a norma) nunca aparece uma vez que todos os artigos possuem o mesmo valor para essa característica e como tal não há vantagem em utilizá-la como filtro. Segundo o exemplo abaixo, todos os artigos pertencem à norma DIN 6798.



Figura 52 - Interface da aplicação ProCatalog 4

No exemplo acima, ao seleccionar um diâmetro interior de 5 mm e um tipo de anilha serrilhada, a aplicação lista o artigo da lista que possui estas características.

7.3.4 Funcionalidades

A terceira página possui algumas funcionalidades: as primeiras seis funcionalidades indicadas nesta secção podem ser acedidas a partir de qualquer *browser*, estando as restantes restringidas ao *browser* interno do Pro/E, nomeadamente as funcionalidades *Visualização dos Objetos Associados a um Artigo* e *Incorporação de objetos no Pro/E*.

7.3.4.1 Filtragem dos Artigos pelos Valores das Características

Os artigos são listados de acordo com os valores seleccionados pelo utilizador.



Figura 53 - Interface da aplicação ProCatalog 5

7.3 Aplicação Final

7.3.4.2 Visualização de um Artigo na Plataforma WebBaan


Através da seleção do ícone , o utilizador pode aceder ao WebBaan e consultar a informação de um artigo.



Figura 54 - Interface da aplicação WebBaan

7.3.4.3 Visualização da Imagem de um Artigo e do Respetivo Texto de Compra

A imagem do artigo pode ser visualizada através da colocação do cursor do rato sobre o artigo.

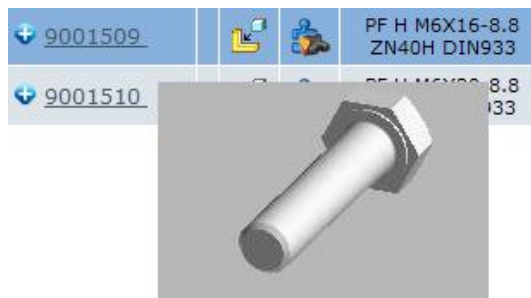


Figura 55 - Interface da aplicação ProCatalog 6

O texto de compra pode ser lido através da colocação do cursor do rato sobre o ícone .

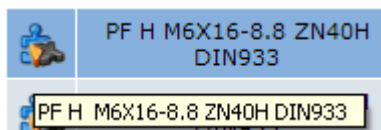



Figura 56 - Interface da aplicação ProCatalog 7

7.3.4.4 Visualização da Informação sobre as Normas

Esta informação adicional pode ser visualizada através da seleção do ícone .

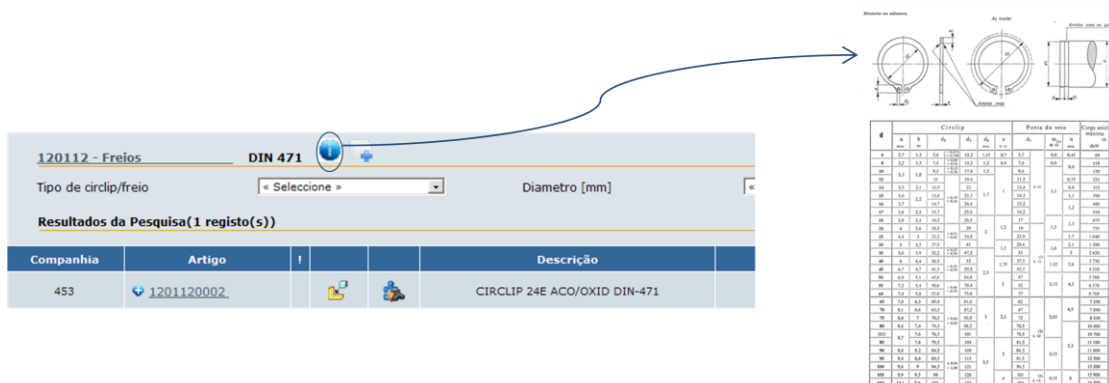



Figura 57 - Interface da aplicação ProCatalog 8

Nota: Quando as normas não possuem informação adicional, o ícone  não é apresentado ao utilizador.


7.3.4.5 Criação de um Novo Artigo na Plataforma GlobalArt

O ProCatalog reencaminha o utilizador para a página de criação de um novo artigo do GlobalArt.

7.3 Aplicação Final



Figura 58 - Interface da aplicação ProCatalog 9

Neste caso as características têm valores selecionados e se o utilizador clicar no ícone , o ProCatalog é redirecionado para a página do GlobalArt com as características, que foram previamente selecionadas no ProCatalog, preenchidas no GlobalArt:

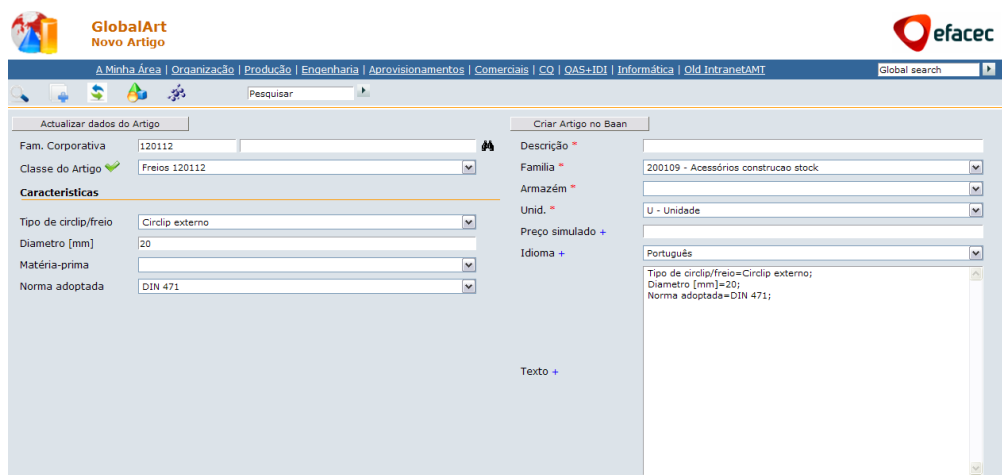



Figura 59 - Interface da aplicação GlobalArt

7.3.4.6 Verificação do Estado de um Artigo

O artigo pode não ser atualmente utilizado e o utilizador tem acesso a essa informação na coluna "!". Se o artigo estiver obsoleto aparece o ícone . Através da colocação do cursor do rato sobre esse ícone a aplicação apresenta a mensagem "obsoleto".




Companhia	Artigo			Descrição	Tipo de circlip/freio	Diametro [mm]	Matéria-prima	Norma adoptada
453	1201120002			CIRCLIP 24E ACO/OXID DIN-471	Circlip externo	24	Aço	DIN 471

Figura 60 - Interface da aplicação ProCatalog 10

7.3.4.7 Visualização dos Objetos Associados a um Artigo

Um artigo pode ter mais do que um objeto associado. Quando o objeto está no estado obsoleto é apresentado o ícone  .

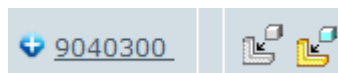






Figura 61 - Interface da aplicação ProCatalog 11

7.3.4.8 Incorporação do Objeto no Pro/E

A incorporação do objeto é feita através da seleção do ícone  . Se o objeto estiver obsoleto, o símbolo  não permite adicioná-lo ao *assembly* do Pro/E.

A partir deste momento o utilizador tem a opção de escolher as restrições pretendidas e de validar as mesmas através do clique nos ícones do Pro/ENGINEER   .

Nota: O utilizador tem de ter um modelo do tipo *assembly* ativo na sessão do Pro/ENGINEER.

7.3 Aplicação Final

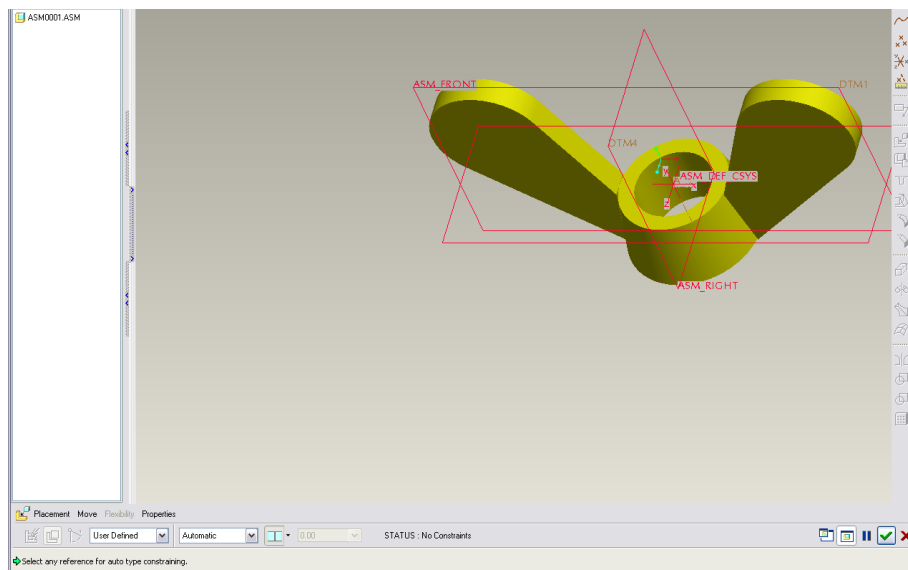


Figura 62 - Aplicação ProCatalog: incorporação de um acessório no Pro/E Wildfire 3.0

7.3.5 Quarta Página – Pesquisa Global

A qualquer momento o utilizador pode utilizar a pesquisa global que é feita pelo **código de artigo, descrição do artigo e valores das características globais do artigo**. A pesquisa é feita em todas as classes de artigo, independentemente se estas são ou não apresentadas pela página principal do catálogo.

Company	artclass	Article			Description
453	120112 - Freios	1201120002			CIRCLIP 24E ACO/OXID DIN-471
453	121110 - Parafusos	1211100018			nuno sol
453	121109 - Fixação - Outros Acessórios	9053004			CIRCLIP 6 E DIN-471
453	121109 - Fixação - Outros Acessórios	9053006			CIRCLIP CIRCLIP 8 E DIN-471
453	121109 - Fixação - Outros Acessórios	9053008			CIRCLIP 10E/ACO OXID NA-104
453	121109 - Fixação - Outros Acessórios	9053010			CIRCLIP 12 E DIN-471
453	121109 - Fixação - Outros Acessórios	9053012			CIRCLIP 14E/ACO OXID NA-104
453	121109 - Fixação - Outros Acessórios	9053014			CIRCLIP EXTERIOR
453	121109 - Fixação - Outros Acessórios	9053018			CIRCLIP 20E/ACO OXID DIN471
453	121109 - Fixação - Outros Acessórios	9053022			CIRCLIP 25E/ACO OXID NA-104
453	120113 - Rodas Rol., Veio, Cavilhas	9053026			CIRCLIP 30E/ACO OXID NA-104

Figura 63 - Interface da aplicação ProCatalog 12

Neste exemplo o utilizador inseriu a pesquisa “din 471” e a função de pesquisa separa as duas palavras “din” e “471” e procura-as individualmente, verificando se alguma destas palavras está contida num código de artigo, descrição de artigo ou nos valores das características globais do artigo, não sendo necessário haver uma correspondência exatamente igual. Aqui também não são listados artigos repetidos, utilizando-se o mesmo critério anterior: se um código de artigo estiver em várias companhias, é apresentado o artigo que está relacionado com uma das companhias, dando-se sempre prioridade à companhia ativa da sessão da intranet. Também por questões de desempenho são listados unicamente os primeiros 100 resultados.

A pesquisa global abrange algumas das funcionalidades da terceira página da aplicação: *Visualização de um Artigo na Plataforma WebBaan; Visualização da Imagem de um Artigo e do Respetivo Texto de Compra; Verificação do Estado de um Artigo; Visualização dos Objetos Associados a um Artigo; Incorporação do Objeto no Pro/E.*

7.4 Vantagens

As principais vantagens do ProCatalog são:

- Interface mais amigável de fácil navegação;
- Diminuição do tempo de procura e de incorporação de um artigo no Pro/ENGINEER devido à melhoria da organização da informação e da presença de imagens apelativas;
- Permite visualizar as características de vários artigos ao mesmo tempo, enquanto o GlobalArt apenas permite visualizar as características de um artigo de forma individual;
- Permite aceder a informação mais detalhada sobre uma determinada característica de um artigo. Nesta aplicação a característica escolhida foi a norma;
- Facilita o mecanismo de criação de um artigo na medida em que o ProCatalog interage com o GlobalArt para lhe fornecer as características previamente selecionadas;
- A organização da informação e a introdução de imagens permitem detetar mais facilmente os erros na classificação dos artigos.

7.5 Sumário



Figura 64 - Identificação de um parafuso na classe de artigo das porcas

- Suporte multilingue – disponível em Português, Inglês e Espanhol.

O ProCatalog foi disponibilizado aos colaboradores em Julho de 2011, tendo sido alvo de notícia no boletim informativo (*newsletter*) da Efacec AMT desse mês. Esta *newsletter* pode ser consultada na secção Anexo 4.

7.5 Sumário

Neste capítulo foi alcançado o objetivo *Implementar Aplicação ProCatalog*, sendo fornecidos três contributos.

Objetivo

- **Implementar Aplicação ProCatalog:** a aplicação ProCatalog, catálogo *Web* para a localização e incorporação de um acessório de fixação num modelo do tipo *assembly* do Pro/E, foi lançada para produção em Julho, sendo recebida com satisfação pelos colaboradores da Efacec.

Contributos

- Código PHP e JavaScript do ProCatalog;
- Manual de utilizador do ProCatalog em Português, Inglês e Espanhol, com a identificação das vantagens, regras das aplicações e descrição das suas funcionalidades;
- Documentação técnica do ProCatalog, para auxiliar os programadores do departamento de SI, com a apresentação do contexto do problema, o modelo de dados e descrição das funções implementadas. A esta documentação foi anexado o

documento com a descrição dos casos de uso e de outras especificações do ProCatalog, desenvolvido na etapa da seleção da API e finalizado nesta fase.

8 MassProp

Este capítulo encontra-se dividido em três secções. A primeira secção **8.1 Contextualização** introduz a importância da necessidade da aplicação ProPMass, explicando alguns conceitos relacionados com o processo de registo das propriedades de massa efetuado na Efacec. A segunda secção **8.2 Aplicação Final** apresenta a aplicação ProPMass final. A terceira secção **8.3 Sumário** identifica as tarefas realizadas e os contributos fornecidos com as tarefas descritas neste capítulo.

As classes e respetivos métodos e propriedades utilizados na aplicação ProPEditor são apresentados na secção **Apêndice I - MassProp: Classes Pro/Web.Link**.

8.1 Contextualização

Com esta aplicação pretende-se obter de uma forma mais rápida algumas propriedades de massa (massa, volume, densidade e área) e outros dados dos componentes de um determinado produto (*assembly*) para facilitar a tarefa dos engenheiros no registo destes dados numa folha de cálculo do Excel.

Devido à simplicidade desta aplicação e uma vez que a sua implementação não fazia parte dos objetivos, esta não foi modelada.

Os produtos da Efacec têm um ciclo de vida longo (20 anos ou mais) e por esta razão alguns modelos foram criados há 15 anos atrás em versões mais antigas do Pro/E e com outras regras. Nessa altura não havia a necessidade de ter a informação da massa e da área na legenda do

desenho e o modelo não tinha material definido. Há cerca de 6 anos atrás, foi decidido que essa informação era muito importante e o Pro/E foi configurado para a mostrar sempre que era criado um modelo novo e os *designers* foram informados para que sempre que criassem uma revisão de um modelo adicionassem essa informação. Os modelos são revistos se tiverem um erro ou se for necessário alterá-lo para o adequar a um novo produto, o que faz com que muitos modelos ainda não possuam a informação da massa dos materiais.

Com a inclusão de normas ambientais nos cadernos de encargos, muitos clientes solicitam informação sobre o ciclo de vida do produto em termos ambientais. Para fornecer esta informação é preciso obter uma listagem dos diferentes tipos de materiais que constituem o produto (aço, cobre, plástico) com as respetivas quantidades (massa em kg), multiplicar essas quantidades por um fator normalizado, em função do material, e assim obter o ciclo de vida.

Um produto corresponde a um modelo do tipo *assembly* e este pode ser constituído por vários modelos do tipo *part* ou *assembly*. Aos modelos que constituem um *assembly* o Pro/E designa-os por componentes. Quando o *assembly* é aberto na sessão todos os seus modelos são carregados para a sessão, não ficando visíveis ao utilizador. Os componentes são considerados tipos de *features* e as *features* podem estar em sete estados. Nesta aplicação apenas serão listados os componentes que estão no estado ativo (*feature* normal), inativo (*feature* não suprimida e que não está em uso por outra razão) e não regenerado (*feature* ativa e que teve um erro de regeneração) [PTC, 2006d].

O trabalho dos engenheiros passa por obter de forma manual através do Pro/E toda a informação necessária sobre os vários modelos que constituem o produto e por registá-la numa folha de cálculo do Excel para facilitar o eventual trabalho de correção dos dados e conseguir dar resposta aos clientes dizendo qual o ciclo de vida dos produtos.

Para obter as propriedades de massa de um determinado modelo é necessário ativá-lo no Pro/E e utilizar a opção *Model Analysis* do menu *Analysis* do Pro/E.

Como foi evidenciado, muitos objetos não possuem a informação requerida e passar por todos os modelos associados a um *assembly* é muito custoso. Por esta razão, com o objetivo de obter a informação necessária de forma mais rápida e sem ter que alterar os objetos, foi necessário criar uma aplicação para apresentar toda esta informação para que os engenheiros possam copiá-la diretamente para uma folha de cálculo do Excel.

8.2 Aplicação Final

Os dados a apresentar por esta aplicação são o nome, a versão e descrição do modelo, o material (indicação da norma.), a sua dimensão (espessura em milímetros cúbicos), o tipo de material (aço, cobre ou outros), o artigo a que o modelo está relacionado, a massa (quilograma), a área (milímetros quadrados), o volume (milímetros cúbicos), a densidade (quilogramas por milímetro cúbico) e as unidades da densidade dos modelos uma vez que estas podem não estar corretas. Caso algum modelo não possua uma massa válida, o utilizador pode corrigir a densidade em função do material e assim obter a massa correta dos objetos que ainda não tenham sido atualizados. O cálculo da massa é: densidade X volume. Estas tarefas requerem muita atenção por parte dos engenheiros pois os dados podem não estar bem.

8.2 Aplicação Final

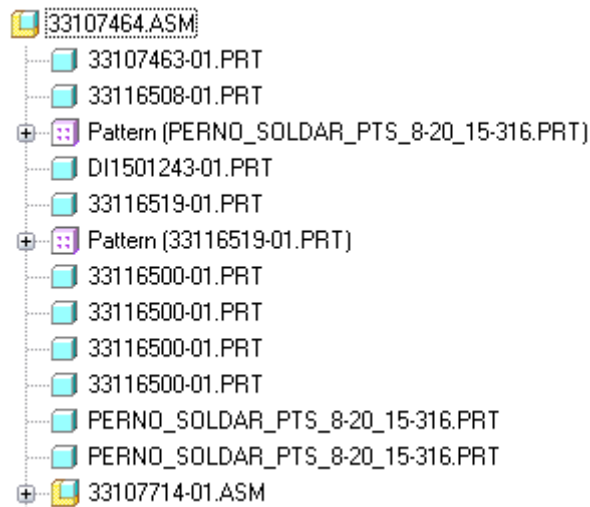


Figura 65 - Estrutura de um *assembly*

Após a ativação do *assembly* da figura acima na sessão do Pro/E e a invocação da aplicação é apresentado o seguinte resultado na janela do *browser*:

8 MassProp

Level	Name	Local Description	Material	Material Type	Material Dimension
0	33107464.asm				
1	... 33107463-01<33107463>.prt	PAINEL FRONTAL DA CUBA	X5 Cr Ni 18 10 DIN 17440		ESP 2mm
1	... 33116508-01<33116508>.prt	SUORTE P/ VALVULA	X5 Cr Ni 18 10 DIN 17440		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		ESP 2mm
1	... di1501243-01<di1501243>.prt	CASTANHA	X5 Cr Ni 18 10 DIN 17440		-
1	... 33116519-01<33116519>.prt	BARRA DE FIXAÇÃO	X5 Cr Ni 18 10 DIN 17440		Esp. 6 mm
1	... 33116519-01<33116519>.prt	BARRA DE FIXAÇÃO	X5 Cr Ni 18 10 DIN 17440		Esp. 6 mm
1	... 33116519-01<33116519>.prt	BARRA DE FIXAÇÃO	X5 Cr Ni 18 10 DIN 17440		Esp. 6 mm
1	... 33116500-01<33116500>.prt	CASTANHA	X5 Cr Ni 18 10 DIN 17440		D 16 mm
1	... 33116500-01<33116500>.prt	CASTANHA	X5 Cr Ni 18 10 DIN 17440		D 16 mm
1	... 33116500-01<33116500>.prt	CASTANHA	X5 Cr Ni 18 10 DIN 17440		D 16 mm
1	... 33116500-01<33116500>.prt	CASTANHA	X5 Cr Ni 18 10 DIN 17440		D 16 mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		D 16 mm
1	... perno_soldar_pts_8-20_15-316.prt	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		D 16 mm
1	... 33107714-01<33107714>.asm	PERNO P/ SOLDAR PTS.8-20/15-316	R St 37-2 DIN 17100		D 16 mm
2 33107695-01<33107695>.prt	REFORÇO DO PAINEL FRONTAL	X5 Cr Ni 18 10 DIN 17440		Esp. 2mm
2 di2901161-01<di2901161>.prt	PORCA M8 P/ SOLDAR	X5 Cr Ni 18 10 DIN 17440		Esp. 2mm

Figura 66 - Interface da aplicação MassProp (parte 1)

Version	Article	Volume	Area	Density	Unit	Mass
		1574492.38	1552832.35			
.0	-	1041229.54	1048317.26	0.0000079		8.2257
.0	33116508	16893.11	7454.4	0.0000079		0.1335
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	-	3567.77	1684.21	1		3567.7696
.0	33116519	4944.19	2669.4	0.0000079		0.0391
.0	33116519	4944.19	2669.4	0.0000079		0.0391
.0	33116519	4944.19	2669.4	0.0000079		0.0391
.0	33116500	7629.32	2428.37	0.0000079		0.0603
.0	33116500	7629.32	2428.37	0.0000079		0.0603
.0	33116500	7629.32	2428.37	0.0000079		0.0603
.0	33116500	7629.32	2428.37	0.0000079		0.0603
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	1045.31	722.19	1		1045.3077
.0	9021517	219318.37	222837.48			
.0	-	218541.26	222132.16	0.0000079		1.7265
.0	G0200AF010	777.11	705.32	1		777.1123

Figura 67 - Interface da aplicação MassProp (parte 2)

Deste modo são listados todos os componentes (*part* e *assembly*) do modelo *assembly* ativo na sessão do Pro/E e todos os dados necessários. Após esta listagem, o utilizador pode copiá-los diretamente para um documento Excel e verificar se estes cálculos estão corretos.

8.3 Sumário

Neste capítulo foi realizada a tarefa *Implementar Aplicação ProPMass*, sendo fornecido um contributo.

Tarefa

- **Implementar Aplicação ProPMass:** a aplicação ProPMass, que fornece as propriedades de massa (volume, área, massa e densidade) dos componentes de um assembly, foi imediatamente lançada para produção em Junho, sendo recebida com satisfação pelos colaboradores da Efacec.

Contributo

- O código PHP e JavaScript da aplicação MassProp.

9 Conclusões

A presente dissertação intitulada *Integração Intranet - Pro/ENGINEER Wildfire 3.0 via Pro/Web.Link* surgiu no contexto de um estágio curricular de oito meses realizado na unidade de Aparelhagem de Média e Alta Tensão (AMT) da Efacec, localizada em Leça do Balio, freguesia pertencente ao concelho de Matosinhos e distrito do Porto.

O estágio tinha como objetivos principais: ***selecionar e estudar uma das APIs do Pro/ENGINEER Wildfire 3.0***, o *software* de CAD desenvolvido pela PTC e utilizado pelas equipas dos departamentos de Engenharia de Produto, Engenharia Industrial, e de I&D (Inovação e Desenvolvimento) da Efacec, existentes em Portugal, Argentina e Índia; ***e implementar as aplicações Web ProPEditor e ProCatalog com a API selecionada*** para integrar a intranet da unidade AMT (SynergyNet) com o Pro/ENGINEER Wildfire 3.0. Estas aplicações foram sugeridas pelo projetista do departamento de Inovação e Desenvolvimento da Efacec Rui Marinho no contexto do programa Colombo, um programa da Efacec que estimula os seus colaboradores a expôr as suas próprias ideias para ajudar a melhorar os métodos de trabalho praticados na empresa e consequentemente a sua produtividade.

A aplicação ProPEditor consiste num formulário *Web* para automatizar a edição da legenda de um desenho do Pro/E e a aplicação ProCatalog consiste num catálogo *Web* para facilitar a incorporação de um acessório de fixação (parafuso, porca) num modelo *assembly* do Pro/E. Este catálogo deve apresentar os acessórios de fixação que estão registados no GlobalArt, uma plataforma de gestão de artigos desenvolvido na Efacec, e possibilitar o agrupamento e a filtragem dos acessórios de fixação por determinadas características globais (norma, diâmetro, etc.).

Para a implementação das aplicações ProPEditor e ProCatalog foi necessário cumprir dois objetivos adicionais: **organizar os dados dos acessórios de fixação no GlobalArt** para responder aos requisitos da aplicação ProCatalog uma vez que o GlobalArt não identificava as características globais dos acessórios de fixação, sendo necessário defini-las e organizar os dados dos acessórios de acordo com a nova estrutura; **estudar o Framework da intranet AMT (SynergyNet)** para conseguir implementar as aplicações do programa Colombo uma vez que era imperativo utilizar o *Framework* para o desenho da interface, para a ligação às bases de dados e para implementar o mecanismo de traduções característico da intranet AMT da Efacec, que permite visualizar as aplicações em Português, Espanhol e Inglês.

Pretendia-se essencialmente com esta dissertação **apresentar não só o trabalho desenvolvido no estágio**, como também **o resultado do estudo da API Pro/Web.Link, a API selecionada, compreendendo as regras e métodos de trabalho necessários para conseguir implementar uma determinada aplicação utilizando esta ferramenta**. Este estudo apoia-se sobretudo na documentação da PTC e foi realizado para complementar a bibliografia existente e ajudar outros programadores a utilizar esta ferramenta.

9.1 Objetivos Cumpridos e Contributos

Ao longo do trabalho foram cumpridos os seguintes objetivos, sendo fornecidos alguns contributos.

9.1.1. Organizar Dados dos Acessórios de Fixação no GlobalArt

Uma das funcionalidades da aplicação ProCatalog é agrupar os acessórios de fixação por uma determinada característica global. Sendo um dos requisitos desta aplicação obter os dados dos acessórios de fixação a partir da aplicação *Web GlobalArt*, foi necessário estruturar esta plataforma uma vez que esta não armazenava as características globais dos acessórios de fixação. Para tal, e assentando no livro das normas da Efacec, foram definidas no GlobalArt as características globais (ex: norma, matéria-prima, comprimento) para cada classe dos acessórios de fixação (anilha, cavilha, golpilha, circlip, freio, parafuso, porca e rebite) e foram identificados e migrados para esta plataforma os valores de todos os acessórios de fixação relativamente a cada uma destas características. Os valores foram inferidos a partir dos dados obtidos do Windchill PDMLink 9.0, um sistema que faz a gestão dos modelos do Pro/E que

9.1 Objetivos Cumpridos e Contributos

estão associados aos acessórios de fixação e outros artigos, uma vez que este continha uma descrição mais completa dos acessórios. Após esta tarefa foi escrito um documento com a classificação dos tipos de acessórios de fixação utilizados na Efacec para auxiliar os utilizadores a classificar os acessórios no GlobalArt.

Contributos:

- Estruturação dos acessórios de fixação na plataforma GlobalArt através da definição das suas características globais;
- Documento com a classificação dos acessórios de fixação para suportar os utilizadores na criação de artigos no sistema GlobalArt.

9.1.2. Estudar Framework da Intranet AMT (SynergyNet)

Os membros do departamento de SI da unidade AMT da Efacec estabeleceram que o ProPEditor e ProCatalog devem ser aplicações *Web* e como tal deviam ser utilizadas algumas funcionalidades do *Framework* da intranet AMT (SynergyNet) para: o desenho da interface; a ligação às bases de dados; a tradução das aplicações permitindo visualizá-las em Português, Inglês e Espanhol. Para que na fase de implementação das aplicações fosse possível estudar somente a API selecionada, reservou-se esta fase para o estudo do *Framework*. Este estudo foi feito com base na documentação do *Framework* e na implementação de duas aplicações que eram úteis para a Efacec, uma de geração dos portais dos departamentos da Efacec e outra para a edição desses mesmos portais. A implementação destas aplicações contribuiu bastante para o processo de aprendizagem do *Framework* uma vez que permitiu colocar em prática todas as funcionalidades requeridas.

Contributos:

- Duas aplicações em PHP para integrarem a intranet da unidade AMT: uma para a geração de portais dos vários departamentos (Produção, Informática, etc.) de cada companhia, e outra para a edição desses portais.

9.1.3. Selecionar API do Pro/ENGINEER

Uma vez que a PTC disponibiliza várias APIs (Pro/Toolkit, J-Link, VB e Pro/Web) para comunicar com o Pro/E foi necessário selecionar a mais adequada para implementar as aplicações do programa Colombo.

A Pro/Toolkit foi descartada desde início pois comporta um elevado custo de licenciamento. Embora a API VB esteja apenas disponível a partir da versão Wildfire 4.0, esta foi considerada uma vez que o *upgrade* para essa versão não era uma barreira caso esta fosse a melhor opção. Para compreender as restantes APIs e ao mesmo tempo auxiliar o processo de seleção resumiu-se numa tabela as características principais das APIs J-link, VB e Pro/Web.Link, e foram identificados quer os requisitos que a API devia cumprir com base nessa tabela, quer as funcionalidades relacionadas com o Pro/E com base na identificação e descrição dos casos de uso das aplicações.

No final do processo, a API sugerida foi a Pro/Web.Link e o departamento de SI concordou que esta é a mais adequada para o desenvolvimento das aplicações do programa Colombo uma vez que esta satisfaz todos os requisitos, permitindo implementar todas as funcionalidades requeridas, e suporta uma linguagem de programação (JavaScript) que o autor e os membros do departamento de SI dominam. O domínio desta linguagem permitirá desenvolver melhores aplicações e de uma forma mais rápida, sendo também vantajoso para a Efacec uma vez que não existirão barreiras de linguagem para qualquer membro do departamento se for necessário efetuar a manutenção das aplicações Colombo ou implementar novas funcionalidades que sejam solicitadas após este trabalho.

Contributos:

- Relatório sobre as APIs do Pro/E com informação essencial sobre as suas principais características para suportar o processo de seleção de uma API;
- Documentos com a descrição dos casos de uso das aplicações ProPEditor e ProCatalog e com outras especificações;
- Sugestão da API Pro/web.Link para o desenvolvimento das aplicações ProPEditor e ProCatalog.

9.1.4. Estudar API Seleccionada (Pro/Web.Link)

Inicialmente, este objetivo foi definido para implementar as aplicações do programa Colombo (ProPEditor e ProCatalog) durante o estágio na Efacec com a API Pro/Web.Link, a API sugerida pelo autor e aceite pelos membros do departamento de SI da Efacec como a mais adequada para o desenvolvimento destas aplicações. Contudo, decidiu-se que também devia ser feito um estudo mais aprofundado a esta API para complementar a literatura existente uma vez que a informação existente, maioritariamente disponibilizada pela PTC, suscita várias dúvidas aos programadores.

Durante o período de estágio foi priorizado o estudo dos tipos de classes, métodos e propriedades necessárias para implementar as aplicações, tendo sido também melhorado o módulo JavaScript da PTC, que contém as funções comuns a todas as aplicações *Web* desenvolvidas com a API Pro/Web, através da organização do seu código e da introdução de mensagens de erro. Esta versão melhorada do módulo JavaScript permite que o utilizador detete e corrija facilmente um erro uma vez que lhe são apresentadas as possibilidades de erro. Também foi feito um breve estudo ao processo de configuração do ambiente de desenvolvimento/execução das aplicações *Web* desenvolvidas com a API Pro/Web.Link e à distribuição destas pelos utilizadores pois este foi necessário para desenvolver as aplicações, melhorar o módulo JavaScript da PTC e distribuí-las pelos seus colaboradores. Somente numa fase posterior, fora do período de estágio, o estudo foi feito de forma mais aprofundada e com o propósito de elaborar procedimentos para apoiar outras pessoas que pretendam conhecer e utilizar a API Pro/Web.Link.

Após o estágio, completou-se o estudo da API Pro/Web.Link incidindo sobre: a sua arquitetura, sendo uma API constituída por um controlador ActiveX implementado num ficheiro DLL, não se pretendendo entrar em detalhes de como funciona a comunicação entre este e o cliente (*browser*); identificação de alguns componentes envolvidos na interação entre o *browser* interno e o Pro/E; os tipos de classes da API e das suas regras de utilização; a capacidade de herança dos seus objetos; algumas noções sobre a API *Wizard*, uma documentação com especificações da API Pro/Web.Link (identificação dos nomes e descrição das classes, métodos e propriedades, etc.); os detalhes do processo de configuração do ambiente de desenvolvimento, execução e distribuição de uma aplicação *Web* desenvolvida com a API Pro/Web.Link, de forma a clarificar as regras e metodologias adequadas ao uso desta

ferramenta, explicando alguns conceitos inerentes a este processo tal como, por exemplo, o porquê de ser necessário configurar as definições no *browser* IE instalado na máquina do utilizador para executar *scripts* no browser interno do Pro/E. Isso acontece porque o *browser* do Pro/E, construído com componentes do *browser* IE, interage com o IE do utilizador para verificar essas definições. Também foi proposto um método de trabalho para desenvolver e implementar aplicações com a API Pro/Web.Link com o intuito de auxiliar outros programadores.

Contributos:

- Versão melhorada do módulo JavaScript da PTC, que contém as funções comuns a todas as aplicações que sejam desenvolvidas com a API Pro/Web.Link, no que diz respeito à organização do código e ao tratamento de erros;
- Documentação sobre a API Pro/Web.Link com a seguinte informação: arquitetura da API; os tipos de classes que a constituem; o processo de configuração do ambiente de desenvolvimento e o *deployment* de uma aplicação; o módulo JavaScript para ser utilizado em todas as aplicações programadas com a Pro/Web.Link; identificação de um método de trabalho adequado para desenvolver aplicações com a API Pro/Web.Link; e outros conceitos relevantes que estão contidos nesta dissertação.

9.1.5. Implementar Aplicação ProPEditor e Aplicação ProCatalog

Com base na análise e descrição das acções executadas pelas aplicações ProPEditor e ProCatalog e da respetiva interação com o utilizador, as aplicações foram programadas recorrendo às seguintes tecnologias: Netbeans IDE; servidor Apache com suporte PHP; Microsoft SQL Server 2003; *Framework* da intranet AMT da Efacec; API Pro/Web.Link; e linguagens JavaScript, HTML, PHP, AJAX. O *Framework* da intranet foi utilizado para desenhar a interface da aplicação, para efetuar a ligação às bases de dados e para permitir a tradução da aplicação para Português, Inglês e Espanhol. A API Pro/Web.Link foi utilizada para desenvolver todas as funcionalidades relacionadas com a comunicação entre a aplicação *Web* e o Pro/ENGINEER Wildfire 3.0, tendo sido necessário configurar o ambiente de desenvolvimento com base nas regras desta API. Após a fase de desenvolvimento foram realizados alguns testes pelo autor e por alguns colaboradores da Efacec e obtendo uma versão funcional da mesma esta foi para produção.

9.1 Objetivos Cumpridos e Contributos

Para além destas aplicações também foi desenvolvida uma pequena aplicação, MassProp, que não fazia parte dos objetivos mas que foi requisitada pela Efacec uma vez que podia ser desenvolvida em apenas um dia. Esta aplicação fornece as *mass properties* (volume, área, massa e densidade) dos componentes de um *assembly* do Pro/E.

A aplicação ProPEditor foi a primeira aplicação do programa Colombo a ser desenvolvida e implementada na Efacec. Após o lançamento definitivo desta para produção, em Abril de 2011, iniciou-se o desenvolvimento da aplicação MassProp que demorou cerca de um ou dois dias a ser implementada, e de seguida foi desenvolvida a aplicação ProCatalog que foi lançada definitivamente para produção em Julho de 2011.

Os colaboradores da Efacec ficaram bastante satisfeitos com estas aplicações.

“As ferramentas desenvolvidas pela Maria João revelam muito potencial e abrem as portas a uma nova filosofia de interface com utilizador, tendo aplicação não só no contexto do projeto para o qual foram desenvolvidas como na produção”. – Rui Marinho, projetista do departamento de Investigação e Desenvolvimento da unidade AMT da Efacec.

“Esta nova ferramenta desenvolvida pela Maria João permite uma maior rapidez na procura de acessórios (anilhas, porca, etc.), bem como no preenchimento da legenda de um desenho. Facilita muito o nosso trabalho”. – Cristina Jorge, projetista do departamento de Engenharia do Produto da unidade AMT da Efacec.

Contributos:






- O código PHP e JavaScript das aplicações ProPEditor, MassProp e ProCatalog;
- Dois manuais de utilizador do ProPEditor e do ProCatalog em Português, Inglês e Espanhol, com a identificação das vantagens, regras das aplicações e descrição das suas funcionalidades;
- Documentação técnica do ProPEditor e ProCatalog para auxiliar os programadores do departamento de SI com a apresentação do contexto do problema, o modelo de dados, descrição das funções criadas e dos métodos e propriedades utilizados da API Pro/Web.Link. A esta documentação foram anexados os documentos, desenvolvidos durante o estágio, com a descrição dos casos de uso e de outras especificações.

9.2 Avaliação dos Objetivos

Ao longo deste trabalho foram realizadas várias tarefas para atingir os objetivos propostos. As contribuições fornecidas pelo autor desta dissertação durante e depois do estágio podem ser indicadores do cumprimento dos objetivos, contudo não são suficientes para determinar se os objetivos foram atingidos com êxito. Para a avaliação dos objetivos foram identificados e analisados alguns critérios de sucesso.

As tabelas seguintes apresentam os critérios de sucesso definidos para cada objetivo, identificando se estes foram ou não satisfeitos. Verifica-se que todos os objetivos foram cumpridos com sucesso.

Tabela 3 - Avaliação do objetivo *Organizar Dados dos Acessórios de Fixação*

<u>Objetivo</u>	<u>Critério de Sucesso</u>	 / 	<u>Observações</u>
Organizar Dados dos Acessórios de Fixação no GlobalArt	A migração dos dados dos acessórios foi efetuada sem erros para a base de dados do GlobalArt.		Foi necessário adaptar previamente o GlobalArt para migrar os dados com a nova estrutura.
	O documento com a classificação dos acessórios de fixação auxiliou positivamente os utilizadores na criação e classificação de artigos no GlobalArt.		Após a distribuição deste documento pelos colaboradores, estes não manifestaram qualquer dúvida na criação e classificação de artigos e não foram verificados erros de classificação após a introdução do novo método de criação de artigos no GlobalArt.
	A nova estruturação dos dados dos acessórios no GlobalArt contribuiu para o funcionamento esperado da aplicação ProCatalog.		A nova estruturação dos dados no GlobalArt permite que a aplicação ProCatalog consiga agrupar e listar os artigos por categorias (norma, classe, etc.) e filtrar os artigos por características.

9.2 Avaliação dos Objetivos

Tabela 4 - Avaliação do objetivo *Estudar Framework da Intranet AMT (SynergyNet)*






<u>Objetivo</u>	<u>Critério de Sucesso</u>	 / 	<u>Observações</u>
Estudar Framework da Intranet AMT (SynergyNet)	O estudo do <i>Framework</i> nesta etapa permitiu focar mais no estudo da API Pro/Web.Link na fase de desenvolvimento das aplicações ProPEditor e ProCatalog.		O estudo do <i>Framework</i> e a utilização prática dos seus métodos e propriedades nesta etapa, através do desenvolvimento das aplicações de geração e edição dos portais, possibilitou a apreensão das bases necessárias para mais tarde focalizar naquilo que realmente é importante: a API Pro/Web.Link.
	As aplicações de geração e edição dos portais conseguiram responder às necessidades requeridas.		Os responsáveis das páginas dos departamentos conseguem editá-las a qualquer momento, ficando estas disponíveis para os utilizadores em tempo real.
	As aplicações de geração e edição dos portais foram bem aceites pelos utilizadores.		Os utilizadores apreciaram o novo <i>layout</i> dos portais dos departamentos e, após as sessões de formação fornecidas pelos membros do departamento de sistemas de informação, adaptaram-se bem às funcionalidades destas novas aplicações.

Tabela 5 - Avaliação do objetivo *Selecionar API do Pro/ENGINEER*




<u>Objetivo</u>	<u>Critério de Sucesso</u>	 / 	<u>Observações</u>
Selecionar API do Pro/ENGINEER	O quadro comparativo das APIs contribuiu para a seleção de uma API adequada para satisfazer os requisitos das aplicações do projeto Colombo.		O quadro comparativo permite ajudar na seleção da API mais adequada para utilizar num determinado projeto, contudo a decisão não deve ser feita unicamente com base neste, sendo necessário analisar na documentação da API se esta possui todas as funcionalidades que devem ser implementadas.

Tabela 6 - Avaliação do objetivo *Estudar API Seleccionada (Pro/Web.Link)*












<u>Objetivo</u>	<u>Critério de Sucesso</u>	 / 	<u>Observações</u>
Estudar API Seleccionada (Pro/Web.Link)	O estudo permitiu compreender o essencial da API Pro/Web.Link.		
	O estudo permitiu complementar a literatura existente sobre a API.		
	O estudo permitiu o desenvolvimento e o <i>deployment</i> das aplicações		

Tabela 7 - Avaliação dos objetivos *Implementar Aplicação ProPEditor e ProCatalog*

<u>Objetivo</u>	<u>Critério de Sucesso</u>	 / 	<u>Observações</u>
Implementar Aplicação ProPEditor e ProCatalog	A aplicação foi desenvolvida dentro do limite de tempo		A aplicação foram lançadas dentro do período de estágio (Abril e Julho).
	A aplicação cumpre os requisitos especificados.		
	A aplicação tem utilidade para os colaboradores da Efacec.		Os colaboradores da Efacec estão muito satisfeitos com a aplicação.
	A documentação técnica do ProPEditor e ProCatalog tem utilidade para os membros do departamento de SI da Efacec.		

9.3 Limitações e Trabalho Futuro

O autor desta dissertação decidiu aceitar o desafio de estudar uma das APIs do Pro/ENGINEER e de implementar as duas aplicações requeridas com o objetivo de contribuir para o crescimento da Efacec com uma nova filosofia de trabalho.

9.3 Limitações e Trabalho Futuro

A realização do estágio curricular na Efacec foi uma experiência bastante positiva e produtiva uma vez que forneceu as condições de trabalho necessárias para a concretização dos objetivos propostos nesta dissertação. A colaboração da empresa e dos orientadores também foi fulcral para o cumprimento deste projeto. Apesar de grande parte do estudo da API Pro/Web.Link ter sido realizada após o estágio, o departamento de SI forneceu todas as ferramentas necessárias para que fosse possível continuá-lo.

Embora o autor não possuísse conhecimentos em PHP não houve muitas dificuldades em utilizar o *Framework* da intranet AMT. Não só o *Framework* está bem estruturado e documentado, como a linguagem PHP é bastante acessível. Em relação ao *software* de CAD Pro/ENGINEER Wildfire 3.0 e à ferramenta Windchill, existiram alguns obstáculos à compreensão de alguns conceitos mas estes foram ultrapassados devido ao enorme apoio do orientador Engenheiro Carlos Pinheiro. Os conhecimentos em relação à linguagem JavaScript facilitaram bastante o desenvolvimento das aplicações *Web* utilizando a API Pro/Web.Link.

A tabela com o resumo das características das APIs J-Link, VB e Pro/Web.Link podia conter outras características, contudo essas foram as características estudadas no estágio durante o prazo estabelecido. Podia ser estudada a possibilidade de executar uma aplicação com a API J-Link numa máquina cliente a partir de um servidor que tenha o Pro/ENGINEER. Isto seria útil, por exemplo, para executar uma aplicação sem ser necessário ter o Pro/E instalado no computador.

As aplicações desenvolvidas para a Efacec podem ser melhoradas, onde por exemplo a aplicação ProCatalog pode permitir incorporar ao mesmo tempo vários acessórios de fixação e definir automaticamente todas as restrições necessárias para que este fique montado no produto.

Pretendia-se compreender melhor como funciona a comunicação entre o Pro/ENGINEER e a API Pro/Web.Link, contudo a escassez de documentação e a reserva por parte da PTC em transmitir essa informação não o tornaram possível. Como trabalho futuro, sugere-se o estudo aprofundado da comunicação entre o Pro/E e a API Pro/Web.Link e o estudo da arquitetura da API Pro/Web.Link para perceber melhor como funciona a interação entre o *browser* e a API.

Na opinião do autor desta dissertação, o estudo efetuado à API Pro/Web.Link foi um passo importante para auxiliar outros utilizadores que pretendam conhecer e utilizar esta API uma vez que agrega toda a informação que existe disponível sobre a API Pro/Web.Link, explicando alguns detalhes, e fornece um método de trabalho de trabalho adequado para o desenvolvimento de aplicações com esta API. Pretende-se fundamentalmente com esta dissertação evitar ou atenuar as dificuldades que foram sentidas durante o desenvolvimento das aplicações ProPEditor e ProCatalog através da identificação de um método de trabalho e do fornecimento de algumas ferramentas: indicação e descrição dos passos do processo de configuração do ambiente de desenvolvimento; fornecimento de uma versão melhorada do módulo JavaScript da PTC. Pretende-se também que esta dissertação incentive o desenvolvimento de aplicações no contexto do CAD, não só na Efacec, como também noutras empresas.

Referências

- [Alexandre L., 2011] Alexandre L., 2011. Uma Abordagem de Alto Nível para a Verificação de Conteúdos na Web. Instituto Superior de Engenharia do Porto. Tese de Mestrado, <http://hdl.handle.net/123456789/95> [último acesso: Out 2012]
- [Asthana and Sinha, 1996] Asthana R.G.S., Sinha N.K., 1996. Computer Graphics for Scientists and Engineers. New Delhi, India.
- [Beck et al., 2001] Beck K., Beedle M., Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R., Mellor S., Schwaber K., Sutherland J., Thomas D. Manifesto for Agile Software Development: Principles behind the Agile Manifesto, <http://agilemanifesto.org/iso/en/principles.html> [último acesso: Out 2012]
- [Cisco, 2009] Cisco, Cisco Distributed Research and Development Solution Deployment Guide for PTC Windchill, http://www.cisco.com/en/US/docs/solutions/Verticals/Distributed_RD/dist_rd.html [último acesso: Out 2012]
- [Civa G., 2010] Civa G., PTC: Windchill em português e novo Pro/ENGINEER, <http://www.baguete.com.br/noticias/software/13/05/2010/ptc-windchill-em-portugues-e-novo-proengineer> [último acesso: Out 2012]
- [Conover S., 2005] Conover S., 2005. Making Pro/ENGINEER data accessible via the Web: Using J-Link and Pro/Web.Link as a component of Web applications, http://www.datajett.com/Cadd/jl_wl_test/web_01/Pro-ENGINEER,%20Windchill,%20and%20other%20PTC%20affiliated%20products%20%20Feature.htm [último acesso: Out 2012]
- [Daintith J., 2005] Daintith J., 2005. A Dictionary of Computing. Oxford.
- [Daughtry et al., 2009] Daughtry J., Farooq U., Myers B., Stylos J., 2009. API Usability:

Referências

- Report on Special Interest Group at CHI. SIGSOFT Software Engineering Notes, 34 (4): 27-29.
http://www.cs.cmu.edu/~NatProg/papers/SEN_DAUGHTRY_FAROOQ_MYERS_STYLOS.pdf [último acesso: Out 2012]
- [Efacec Switchgear, 2009] Efacec Swithgear, 2009. GlobalArt User's Guide.
- [Efacec, 2012a] Efacec, História,
http://www.efacec.pt/PresentationLayer/efacec_historia_00.aspx?idioma=1&local=13&area=1#2008 [último acesso: Out 2012]
- [Efacec, 2012b] Efacec, Efacec no Mundo,
http://www.efacec.pt/PresentationLayer/efacec_ctexto_00.aspx?idioma=1&local=6&area=1 [último acesso: Out 2012]
- [Efacec, 2012c] Efacec, Quem Somos,
http://www.efacec.pt/PresentationLayer/efacec_ctexto_00.aspx?idioma=1&local=5&area=1 [último acesso: Out 2012]
- [Efacec, 2012d] Efacec, Mercados,
http://www.efacec.pt/PresentationLayer/efacec_ctexto_00.aspx?idioma=1&local=50&area=8&mercado=36 [último acesso: Out 2012]
- [Efacec, 2012e] Efacec, Áreas de Negócio,
http://www.efacec.pt/PresentationLayer/efacec_ctexto_00.aspx?idioma=1&area=2&local=48 [último acesso: Out 2012]
- [Felco Solutions, Inc., 2011] Felco Solutions, Inc., 2011. Creo Parametric Application Programing Interface (API),
http://www.felcosolutions.com/creo_api.html [último acesso: Out 2012]
- [Garcia S., 2010] Garcia S, 2010. As tecnologias CAD no design de produto. Da ideia à sua materialização. Lisboa: FA. Tese de Mestrado,
<http://hdl.handle.net/10400.5/3002> [último acesso: Out 2012]
- [Kogent Learning Solutions, Inc. 2011] Kogent Learning Solutions, Inc., 2011. Pro/ENGINEER Wildfire 4.0 Essentials. Sudbury, Mass.: Jones and Bartlett Publishers.
- [Lewis C., 2008] Lewis C., 2008. A Pro/Engineers Guide to Pro/Web.Link.
- [Lewis C., 2009] Lewis C., Pro/ENGINEER Toolbox,
<http://www.proetoolbox.co.uk/Index.html> [último acesso: Out 2012]
- [Mozilla Developer Network, 2012] Mozilla Developer Network, 2012. XPCOM,
<https://developer.mozilla.org/en-US/docs/XPCOM> [último acesso: Out 2012]
- [MSDN, 2012a] MSDN, Microsoft Developer Network, 2012. COM Clients and Servers,
<http://msdn.microsoft.com/en-us/library/windows/desktop/ms683835%28v=vs.85%29.aspx>

Referências

- [último acesso: Out 2012]
- [MSDN, 2012b] MSDN, Microsoft Developer Network, 2012. Automation, <http://msdn.microsoft.com/en-us/library/dt80be78.aspx> [último acesso: Out 2012]
- [MSDN, 2012c] MSDN, Microsoft Developer Network, 2012. The Component Object Model, <http://msdn.microsoft.com/en-us/library/windows/desktop/ms694363%28v=vs.85%29.aspx> [último acesso: Out 2012]
- [MSDN, 2012d] MSDN, Microsoft Developer Network, 2012. Introduction to ActiveX Controls, http://msdn.microsoft.com/en-us/library/aa751972%28v=vs.85%29.aspx#ActiveX_Controls [último acesso: Out 2012]
- [MSDN, 2012e] MSDN, Microsoft Developer Network, 2012. Windows Internet Explorer: About the Browser, <http://msdn.microsoft.com/en-us/library/ie/aa741313%28v=vs.85%29.aspx> [último acesso: Out 2012]
- [MSDN, 2012f] MSDN, Microsoft Developer Network, 2012. DLLGetClassObject entry point, <http://msdn.microsoft.com/en-us/library/windows/desktop/ms680760%28v=vs.85%29.aspx> [último acesso: Out 2012]
- [Oliveira e Silveira, 2006] Oliveira C., Silveira P., 2006. Criação de uma Infra-Estrutura Computacional para Visualização de Informações Científicas 3D – Estrutura de Dados. Trabalho realizado para a obtenção do Bacharel em Engenharia de Computação, <http://www3.iesam-pa.edu.br/ojs/index.php/computacao/article/viewFile/86/81> [último acesso: Out 2012]
- [Pinheiro C., 2011] Pinheiro C., 2011. SharePlace. Instituto Superior de Engenharia do Porto. Relatório para a obtenção da Licenciatura em Engenharia Informática.
- [Poter C., 2000] Potter C., 2009. The CAID Connection, <http://www.cgw.com/Publications/CGW/2000/Volume-23-Issue-3-March-2000-/The-CAID-Connection.aspx> [último acesso: Out 2012]
- [PTC, 2002] Parametric Technology Corporation: Pro/ENGINEER® Wildfire™ Preproduction Release Notes, http://proe.cad.de/programme/preprod_relnotes.pdf [último acesso: Out 2012]
- [PTC, 2006a] PTC, 2006. Pro/ENGINEER® Wildfire® 3.0 Pro/TOOLKIT® User's Guide.

Referências

- [PTC, 2006b] PTC, 2006. Pro/ENGINEER® Wildfire™ 3.0 Embedded-browser Based, Pro/Web.Link® User's Guide.
- [PTC, 2006c] PTC, 2006. Pro/ENGINEER® Wildfire™3.0 Installation and Administration Guide.
- [PTC, 2006d] PTC, 2006. Pro/Web.Link Wildfire 3.0 API Wizard.
- [PTC, 2006e] PTC, 2006. Pro/ENGINEER® Wildfire™ 3.0 J-Link® User's Guide.
- [PTC, 2007a] PTC, 2007. Windchill® 9.0: Windchill® PDMLink™ User's Guide.
- [PTC, 2007b] PTC, 2007. Windchill® 9.0: Using Pro/ENGINEER® Wildfire™ with Windchill®.
- [PTC, 2009a] PTC, 2009. Windchill Workspaces – An Overview.mht.
- [PTC, 2009b] PTC, 2009. Pro/ENGINEER® Wildfire® 4.0 VB API User's Guide.
- [PTC, 2009c] PTC, 2009. Artigo da PTC Express: Creating a Family Table Driven Assembly UDF with Component Instances in Pro/ENGINEER Wildfire 4.0, http://www.imakenews.com/ptcexpress/e_article001469782.cfm?x=bfNsQTc,b3jsqcsB,w [último acesso: Out 2012]
- [PTC, 2012a] PTC, Creo Parametric, <http://www.ptc.com/products/creo/parametric> [último acesso: Out 2012]
- [PTC, 2012b] PTC, History and Acquisitions, <http://www.ptc.com/company/history-and-acquisitions.htm> [último acesso: Out 2012]
- [PTC, 2012c] PTC, Product Development System (PDS), <http://www.ptc.com/products/product-development-system.htm> [último acesso: Out 2012]
- [PTC, 2012d] PTC, Products, <http://www.ptc.com/products/index.htm> [último acesso: Out 2012]
- [PTC, 2012e] PTC, Product name updates for Pro/ENGINEER, coCreate and ProductView, <http://www.ptc.com/products/creo/product-mappings/index.htm> [último acesso: Out 2012]
- [PTC, 2012f] PTC, Enhancement Details – Visual Basic Application Programming Interface (API), http://www.ptc.com/appserver/wcms/relnotes/note.jsp?im_dbkey=51392&icg_dbkey=826 [último acesso: Out 2012]
- [Melodie S., 2009] Melodie S., 2009. Windchill PDMLink 9.1 Quick Reference Guide. Phoenix, Arizona: TriStar Inc., http://www.tristar.com/shop/qrg/Windchill_PDMLink_91_QR G.pdf [último acesso: Out 2012]

Referências

- [University of Cambridge, 2012] University of Cambridge. Using the “proep” Protocol. http://www.eng.cam.ac.uk/DesignOffice/cad/proewild3/usasci/proe/collaboration/using_the__proep__protocol.htm [último acesso: Out 2012]
- [VMware, 2008] VMware, 2008. Technical White Paper: Disaster Recovery for PTC Windchill Solutions with VMware® Site Recovery Manager. http://www.vmware.com/files/pdf/DR_for_PTC_Windchill_Solutions_Site_Recovery_Manager.pdf [último acesso: Out 2012]
- [Westin S., 1998] Westin S, 1998. Computer-aided Industrial Design. Cornell University. UACM SIGGRAPH Computer Graphics. Volume 32. <http://dl.acm.org/citation.cfm?id=279457&coll=portal&dl=ACM> [último acesso: Out 2012]
- [Zancul E., 2009] Zancul E., 2009. Gestão do Ciclo de Vida de Produtos: Seleção de Sistemas PLM com base em sistemas de referência, <http://www.teses.usp.br/teses/disponiveis/18/18140/tde-27052009-132444/en.php> [último acesso: Out 2012]

Referências

Anexos

Anexo 1 - Enunciado do ProPEditor

Preenchimento de Legendas nos Desenhos Pro/E - Ideia 366

Objectivo

Criar uma aplicação *Web* que facilite o preenchimento da legenda dos desenhos feitos em Pro/ENGINEER (Pro/E).

Processo

A aplicação inicia-se com o click de um ícone no Pro|E, este invoca uma página *Web* no *browser* integrado do Pro|E. Esta página vai permitir a edição da legenda dos desenhos do Pro|E através de um formulário sendo que alguns deles vão obedecer a listas predefinidas obtidas de uma base de dados.

Quando a aplicação é invocada deve ler o valor dos campos do desenho activo na sessão de Pro|E e obter os valores possíveis para os parâmetros que obedecem a listas e apresentar o formulário ao utilizador.

O utilizador preenche o formulário e após gravação, a aplicação regista nos parâmetros os valores preenchidos.

A aplicação deve implementar alguns automatismos para facilitar o preenchimento do formulário:

- Se o modelo é novo, então os parâmetros *PROJECTISTA* e *DESENHADOR* são automaticamente preenchidos com o nome de CAD associado ao utilizador corrente;
- O parâmetro *APROVADOR* deve ser obtido da configuração local do utilizador;
- O parâmetro *DATA_APROV* deve ser sugerido com o valor da data actual incrementada de n dias úteis consoante a configuração do utilizador.

Tabela 8 - Parâmetros do ProPEditor

Parâmetro	Tipo	Automatismo
DESIGNACAO	String – Livre (max. 80)	
DESCRIPTION	String – Livre (max. 80)	Tradução automática por API da Google.

SUB_DESIGNACAO	String – Livre (max. 80)	
MATERIAL	String – Lista de Elementos	
DIM_MATERIAL	String – Livre (max. 80)	
PROTECCAO	String – Lista de Elementos	
NQ	String – Lista de Elementos	
PROJECTISTA	String – Lista de Elementos	Automático para novos objetos.
DESENHADOR	String – Lista de Elementos	Automático para novos objetos.
APROVADOR	String – Lista de Elementos	Automático para novos objectos.
DATA_APROV	Date	Data atual.
NUM_MOD	String	Obtido da BD. Pode ser uma lista ou pode estar atribuído.
DATA_MOD	Date	Data atual.

O formulário deve ser multilingue (PT, EN e ES) usando o mecanismo de tradução da intranet.

Requisitos funcionais

Edição da legenda de um desenho do Pro/ENGINEER através de um formulário numa página *Web*.

Requisitos não funcionais

- A aplicação deve ser *Web-based*;
- A interface da aplicação deve ser feita em PHP utilizando o *Framework* da intranet
- A aplicação deve ser multilingue;
- A aplicação deve ser síncrona;
- A aplicação apenas deve executar no *browser* interno do Pro/ENGINEER;
- A aplicação deve usar a API Pro/Web.Link do Pro/ENGINEER;
- A aplicação deve funcionar na intranet AMT, utilizando o seu sistema de *login* e de traduções;
- O acesso às bases de dados MS SQLServer para leitura de dados deve ser feito com recurso ao *Framework* da intranet;
- Todas as funcionalidades da Pro/Web.Link devem ficar num módulo JavaScript para possibilitar a sua reutilização noutras aplicações.

Anexo 2 - Enunciado do ProCatalog

Alteração da Interface e Integração no Pro/E – Ideia 348

Objectivo

Criar um catálogo *Web* que facilite a localização de artigos e, no caso do Pro/ENGINEER (Pro/E), a incorporação destes nos modelos *assembly*. Este catálogo deve assentar na plataforma GlobalArt.

Processo

À semelhança dos catálogos eletrónicos da *Web*, pretende-se uma aplicação que através de uma interface gráfica permita facilmente localizar artigos em função da sua família e características. A aplicação mostrará inicialmente uma lista de imagens em que cada uma corresponde a uma família. Após click do utilizador sobre uma delas, serão mostrados todos os grupos dessa família e após novo click no grupo, será apresentada uma interface com as características dos artigos desse grupo permitindo configurar o artigo pretendido e, no caso de a página estar a ser visualizada no *browser* interno do Pro/E, permitirá incorporar o artigo clicando num botão.

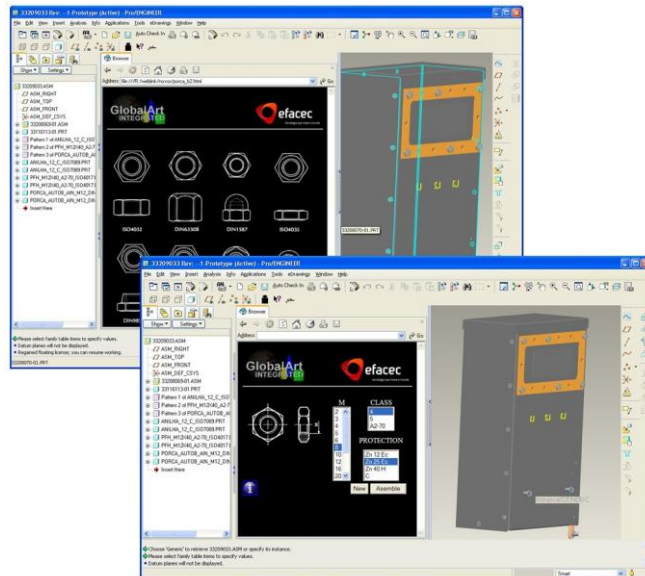


Figura 68 - Proposta de apresentação da interface do ProCatalog (repetição)

Requisitos Funcionais

- Pesquisa, seleção e incorporação de artigos a partir de uma página *Web*.

Requisitos Não Funcionais

- A fonte de informação deve ser o GlobalArt;
- A aplicação deve ser *Web-based*;
- A interface da aplicação deve ser feita em PHP utilizando o *Framework* da intranet;
- A aplicação deve ser multilíngue;
- A aplicação deve ser síncrona;
- A aplicação deve poder ser executada num browser externo ao Pro/E, no entanto as funcionalidades relacionadas com o Pro/E devem restringir-se ao *browser* interno do Pro/ENGINEER;
- A aplicação deve usar a API Pro/Web.Link do Pro/ENGINEER;
- A aplicação deve funcionar na intranet AMT, utilizando o seu sistema de *login* e de traduções;
- O acesso às bases de dados MS SQLServer para leitura de dados deve ser feito com recurso ao *Framework* da intranet;
- Todas as funcionalidades da Pro/Web.Link devem ficar num módulo JavaScript para possibilitar a sua reutilização noutras aplicações.

Anexo 3 - Newsletter da Efacec AMT de Abril de 2011

NÚMERO
06
Abril
2011



Synergy

JORNAL DE
SISTEMAS DE
INFORMAÇÃO
DA EFACEC AMT

Checkin - Notificação de recepção

A pedido da Eng. Industrial foi implementado na aplicação CheckIn, que gere a recepções de ordens de compra, mais uma importante funcionalidade. É agora enviada uma notificação ao requisitante sempre que é recepcionada uma ordem de compra originada por uma requisição. Este aviso evita que material requisitado fique perdido na recepção de materiais e que o requisitante tenha que fazer pesquisa por ordens de compra para verificar se a recepção foi realizada.

DB SCHENKER

Web Service - Stock amazém

Foi criado mais um *web service* que permite a obtenção online do stock físico no ERP (Baan) dos artigos no armazém externo associado a um abastecedor (neste caso ao parceiro Schenker, armazém 182025). Desta forma poderá comparar se os stocks no armazém da Schenker estão idênticos aos stocks no ERP. O ganho que foi estimado para esta funcionalidade ascende aos 2 240 €/ano.

GlobalART

Foi disponibilizada no GlobalART uma funcionalidade que permite ao Departamento de Compras registar informações do fornecedor do artigo e criar registos nos livros de preços.







Modelo	PISTON.PRT	Versão	Estado
Descrição Local	PISTON		PT
Descrição	PISTON		EN
Sub-designação			
Projectista	C.Pinheiro		
Desenhador	C.Pinheiro		
Aprovador			
Data Aprovação	11-04-2011		
Num. Modificação			
Data Modificação	30-04-2011		
Material			
Dim. Material			
Proteção			
Nível Qualidade			

Integração CAD com Intranet: Uma realidade em AMT

Os SI da Efacec AMT exploram com sucesso uma nova tecnologia de integração do software CAD (Pro/ENGINEER) com a SynergyNet e restantes sistemas da Efacec. Uma aplicação informática para auxiliar o preenchimento das legendas dos desenhos está já em funcionamento usando esta tecnologia denominada Pro/Web.Link e outra para auxílio da montagem 3D de acessórios de fixação está já em desenvolvimento.

O ProPEditor, software para edição de parâmetros no Pro/ENGINEER nasceu de uma ideia que o nosso colaborador [Rui Marinho](#) expôs no Colombo. Tem como objectivo principal agilizar a tarefa de preenchimento da legenda dos desenhos proporcionando vários automatismos quando esta está a ser criada.

Esta aplicação foi desenvolvida para interagir com o Pro/ENGINEER. Permite através de um formulário editar os parâmetros de objectos ("part", "drawing", "assembly", etc.). Os parâmetros CAD a alterar tornam-se facilmente perceptíveis para os utilizadores graças ao característico suporte multilingue das aplicações da SynergyNet. Uma das principais vantagens desta aplicação é a sugestão de valores para determinados parâmetros quando o objecto é novo, tais como:

- Sugestão do nome do projectista
- Sugestão do nome do desenhador
- Sugestão do nome do aprovador
- Sugestão da data de aprovação
- Sugestão da data de modificação

Outra vantagem do ProPEditor é a atribuição automática do valor da *Change Notice* nas revisões dos desenhos. Basta que o utilizador tenha previamente associado a revisão do desenho à respectiva *Change Notice* para que a aplicação automaticamente a coloque na legenda no campo correcto. Caso não tenha previamente efectuado essa associação, então ser-lhe-á apresentada uma lista com as suas *Change Notices* activas para que comodamente possa seleccionar a pretendida.

Esta aplicação possui ainda uma interessante e muito útil funcionalidade que permite a tradução automática da descrição do desenho de e para português, inglês ou espanhol usando os serviços do Google®. Esta aplicação permitirá um ganho de superior a 1500€ por ano mas mais importante ainda é o facto de ser uma alavanca para futuros desenvolvimentos usando a mesma tecnologia.

Para mais informações sobre esta aplicação consulte o [manual](#) do ProPEditor disponível na SynergyNet.

DIVULGAÇÃO

Nesta edição apresentámos-lhe a concretização de uma ideia de um nosso colega. Participe também com a sua experiencia e criatividade para, apresente as suas ideias e contribua assim de uma forma mais pró-activa para o sucesso da nossa empresa! Use o programa Colombo ou [escreva-nos](#).

Figura 69 - Newsletter da Efacec AMT de Abril de 2011

Anexo 4 - Newsletter da Efacec AMT de Julho de 2011

NÚMERO
09
JULHO
2011



Synergy

JORNAL DE
SISTEMAS DE
INFORMAÇÃO
DA EFACEC AMT

Comunicador 

O comunicador está a tornar-se um pilar fundamental da SynergyNet. Esta forma mais eficiente e racional de trocar mensagens, evitando emails, aumenta a produtividade dos utilizadores e a fiabilidade das aplicações. Está agora também disponível como uma aplicação autónoma. Aplicações como o iWork, iDevelopment ou o Geta tiram já pleno partido desta funcionalidade e as novas aplicações usarão também este sistema.

SharePlace 

Em breve estará totalmente disponível uma nova e importante aplicação que revolucionará a forma como são geridos documentos e ficheiros na SynergyNet. Esta aplicação irá substituir o "Arquivo Geral" que durante mais de 10 anos promoveu o acesso a toda a informação associada a propostas e encomendas contribuindo para um extraordinário aumento de produtividade de todos os colaboradores. Pode já ter um primeiro contacto com esta ferramenta acedendo à nova versão do Geta já disponível em Produção.

Bem vindos 

Damos as boas vindas ao estagiários João Vieira e Alberto Silva que ajudarão na configuração do GlobalArt.

DICAS E SUGESTÕES – *Esteja atento às mensagens do comunicador na Agenda da SynergyNet*

Sabia que sempre que tiver mensagens do comunicador a si destinadas elas aparecerão automaticamente na sua Agenda e aí permanecerão até que as marque como tratadas? Dê atenção à sua agenda sempre que se ligar na SynergyNet e marque como tratadas as suas mensagens logo que possível.



ProCatalog – Catálogo de artigos

Está já disponível na SynergyNet o ProCatalog. Esta aplicação consiste num catálogo electrónico de acessórios de fixação com um processo muito rápido e intuitivo para se encontrar o componente pretendido e com a possibilidade de o montar num conjunto CAD do ProEngineer com apenas um clique. Esta aplicação é fruto de uma ideia inserida no programa Colombo pelo nosso colega Rui Marinho e vem diminuir o tempo de procura e de montagem de acessórios de fixação no sistema de CAD.

Além de aumentar a produtividade dos utilizadores de CAD, o ProCatalog permite ainda que todos os utilizadores tenham acesso a um catálogo de acessórios muito intuitivo e global. Este catálogo é inteiramente baseado na plataforma GlobalArt pelo que dispensa uma manutenção própria. Sempre que se configurar um novo acessório de fixação (Parafuso, anilha, rebite, etc...) no GlobalArt ele automaticamente aparece no Catálogo. É de salientar que o catálogo abrange todas as companhias que usam GlobalArt. Este catálogo possui uma interface gráfica muito apelativa que permite ao utilizador uma fácil navegação e facilmente encontrar o que pretende poupando assim horas de trabalho.

O [ProCatalog](#) apresenta inúmeras vantagens, destacando-se:

- Interface muito amigável e de fácil navegação.
- Diminuição do tempo de procura e de montagem de um componente no Pro/ENGINEER devido à melhoria da organização da informação e da existência de imagens elucidativas.
- Permite visualizar todas as características de cada componente de uma forma mais directa.
- Permite aceder a informação mais detalhada sobre uma determinada norma.
- Não requer manutenção pois interage com o GlobalArt que lhe fornecer toda a informação necessária ao seu funcionamento.
- A organização da informação permite detectar mais facilmente eventuais erros que existam na classificação dos artigos.
- Suporte multilingue – disponível em Português, Inglês e Espanhol, podendo ser usado em qualquer companhia.

O ProCatalog pode ser acedido por qualquer utilizador autorizado, no entanto para utilizar as funcionalidades relacionadas com o Pro/ENGINEER a aplicação necessita de ser executada dentro Pro/Engineer usando o seu próprio navegador.

Para mais informações consulte o manual do ProCatalog ou contacte a equipa de AMT-SI.



Figura 70 - Newsletter da Efacec AMT de Julho de 2011

Apêndices

Apêndice A - Aplicação Geração dos Portais

A aplicação de geração dos portais dos departamentos gera de forma automática os portais a partir dos dados armazenados numa base de dados. Integrando esta aplicação na intranet da unidade da AMT, cada utilizador pode ver os portais com a informação sempre atualizada. A geração das páginas deve ser feita em tempo real porque a aplicação de edição dos portais deve permitir que a informação seja modificada a qualquer altura, sendo importante que os portais consultados reflitam todas essas alterações.

Um dos requisitos estabelecia que os portais devem ter o mesmo *layout* do Portal da Produção, o único portal recentemente renovado que tinha a mesma limitação pelo facto de não possibilitar a sua edição.

A *Figura 71* apresenta o *layout* a ser tomado como referência e realça os elementos que a aplicação deve estar preparada para receber e tratar.

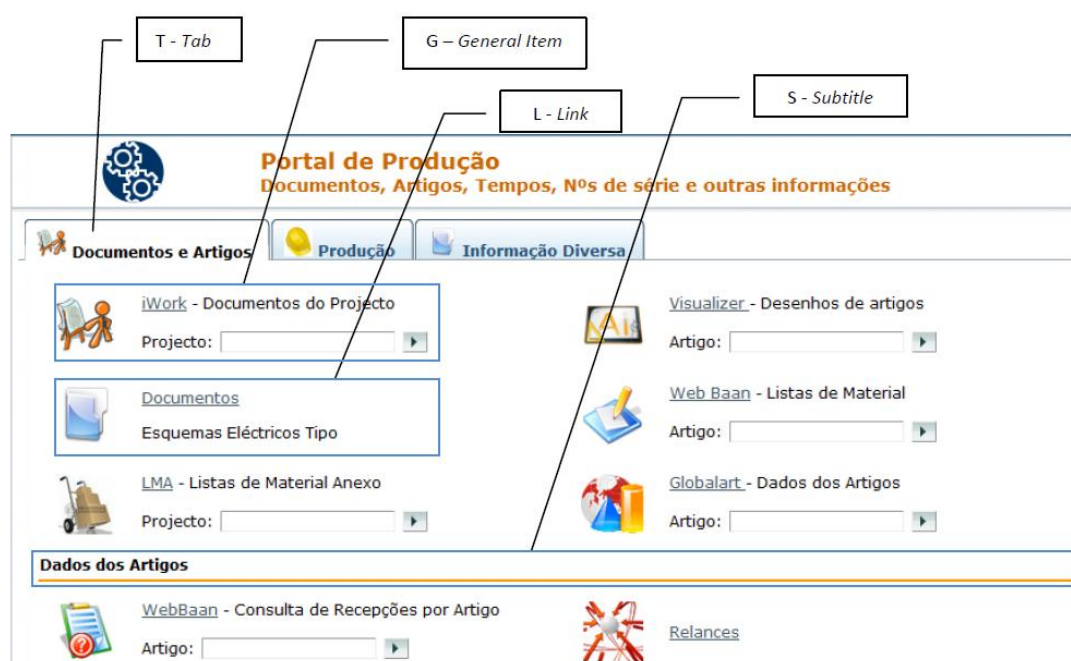


Figura 71 - Antigo portal do departamento de Produção

Cada portal tem de estar preparado para criar elementos dos seguintes tipos:

- a) **Tab**: Separador com imagem e título para definir uma nova página dentro do portal de um departamento.

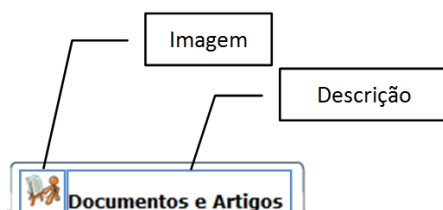


Figura 72 - Elemento *tab*

- b) **Link**: Hiperligação para uma determinada aplicação da SynergyNet, endereço de rede ou local, ou para outro local. Este elemento envolve a imagem, título e descrição da hiperligação.

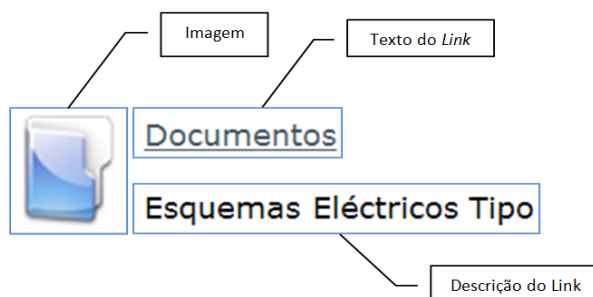


Figura 73 - Elemento *link*

Neste caso, se o utilizador seleccionar a hiperligação “Documentos” este será redirecionado para uma pasta na rede que guarda documentos com esquemas eléctricos.

- c) **General Item**: Atalho de pesquisa com o título do objeto de pesquisa e campo de pesquisa. Contém também a imagem, título e descrição da aplicação, referenciada pela hiperligação, onde a pesquisa é feita, caixa de texto e botão de pesquisa.

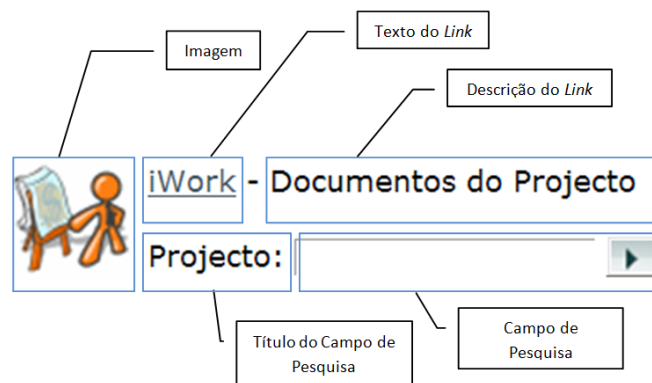


Figura 74 - Elemento *general item*

Com este campo de pesquisa, o utilizador consegue pesquisar, por exemplo, por um projeto na aplicação IWork. Após a seleção do botão de pesquisa, o utilizador é remetido para a página do IWork com os resultados da procura.

d) *Subtitle*: Subtítulo para organizar a informação dentro de um determinado separador.



Figura 75 - Elemento *subtitle*

O título do portal não foi levado em consideração porque a SynergyNet é que se preocupa com esse detalhe. Também ficou também definido que todos os textos dos portais devem suportar a opção multilingue da intranet.

Tendo em atenção estas considerações foi necessário identificar as classes e todos os atributos que armazenam os dados e que são invocados para a criação automática do portal. Após a análise foi identificada uma classe à qual se deu o nome de *tbpage_parameters*.

tbpage_parameters
company
tmodule
tseqn
torder
ttype
timage
tlink
tlinktext
tdesc
tlabel
tvariable
tpage
tmode
toperation
modifiedby
modifdate

Figura 76 - Classe *tbpage_parameters*

Esta entidade possui uma chave primária composta por três atributos: *company*, *tmodule* e *tseqn*. Em conjunto, estes três atributos identificam de forma única um registo da tabela, não havendo linhas com a mesma combinação de valores.

A entidade *tbpage_parameters* é constituída por 16 atributos que representam o seguinte:

- *company*: código da companhia em que o portal está inserido;
- *tmodule*: nome do departamento a que o portal está associado;
- *tseqn*: número inteiro que representa a ordem de entrada dos dados. Este é sempre único dentro de cada companhia;
- *torder*: ordem de exibição dos elementos (*tab*, *link*, *General item*, *subtitle*) de cada departamento;
- *ttype*: tipo de elemento: *tab*, *link*, *general item*, *subtitle*;
- *timage*: imagem associada ao elemento;
- *tlink*: endereço da hiperligação;
- *tlinktext*: nome da hiperligação;
- *tdesc*: descrição do separador (*tab*) ou do subtítulo (*subtitle*) ou a descrição da hiperligação (*general item* ou *link*);
- *tlabel*: título do campo de pesquisa (quando é um elemento *general item* ou *link*);
- *tvariable*: nome do campo de pesquisa da aplicação referenciada pela hiperligação;
- *tpage*: módulo da intranet associada à hiperligação (ex: GlobalArt). Este valor é definido pelo parâmetro *page* do URL da aplicação da hiperligação;

Apêndice A – Aplicação Geração dos Portais

- *tmode*: geralmente indica a funcionalidade do módulo da hiperligação. Este valor é definido pelo parâmetro *mod* do URL da aplicação da hiperligação;
- *toperation*: geralmente indica a operação dentro da funcionalidade do módulo da hiperligação. Este valor é definido pelo parâmetro *operation* do URL da aplicação da hiperligação;
- *modifiedby*: Número de identificação do colaborador que modificou o registo;
- *modifdate*: Data de modificação do registo.

Os campos obrigatórios são *company*, *tmodule*, *tseqn*, *torder*, *ttype*, *modifiedby* e *modifdate*. Enquanto o atributo *tseqn* é um número que nunca se repete, o atributo *torder* reinicia sempre que *tmodule* muda. Todos os restantes campos podem estar ou não preenchidos dependendo do tipo de elemento.

Uma vez que a aplicação de geração dos portais foi a primeira a ser programada e, como tal, foi imprescindível inserir alguns dados na base de dados para testar se esta funciona corretamente. Tendo como referência o portal da produção, foram inseridos os seguintes dados:

1	company	tmodule	tseqn	torder	ttype	timage	tlink	tlinktext
3	453	Producao	2	1	T	iwork.gif		
4	453	Producao	3	2	G	iwork.gif	http://intraamt.efacec.pt/intra/AMT.php?PAG=iwork&Mod=DOCSPROJ	iWork
5	453	Producao	4	3	G	Visualizer.png	http://intranetamt.efacec.pt/intra/amt.php?page=visualizer	Visualizer
6	453	Producao	5	4	L	folder_blue.png	\\s009\amt_arggeral\Documentos\Esquemas Tipo	Documentos
7	453	Producao	6	5	G	WBMatList.png	http://intranetamt.efacec.pt/intra/amt.php?page=wbmatlist	Web Baan
8	453	Producao	7	6	G	lma.jpg	http://intraamt.efacec.pt/intra/AMT.php?PAG=lma	LMA
9	453	Producao	8	7	G	globalart.gif	http://intraamt.efacec.pt/intra/AMT.php?PAG=artpage	Globalart
10	453	Producao	9	8	S			
11	453	Producao	10	9	G	WBPurchaseOrd.gif	http://intraamt.efacec.pt/intra/AMT.php?PAG=wbpurchaseord	WebBaan

Figura 77 - Dados do departamento de Produção (parte 1)

1	tdesc	tlabel	tvariable	tpage	tmode	toperation	modifiedby	modifdate
3	Documentos e Artigos						E1931	30-11-2010 09:16
4	Documentos do Projecto	Projecto	project	iwork	DOCSPROJ		E1931	30-11-2010 09:16
5	Desenhos de Artigos	Artigo	article	Visualizer			E1931	30-11-2010 09:16
6	Esquema Eléctricos Tipo						E1931	30-11-2010 09:16
7	Listas de Material	Artigo	item2	wbmatlist			E1931	30-11-2010 09:16
8	Listas de Material Anexo	Projecto	cprj	lma			E1931	30-11-2010 09:16
9	Dados dos Artigos	Artigo	Artigo	artpage			E1931	30-11-2010 09:16
10	Dados dos Artigos						E1931	30-11-2010 09:16
11	Consulta de Recepções por Artigo	Artigo	item2	wbpurchaseord			E1931	30-11-2010 09:16

Figura 78 - Dados do departamento de Produção (parte 2)

Apêndice A – Aplicação Geração dos Portais

Os dados colocados na base de dados constroem a seguinte página:

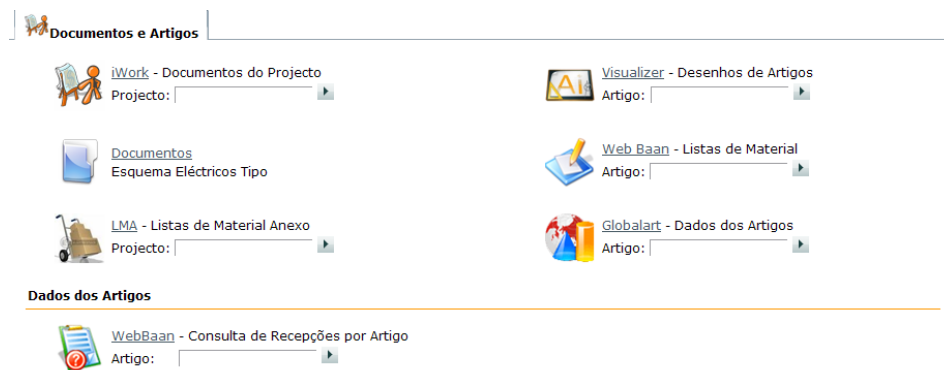


Figura 79 - Novo portal do departamento de Produção

Com a introdução de novos dados na base de dados, a aplicação de geração permite gerar, por exemplo, o seguinte portal do departamento de informática:



Figura 80 - Novo portal de informática

Apêndice B - Aplicação Edição dos Portais

As páginas dos departamentos podem ser editadas diretamente pelos utilizadores responsáveis, denominados gestores, através da seguinte aplicação:



Maria Joao Gamelas | Ajuda | Sair | Links ▼ | 453 AMT ▼ | Português ▼

A Minha Área 

A Minha Área | Organização | Produção | Engenharia | Aprovisionamentos | Comerciais | CQ | QAS+IDI | Informática | Old IntranetAMT Global search

Aplicação	Codigo do canal <input type="checkbox"/>	Descrição	Gestor
channels	engineering	Engenharia	Luis Tovar
channels	is	Informática	Luis Gomes
channels	production	Produção	António Matos

Figura 81 - Aplicação de edição dos portais dos departamentos 1

A página principal lista os canais (departamentos) a que o utilizador da atual sessão tem acesso. Neste caso, para testar a aplicação, havia acesso aos canais “engineer”, “is” e “production”.

Ao seleccionar o código do canal, o utilizador é direccionado para a seguinte página com os vários elementos que constituem o portal (*link, general item, subtitle, tab*):



Maria Joao Gamelas | Ajuda | Sair | Links ▼ | 453 AMT ▼ | Português ▼

A Minha Área #production# 

A Minha Área | Organização | Produção | Engenharia | Aprovisionamentos | Comerciais | CQ | QAS+IDI | Informática | Old IntranetAMT Global search




Ordem	Tipo	Imagem	Link	Título do link	Descrição do link	Etiqueta	Variável	Módulo	Modo	Alterado por	Data modificação	?
1	T				Documentos e Artigos					Paulo Sousa	2010-11-30 09:16:00	  
2	G		http://...	iWork	Documentos do Projecto	Projecto	project	iwork	DOCSPROJ	Paulo Sousa	2010-11-30 09:16:00	  
3	G		http://intraamt.efacec.pt/intra/AMT.php?PAG=iwork&Mod=DOCSPROJ	Artigos de Artigos		Artigo	search	Visualizer	10	Paulo Sousa	2010-11-30 09:16:00	  
4	L		http://...	Documentos	Esquema Eléctricos Tipo					Paulo Sousa	2010-11-30 09:16:00	  
5	G		http://...	Web Baan	Listas de Material	Artigo	gsearch	wbmatlist	gsearch	Luis Gomes	2011-01-05 17:29:00	  
6	G		http://...	LMA	Listas de Material Anexo	Projecto	cprj	lma		Paulo Sousa	2010-11-30 09:16:00	  
7	G		http://...	Globalart	Dados dos Artigos	Artigo	gsearch	globalart	gsearch	Luis Gomes	2011-01-26 12:03:00	  
8	S				Dados dos Artigos					Paulo Sousa	2010-11-30 09:16:00	  
9	G		http://...	WebBaan	Consulta de Receções por Artigo	Artigo	item2	wbpurchaseord		Paulo Sousa	2010-11-30 09:16:00	  
10	L		http://...	Relances				WBSearch		Paulo Sousa	2010-11-30 09:16:00	  
11	L		http://...	Vales de Material				valemat		Paulo Sousa	2010-11-30 09:16:00	  


Figura 82 - Aplicação de edição dos portais dos departamentos 2




Apêndice B – Aplicação Edição dos Portais

A partir deste momento, é possível visualizar, editar, apagar ou alterar a ordem dos seus elementos. Estes estão organizados por ordem crescente do campo “ordem” e, através da seleção das setas  , é possível alterar de forma automática essa mesma ordem.


1	T				Documentos e Artigos					Paulo Sousa	2010-11-30 09:16:00	    
2	G		http://...	iWork	Documentos do Projecto	Projecto	project	iwork	DOCSPROJ	Paulo Sousa	2010-11-30 09:16:00	    
3	G		http://intraamt.efacec.pt/intra/AMT.php?PAG=iwork&Mod=DOCSPROJ	inhos de Artigos	Artigo	search	Visualizer		10	Paulo Sousa	2010-11-30 09:16:00	    

Figura 83 - Aplicação de edição dos portais dos departamentos 3

Neste caso, se a seta  da linha com ordem 2 for seleccionada, esta passa a ter a ordem da linha abaixo (ordem 3) e a linha abaixo passa a ter ordem 2. Ao seleccionar as setas o elemento seleccionado fica com a ordem do elemento que está acima (se a seta seleccionada estiver virada para cima) ou abaixo (se a seta seleccionada estiver virada para baixo).

Ao escolher as opções de visualização , edição , ou de criação de um novo elemento , que estão sempre visíveis no menu principal da aplicação, o utilizador é remetido para a seguinte página:

Maria Joao Gamelas | Ajuda | Sair | Links ▼ | 453 AMT ▼ | Português ▼

A Minha Área
#production# - Alterar 

A Minha Área | Organização | Produção | Engenharia | Aprovisionamentos | Comerciais | CQ | QAS+IDI | Informática | Qld IntranetAMT

[Mostrar todas as imagens](#) [Mostrar nomes das imagens](#)

Guardar Anterior

Canal * Tipo *

Ordem * Imagem *

Link * #linktext# *

Descrição do link Etiqueta

Variável * Módulo *

Modo Operações
























Ordem	Tipo	Imagem	Link	Título do link	Descrição do link	Etiqueta	Variável	Módulo	Modo	Alterado por	Data modificação	?
1	T				Documentos e Artigos					Paulo Sousa	2010-11-30 09:16:00	    
2	G		http://...	iWork	Documentos do Projecto	Projecto	project	iwork	DOCSPROJ	Paulo Sousa	2010-11-30 09:16:00	    
3	G		http://intraamt.efacec.pt/intra/AMT.php?PAG=iwork&Mod=DOCSPROJ	inhos de Artigos	Artigo	search	Visualizer		10	Paulo Sousa	2010-11-30 09:16:00	    
4	L		Vs009	Documentos	Esquema Eléctricos Tipo					Paulo Sousa	2010-11-30 09:16:00	    

Figura 84 - Aplicação de edição dos portais dos departamentos 4

Apêndice B – Aplicação Edição dos Portais

Esta página permite visualizar os conteúdos de um elemento, onde o utilizador apenas tem permissões de leitura, editar os conteúdos dos elementos e inserir novos elementos de um departamento através do preenchimento dos vários campos. Quando se seleciona o tipo do campo (*tab*, *link*, etc.) a aplicação mostra os campos que são necessários preencher para esse tipo.

Apêndice C - Configuração do Ambiente Desenvolvimento e Deployment

1. Configuração do Ambiente de Desenvolvimento e de Execução em Ambiente Windows

As aplicações que interagem com a API Pro/Web.Link podem ser programadas com qualquer editor de código e não necessitam de configurações especiais.

Para executar estas aplicações *Web* no *browser* IE do Pro/E, o programador tem de cumprir três passos:

a) Registo do controlador ActiveX desenvolvido pela PTC:

O controlador ActiveX “pfcscm.dll”, desenvolvido pela PTC e considerado como “não seguro” uma vez que a PTC não limitou as suas capacidades, constitui a API Pro/Web.Link [PTC, 2006b]. Na versão do Pro/E Wildfire 3 para o Windows NT, este encontra-se no diretório “<diretório do Pro/E>\i486_nt\lib\pfcscm.dll”.

O controlador ActiveX, particularidade do sistema operativo Windows, é baseado na tecnologia COM que é usada para criar componentes reutilizáveis e é um tipo de servidor de Automação que expõe as suas funcionalidades às outras aplicações. Os controladores ActiveX têm de suportar o auto registo através da implementação das funções DllRegisterServer e DllUnregisterServer”. No sistema operativo UNIX, como o *browser* embutido é Mozilla, em vez do controlador ActiveX utiliza-se a tecnologia XPCOM que é semelhante ao modelo COM da Microsoft e foi criada pela Fundação Mozilla.

A instalação do Pro/E não abrange o registo do controlador ActiveX da PTC e sem este, e sem outras configurações que são descritas mais à frente, não se consegue aceder às classes da API Pro/Web.Link.

Para registar o controlador ActiveX ou anular o seu registo é obrigatório ter privilégios de administração porque estes processos implicam criar ou eliminar chaves dentro da chave pré-definida “HKEY_CLASSES_ROOT” do registo do Windows para que os seus serviços fiquem disponíveis [PTC, 2006b]. A Chave “HKEY_CLASSES_ROOT” é um atalho para a chave “HKEY_LOCAL_MACHINE\Software\Classes”.

O registo do controlador ActiveX “pfcscm.dll” pode ser feito de dois modos :

- **Automático:** Se o Pro/ENGINEER for executado como administrador e o controlador ActiveX `pfcscm.dll` não estiver registado no Windows, este é registado automaticamente assim que o Pro/E iniciar. A PTC diz que é necessário ter no seu ficheiro de configuração (`config.pro`) a opção `web_enable_javascript` ativa mas verificou-se que isso não corresponde à realidade [Lewis C., 2009].
- **Manual:** Executar a linha de comandos como administrador e inserir o comando **Regsvr32 “<diretório do controlador ActiveX>\pfcscm.dll”**. Se a linha de comandos não for executada como administrador este comando provoca uma falha na chamada à função `DllRegisterServer`.

A anulação do registo do controlador ActiveX “`pfcscm.dll`” tem de ser feita de forma manual e com permissões administrativas através da inserção do seguinte comando na linha de comandos:

Regsvr32 /u “<diretório do controlo ActiveX>\pfcscm.dll”

Se a linha de comandos não for executada como administrador este comando provoca uma falha na chamada à função `DllUnregisterServer`.

Cada versão do Pro/E (Pro/E Wildfire 3, Pro/E Wildfire 4, etc.) possui um diferente controlador ActiveX e por isso quando diferentes versões do Pro/ENGINEER estão instaladas no mesmo computador deve-se certificar que o Pro/E deteta o registo correspondente à sua versão antes de inicializar. Por esta razão é aconselhável que haja unicamente um controlador ActiveX do Pro/E registado no Windows e que antes de iniciar uma versão do Pro/E se anule manualmente o último registo efetuado para que este efetue o registo do controlador mais adequado [PTC, 2006b].

b) Edição do ficheiro de configuração do Pro/E (`config.pro`) de forma a ativar a opção `web_enable_javascript`:

Esta opção permite ativar (*on*) ou desativar (*off*) a API `Pro/Web.Link` no *browser* embutido do Pro/E. Para ativar esta opção coloca-se o texto “`web_enable_javascript on`” no ficheiro `config.pro`. Se esta opção não for apresentada no ficheiro de configuração significa que esta está desativa, sendo o seu valor por defeito. Para que o `Pro/Web.Link` e o Pro/E consigam comunicar é obrigatório que esta opção esteja ativa. Após a edição do ficheiro de

configuração o utilizador deve fechar e voltar a abrir o Pro/E pois é na fase de carregamento do Pro/E que este faz a leitura do ficheiro “config.pro”.

Ao ficheiro de configuração podem ser adicionadas outras opções disponibilizadas pelo Pro/E que definem o seu comportamento perante diversas situações relacionadas com as permissões das APIs, operações de modelação, com os desenhos, as cores. Embora múltiplos ficheiros “config.pro” possam coexistir e ser colocados noutros diretórios, na fase de desenvolvimento da aplicação opta-se por colocar o ficheiro no diretório local “<directório do Pro/E>\text” para tornar o processo de leitura mais rápido uma vez que é o primeiro local onde o Pro/E o procura por omissão.

O ficheiro “config.pro” pode ser editado a partir de um editor de texto ou a partir da opção “options” do menu “tools” do Pro/E.

d) Configuração das definições de segurança do *browser* Internet Explorer:

Para que o *browser* permita a execução de scripts e a interação entre o controlador ActiveX da PTC e a aplicação *Web (script)*, é necessário configurar as definições de segurança para a zona de alojamento da aplicação *Web* (Internet, Intranet Local, Sites fidedignos ou Sites restritos) em **Ferramentas** (Menu do IE)> **Opções da Internet** (opção)> **Segurança** (separador)> “<Zona Pretendida>” >**Personalizar nível...** (botão).

As definições que devem estar ativas são as seguintes [PTC, 2006b]:

- **Processamento de scripts ativo** (do grupo “Processamento de Scripts”);
- **Inicializar e efetuar o script de controlos ActiveX não assinalados como seguros para o processamento de scripts** (do grupo “Controlos e extensões ActiveX”). Cabe ao programador selecionar a opção “ativar” (ou “activate”) ou “pedir” (ou “prompt”) se desejar que outros scripts não seguros não sejam executados sem a autorização do utilizador.

Caso se decida que estas definições sejam aplicadas somente à aplicação *Web* que vai comunicar com o Pro/E através da API Pro/Web.Link e não a todas as aplicações da zona internet ou intranet por razões de segurança, é necessário adicionar o endereço dessa aplicação à zona dos “Sites fidedignos” em **Ferramentas** (Menu do IE)> **Opções da Internet** (opção)> **Segurança** (separador)> **Sites fidedignos**> **Sites** (botão) e devem ser personalizados os níveis de segurança para essa zona.

2. Distribuição da Aplicação (*Deployment*)

Para distribuir a aplicação pelos vários colaboradores de uma empresa é necessário fornecer-lhes acesso à aplicação *Web* e assegurar que os computadores têm as mesmas configurações do ambiente de desenvolvimento que foram referidas anteriormente. Com o conceito de intranet no contexto empresarial, as configurações para cada computador de cada colaborador conseguem ser bastante ágeis:

- a) **Fornecimento do acesso à aplicação *Web*:** Fornecer o endereço URL da aplicação *Web* aos colaboradores e estes adicionam manualmente aos favoritos do Pro/E ou fornecer o documento XML “web_favorites.xml”, que contém todos os endereços das páginas que constituem os favoritos do Pro/E, com o novo URL da aplicação e os colaboradores substituem pelo documento guardado no seu computador.

Antes de enviar o documento XML ao colaborador é necessário editá-lo adicionando o novo URL da aplicação. O documento “web_favorites.xml” está armazenado localmente e para encontrá-lo insere-se o seguinte comando na barra de endereços de uma janela do Windows:

```
%USERPROFILE%\Application Data\PTC\ProENGINEER\Wildfire\wf\.Settings
```

Para adicionar o novo URL insere-se a seguinte linha de código abaixo do código “<bookmarks_nodes_info SOAP-ENC:Array=[X⁸]”><bookmarks_nodes_info_ENTRY label="Personal Favorites" pointer="TRUE" (...):

```
<bookmarks_nodes_info_ENTRY label="Google" pointer="TRUE"  
location="URL_da_pagina_web" depth_in_tree="0" is_file_bookmark="8" />
```

Os colaboradores substituem o antigo documento pelo novo e reiniciam o Pro/E para as alterações surtirem efeito.

- b) **Registo do controlador ActiveX desenvolvido pela PTC:** Se todos os colaboradores iniciam o Pro/E com privilégios de administração para desenvolver os produtos o registo é feito de forma automática pelo Pro/E. Caso isto não aconteça e/ou estejam

⁸ Esta variável representa o número de endereços que constituem os favoritos do Pro/E. Para adicionar um endereço é necessário incrementar esta variável.

instaladas diversas versões do Pro/E no mesmo computador, o administrador do sistema deve registar o controlador ActiveX de forma manual.

- c) **Edição do ficheiro de configuração do Pro/E (config.pro) de forma a ativar a opção “web_enable_javascript”:** Se o Pro/E instalado em cada computador de cada colaborador tiver acesso a um ficheiro “config.pro” disponível na rede, apenas é necessário modificar esse ficheiro. Outra alternativa é distribuir o ficheiro pelos vários colaboradores.

- d) **Configuração das definições de segurança do *browser* Internet Explorer:** As definições de segurança do Internet Explorer podem ser configuradas num único local usando a ferramenta “Group Policy Management Editor” do Windows Server e desta forma as definições requeridas são aplicadas a todos os computadores pertencentes à rede.

Apêndice C – Configuração do Ambiente de Desenvolvimento e *Deployment*

Apêndice D - Módulo JavaScript

Para atender somente aos requisitos de uma determinada aplicação pode ser criado um módulo JavaScript com as funções comuns a todas as aplicações. Este módulo deve permitir a criação de um objeto da API JavaScript no sistema operativo Windows e a obtenção do objeto sessão do Pro/ENGINEER. Se o módulo for executado com sucesso significa que a configuração do ambiente de desenvolvimento está correta e que a aplicação pode ser executada dentro do *browser* interno do Pro/E. Este módulo é uma versão adaptada do ficheiro *pfcutils.js* disponibilizado pela PTC com a vantagem de permitir detetar mais facilmente os erros do processo de configuração do ambiente de desenvolvimento ou *deployment*.

O ficheiro "moduloJavaScript.js" é composto pelas funções **pfcCreateSession()**, **pfcCreate(className)**, **isProEEmbeddedBrowser()**, **pfcIsWindows()**. Este ficheiro contém também quatro variáveis globais: **msg**, para receber as mensagens de erro retornadas por este módulo; **mGlob**, para guardar o objeto da classe **MpfcCOMGlobal** para que as restantes aplicações consigam utilizar os restantes métodos deste objeto; **session**, para guardar o objeto sessão.

3.1 Função pfcCreateSession()

Esta função é responsável pela obtenção do objeto sessão do Pro/ENGINEER.

```
var msg;
var mGlob;
var session;
function pfcCreateSession()
{
    mGlob = pfcCreate("MpfcCOMGlobal");

    if(mGlob)
    {
        try{
            session = mGlob.GetProESession();
            return true;
        }
        catch(err)
        {
            msg = "A opção \"web_enable_javascript\" do ficheiro de
            configuração do Pro/E não está ativa.";
            return false;
        }
    }
}
```

Código 7 – Exemplo de extrato de código

Nesta função é instanciado um objeto da classe **MpfcCOMGlobal**. Se o objeto for instanciado com sucesso então é utilizado o método **GetProESession()** dessa classe. Se o método não lançar uma exceção é retornado o valor “true”, caso contrário é atribuída à variável **msg** uma mensagem de erro. O único erro que pode acontecer nesta situação é pelo facto da opção “**web_enable_javascript**” estar desativa, uma vez que no instante imediatamente anterior foi instanciado um objeto da classe **MpfcCOMGlobal** e não houve problemas com as definições de segurança do IE e a aplicação foi executada no browser interno do Pro/E.

3.2 Função isProEEmbeddedBrowser()

Esta função é responsável por verificar se a aplicação está a ser executada dentro do *browser* interno do Pro/ENGINEER.

```
function isProEEmbeddedBrowser()
{
    if(window.external && window.external.ptc)
        return true;
    else
        return false;
}
```

Código 8 - Extrato de Código

O objeto **window** é um objeto do JavaScript que representa uma janela do *browser*.

3.3 Função pfcIsWindows()

Esta função é responsável por verificar se a aplicação está a ser executada no *browser* do IE.

```
function pfcIsWindows()
{
    if(navigator.appName.indexOf("Microsoft")!=-1)
        return true;
    else
        return false;
}
```

Código 9 - Extrato de Código

O **navigator** é um objeto do JavaScript que contém informações sobre o *browser*. Uma das suas propriedades é a **appName**, que indica o nome do *browser*. Se o nome do *browser* contém a palavra “Microsoft” retorna o valor booleano “true”, senão retorna “false”.

3.4 Função pfcCreate()

Esta função é responsável pela instanciação de uma determinada classe. À exceção das classes relacionadas com o Pro/E, todas as restantes classes podem ser instanciadas utilizando esta função.

```
function pfcCreate(className)
{
    if(isProEEmbeddedBrowser())
    {
        if(pfcIsWindows())
        {
            try
            {
                return new ActiveXObject("pfc."+className);
            }
            catch(err)
            {
                msg = "Erro! Possíveis problemas:\n 1. A opção
                \nInicializar e efetuar o script de controlos ActiveX não
                assinalados como seguros para o processamento de scripts\n
                não está ativa nas definições de segurança do IE;\n 2.A API
                Pro/Web.Link não está registada;\n 3.A classe \""+className+
                "\" não existe";

                return false;
            }
        }
    }
    else
    {
        msg = "Esta aplicação não suporta o browser Mozilla do Pro/E.";
        return false;
    }
}
else
{
    msg = "Esta aplicação só pode ser executada no browser do
Pro/E.";
    return false;
}
}
```

Antes de instanciar a classe é necessário verificar se a aplicação está a ser executada no *browser* interno do Pro/E e em caso negativo armazena-se a mensagem de erro. Em caso afirmativo a aplicação verifica se o *browser* é o IE e se sim instancia a classe, senão armazena uma mensagem de erro na variável **msg**. Neste contexto a aplicação apenas suporta a versão Windows do Pro/ENGINEER. Para instanciar objetos na plataforma UNIX é necessário utilizar outra função que não é aqui referida por não ser objeto de interesse.

Dado que a API Pro/Web.Link é um controlador ActiveX, para instanciar a classe é necessário utilizar um objeto do JavaScript que possui a seguinte sintaxe:

new ActiveXObject(servername.typename[, location])

Este objeto retorna uma referência a um objeto de automatização. Os argumentos a indicar obrigatoriamente são o “**servername**”, o nome da aplicação que fornece o objeto e neste caso é “pfc”, e o “**typename**” que corresponde à classe do objeto a obter

Apêndice E - Caso de Uso *Editar Legenda*

Formato Completo

Actor Principal

Utilizador

Partes interessadas e seus interesses

Utilizador: Diminuição do tempo despendido no processo de alteração dos campos da legenda, que passa a ser feita de forma mais rápida, simples e automática.

Empresa: Aumento da produtividade. Devido ao aumento da eficiência do processo de edição da legenda, os colaboradores podem dedicar mais tempo a outras tarefas mais importantes.

Objetivo

O propósito deste caso de uso é permitir editar os campos da legenda de um desenho.

Pré-Condições

O utilizador tem de ter uma sessão válida na intranet.

Pós-Condições em caso de sucesso

Os campos da legenda são atualizados.

Trigger

O utilizador solicita ao *browser* a execução da aplicação *Web*.

Percurso Principal de Sucesso

1. A aplicação obtém as características dos campos (nome do campo, tipo de dados que aceita, indicação se é um parâmetro ou um campo da legenda, e outros elementos.) a ser editados e constrói o formulário de acordo com essas informações.

2. A aplicação executa o módulo JavaScript “*moduloJavascript.js*” ([Apêndice D](#))

3. O módulo JavaScript é executado com sucesso e a aplicação verifica se o objeto ativo da sessão do Pro/E é do tipo *drawing* e se sim a aplicação adquire a lista de modelos associados e procura pelo primeiro modelo que encontrar que tenha o nome do desenho no seu prefixo.

4. A aplicação verifica se o modelo encontrado no passo anterior é uma instância e em caso afirmativo obtém o genérico desse modelo.

5. A aplicação lê, do objeto genérico, os valores atuais dos parâmetros obtidos no passo 1 e lê os valores dos restantes campos, não associados a parâmetros, a partir da instância do símbolo com o nome “*TEXTO-LEGENDA*” que está situado na página ativa do desenho.

6. A aplicação apresenta ao utilizador os seus valores da legenda obtidos no passo anterior e, nos casos aplicáveis, a gama de valores possíveis que estes podem assumir.

7. A aplicação efetua a validação do Windchill verificando se o Pro/E tem um servidor do Windchill ativo e se sim verifica se o utilizador fez *check out* ao desenho, ao modelo genérico a ele associado e, se for aplicável, às instâncias que estão associadas ao modelo genérico.

8. Caso a validação do Windchill seja efetuada com sucesso no ponto anterior, a aplicação sugere o valor para o parâmetro "NUM_MOD" e determina se o objeto a ser editado é novo. Caso o objeto seja novo, a aplicação sugere valores, que estão definidos por omissão, para alguns dos restantes parâmetros.

9. O utilizador edita os campos que desejar e indica ao sistema para registar os valores modificados.

10. Antes de modificar a legenda, a aplicação executa novamente o passo 7 para a validação do Windchill. Caso a validação seja efetuada com sucesso, a aplicação procede à alteração dos valores da legenda colocando com fundo verde os campos que foram alterados com sucesso e apresentando uma mensagem de sucesso.

11. Para que o utilizador possa ver as atualizações feitas em caso de sucesso, a aplicação regenera o desenho, o modelo genérico que contém os parâmetros e, caso existam, as instâncias associadas.

Fluxos Alternativos

*a Em qualquer altura o utilizador pode cancelar a operação de edição do parâmetro ou reiniciar o formulário, caso ainda não tenha confirmado o registo dos valores, através da opção adequada da interface. Desta forma, as alterações feitas pelo utilizador não vão ter qualquer efeito.

2a. O módulo JavaScript não é executado com sucesso:

1. A aplicação termina apresentando uma mensagem a indicar o erro ocorrido.

3a. Não existe um objeto ativo do tipo "*drawing*" na sessão:

2. A aplicação termina apresentando uma mensagem de aviso a informar que o objeto ativo na sessão não é um *drawing*.

3b. Não existem modelos associados ao desenho ou não existe um modelo associado ao desenho com o mesmo nome do desenho no seu prefixo:

1. A aplicação termina apresentando uma mensagem de aviso a informar que não encontrou o modelo que possui os parâmetros da legenda.

7a. O servidor Windchill está ativo no Pro/E e o utilizador não fez *check out* ao desenho, ao modelo genérico e às instâncias:

1. A aplicação bloqueia o formulário, não permitindo a sua edição, e termina.

10a. Erro na atualização do valor de pelo menos um campo:

1. O sistema coloca com o fundo vermelho os campos que não foram modificados devido a um erro e apresenta uma mensagem a indicar quais os erros que ocorreram durante a modificação.
2. O sistema volta ao ponto 9 e continua a mostrar as modificações que o utilizador efetuou.

Especificações Suplementares

Enquanto a subsecção anterior descreve os passos que constituem o percurso de execução da aplicação ProPEditor, esta subsecção apresenta algumas especificações adicionais associadas a cada um dos passos de forma a contribuir para uma melhor compreensão dos mesmos.

Passo 1: O departamento de SI decidiu que as características dos campos deviam ser fornecidas pelo código da aplicação em vez de estarem armazenadas num ficheiro ou numa tabela numa base de dados.

O formulário deve conter um cabeçalho com dados do objeto: nome do modelo (fornecido pela Pro/Web.Link), a versão do modelo (fornecida pelo parâmetro "PTC_WM_VERSION" ou pela Pro/Web.Link), e o estado do modelo (fornecido pelo parâmetro "PTC_WM_LIFECYCLE_STATE").

Quando os parâmetros são do tipo *integer* ou *real number*, o formulário deve permitir unicamente a introdução de números.

Disponibilizar um botão (imagem) para a tradução de alguns campos. Neste caso é necessário traduzir o campo "DESCRIPTION":

- A aplicação deve selecionar automaticamente o idioma de origem, sendo este igual ao que está ativo na intranet.
- Se o idioma ativo na intranet for o inglês, a aplicação deve selecionar automaticamente o português como idioma destino;
- Se o idioma ativo na intranet for o português ou o espanhol, a aplicação deve selecionar automaticamente o inglês como idioma destino.

Apêndice E - Caso de Uso Editar Legenda

Passo 5: Segundo o departamento de SI, o ProPEditor apenas tem de permitir a leitura e alteração dos parâmetros do modelo genérico uma vez que em 95% dos casos os valores dos parâmetros das instâncias são iguais aos do genérico e ao alterá-los no genérico estes alteram-se automaticamente nas instâncias (parâmetros *Generic-Driven* ou *Inherited Table-Driven* com o valor *). Se no futuro se justificar a colocação de uma opção que permita também editar os valores dos parâmetros das instâncias que se distinguem dos valores dos parâmetros do genérico (parâmetros *Independent Table-Driven* ou *Inherited Table-Driven* com um valor diferente de *), esta pode ser implementada fora do âmbito desta dissertação.

Passo 6: É necessário apresentar uma lista de valores pré-definidos para os parâmetros “PROJECTISTA”, “DESENHADOR”, “MATERIAL”, “PROTECCAO” e “NQ”. Estes valores estão listados numa base de dados.

Passo 7: Se o Pro/E tiver uma ligação ativa ao Windchill e se o utilizador quiser alterar no Pro/ENGINEER os campos do desenho, este tem de fazer *check out* ao desenho no Windchill e, no caso da edição dos campos que estão relacionados com parâmetros, este tem de fazer *check out* ao modelo genérico (que está associado ao desenho e que tem o nome igual a este) e às suas instâncias. Para simplificação da aplicação ProPEditor, se um servidor do Windchill estiver ativo o utilizador tem de fazer *check out* ao desenho, ao modelo genérico e às suas instâncias para conseguir editar um campo da legenda, independentemente se este é parâmetro ou não. Esta verificação serve como reforço pois ao gravar o desenho no Windchill o Pro/E vai solicitar ao utilizador para fazer *check out* aos modelos necessários. Se o servidor do Windchill não estiver ativo no Pro/E devido a problemas do servidor ou porque a empresa deixou de trabalhar com o Windchill ou se os modelos não estiverem guardados no Windchill, o ProPEditor permite que o utilizador edite a legenda do desenho e se este for gravado mais tarde no Windchill o Pro/E certifica que o utilizador faz *check out* aos modelos. Se o ProPEditor não detetar os modelos no Windchill este também deve permitir que o utilizador edite a legenda do desenho.

Passo 8: O modelo genérico associado ao desenho pode ser novo ou uma revisão. A aplicação verifica se o modelo é novo a partir da verificação da versão que é fornecida pelo seu parâmetro “PTC_WM_VERSION”. Se este parâmetro não existir ou possuir os valores “-.1” ou “.0” significa que o objeto é novo.

Quando o modelo genérico é novo a aplicação deve sugerir valores para os seguintes parâmetros:

- PROJECTISTA e DESENHADOR: Parâmetro USER_CAD_NAME definido na aplicação PROFILER e associado ao utilizador da sessão.
- APROVADOR: Parâmetro DEFAULT_CAD_APPROVER definido na aplicação PROFILER e associado ao utilizador da sessão.
- DATA_APROV: 10 dias úteis após a data atual;
- DATA_MOD: data atual.

Estes parâmetros devem ser sugeridos quando o modelo é novo uma vez que os seus valores devem manter-se para sempre e mesmo no caso de uma revisão estes não são alterados.

A aplicação deve sugerir sempre um valor para o parâmetro “NUM_MOD” pois o utilizador deve editá-lo sempre uma vez que este varia de acordo com a revisão do modelo:

- “NUM_MOD”: Apresenta a change Notice associado ao objeto no Visualizer ou, no caso de não existir, uma lista das change notices ativas associadas ao utilizador. A partir desta lista o utilizador pode selecionar a *change notice* que quer. O ProPEditor deve fazer o registo da change notice no respetivo parâmetro do Pro/E e os utilizadores têm que ir ao Visualizer associar esta change notice ao objeto. A integração de algumas funções do Visualizer, nomeadamente a criação de change notices e respetiva associação com os modelos, com o ProPEditor pode ser uma melhoria a implementar numa fase posterior e fora do âmbito desta dissertação porque foi detetada numa fase tardia e, não envolvendo a API Pro/Web.Link, não foi considerada relevante.

Passo 10: Os parâmetros devem ter um visto na opção “designate” para que o Windchill possa reconhecê-los no caso do servidor do Windchill estiver ativo no Pro/E.

Apêndice E - Caso de Uso Editar Legenda

Apêndice F - Caso de Uso *Adicionar Artigo*

Formato Completo

Actor Principal

Utilizador

Partes interessadas e seus interesses

Utilizador: Diminuição do tempo despendido no processo de seleção dos artigos de fixação, que passa a ser feita de forma mais rápida, simples e automática.

Empresa: Aumento da produtividade. Devido ao aumento da eficiência do processo de procura e montagem de um artigo, os colaboradores podem dedicar mais tempo à modelação dos objetos.

Objetivo

O propósito deste caso de uso é permitir a seleção de um artigo a partir de um catálogo *Web* e a colocação desse artigo no modelo *assembly* do Pro/ENGINEER.

Pré-Condições

O utilizador tem de ter uma sessão válida na intranet.

Pós-Condições em caso de sucesso

O artigo selecionado é importado para o modelo *assembly* do Pro/E.

Trigger

O utilizador solicita ao *browser* a execução da aplicação *Web*.

Percurso Principal de Sucesso

1. A aplicação obtém as classes de artigo que devem ser apresentadas (por exemplo, classes de parafusos e anilhas) e apresenta-as ao utilizador. A aplicação também apresenta uma funcionalidade de pesquisa geral.
2. O utilizador seleciona uma classe de artigo.
3. A aplicação agrupa os artigos, da classe previamente selecionada, por uma determinada característica e o utilizador escolhe um grupo.
4. A aplicação apresenta as características do grupo escolhido anteriormente e apresenta uma listagem dos artigos e outros dados, permitindo que o utilizador filtre os artigos por determinados valores das características.
5. O utilizador seleciona um modelo associado ao artigo desejado e pede ao Pro/E que importe para o *assembly* ativo o modelo associado ao artigo.
6. A aplicação executa o módulo JavaScript “*moduloJavascript.js*” ([Apêndice D](#)).

7. A aplicação verifica se o modelo ativo é do tipo “*assembly*”.

8. A aplicação efetua a validação do Windchill verificando se o Pro/E tem um servidor do Windchill ativo e se sim verifica se o utilizador fez *check out* ao *assembly*.

9. Caso a validação do Windchill seja efetuada com sucesso no ponto anterior, a aplicação procura pelo modelo (associado ao artigo selecionado pelo utilizador no passo 5) na sessão do Pro/E e, caso exista um servidor Windchill ativo no Pro/E, no *workspace* e no *commonspace* e importa-o para o *assembly* ativo no Pro/E, apresentando uma janela ao utilizador para que este defina as restrições para o novo modelo adicionado.

Fluxos Alternativos

2a. O utilizador insere um valor no campo de pesquisa global:

1. A aplicação verifica quais os artigos que contêm o valor inserido pelo utilizador no seu código ou na descrição ou no valor de uma determinada característica e lista esses artigos.
2. A aplicação segue para o passo 5.

5a. Todos os modelos associados ao artigo estão obsoletos:

3. A aplicação não permite que o utilizador selecione um modelo.

7a. Não existe qualquer objeto ativo do tipo “*assembly*” na sessão:

1. A aplicação informa que não existem objetos do tipo “*assembly*” ativos na sessão, não possibilitando importar o artigo selecionado.

8a. O servidor Windchill está ativo no Pro/E e o utilizador não fez *check out* ao *assembly* ativo do Pro/E:

1. A aplicação avisa o utilizador que tem de fazer *check out* ao *assembly*, não importando o artigo pretendido, e volta ao ponto 5 para permitir que o utilizador selecione novamente o artigo que deseja importar para o *assembly*.

9a. A aplicação não encontra o modelo associado ao artigo pretendido:

1. A aplicação informa que o artigo pretendido não pode ser importado uma vez que não o encontra e volta ao ponto 5, permitindo que o utilizador selecione novamente o artigo.

Especificações Suplementares

Enquanto a subsecção anterior descreve os passos que constituem o percurso de execução da aplicação ProCatalog, esta subsecção apresenta algumas especificações adicionais associadas a cada um dos passos de forma a contribuir para uma melhor compreensão dos mesmos.

Passo 1: Na pesquisa geral os utilizadores podem pesquisar artigos de todas as classes presentes no GlobalArt.

Se o utilizador quiser visualizar o catálogo, a aplicação pode ser acedida a partir de qualquer navegador, mas para importar um modelo para o *assembly* ativo do Pro/E é necessário executar a aplicação a partir do navegador do Pro/ENGINEER para utilizar funções da API Pro/Web.Link.

As classes de artigos que devem ser apresentadas no catálogo são:

- 120112 – Freios (Freios e Circlips);
- 120113 – Rodas, Rol., Veio, Cavilhas (Cavilhas e Golpilhas);
- 121102 – Anilhas e Placas Oblíquas (Anilhas);
- 121110 – Parafusos;
- 121112 – Porcas;
- 121114 – Rebites.

Passo 3: Após a apresentação e seleção de uma classe de artigo, a aplicação deve agrupar todos os artigos pela característica “norma”. Neste caso, se a classe das anilhas for selecionada devem ser apresentados os vários valores das normas que estas podem assumir. A aplicação deve estar preparada para que diferentes artigos possam ser agrupados por características diferentes.

Os artigos têm de estar bem classificados no GlobalArt e a aplicação deve ser programada levando isso em consideração. Se por alguma razão a classificação dos artigos não estiver correta essa situação deve ser sempre corrigida no GlobalArt.

Passo 4: Pretende-se que o ProCatalog apresente os artigos de todas as companhias. A listagem de artigos não pode conter artigos repetidos e se um mesmo artigo estiver na companhia ativa da sessão da intranet e também noutra(s) companhia(s) é apresentado o artigo que está associado à companhia ativa.

Apêndice F – Caso de Uso Adicionar Artigo

Deve ser colocado um botão de informações de cariz técnico para cada valor da característica responsável pelo agrupamento dos artigos. Neste caso as informações podem ser acerca da característica “norma” (norma din 147, etc.).

Incluir um botão para a criação de um artigo no GlobalArt. Como esta funcionalidade está implementada no GlobalArt esse botão é remetido para a respetiva página do GlobalArt.

Quando o utilizador colocar o ponteiro do rato sobre o artigo, o ProCatalog deve apresentar a imagem do artigo. Esta funcionalidade pode ser copiada do GlobalArt.

Criação de uma hiperligação em cada artigo para que o utilizador tenha acesso à informação colocada na aplicação WebBaan sobre esse artigo.

O utilizador deve saber se o artigo está obsoleto ou não e se estiver apresenta-se um aviso.

Passo 5: Um artigo pode ter vários modelos CAD associados. Todos estes modelos associados devem ser apresentados ao utilizador. A aplicação permite adicionar um desses modelos ao *assembly* ativo do Pro/E se o modelo pretendido não estiver obsoleto. Um artigo também pode estar obsoleto e nesse caso deve ser apresentado um símbolo de aviso.

Passo 8: A validação do Windchill não é feita com sucesso no seguinte caso: um servidor do Windchill está ativo no Pro/E e o *assembly* não está *checked out*.

Apêndice G - ProPEditor: Classes Pro/Web.Link

Este apêndice apresenta as classes e respectivos métodos e propriedades utilizados na aplicação ProPEditor. As descrições dos métodos e propriedades são feitas com base na API *Wizard* da Pro/Web.Link.

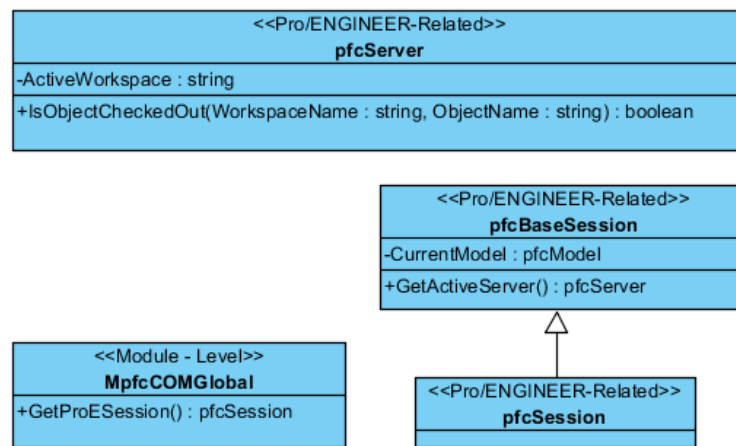


Figura 85 - ProPEditor: Diagrama de classes Pro/Web.Link 1

Classe MpfcCOMGlobal

- **Método GetProESession()** -> **pfcSession**: retorna a sessão do Pro/ENGINEER.

Classe pfcBaseSession

- ❖ **Propriedade CurrentModel** -> **pfcModel**: retorna o modelo ativo da sessão do Pro/ENGINEER. Apesar da documentação da API Pro/Web.Link indicar que o objeto retornado é da classe **pfcModel** na verdade o objeto pertence à classe **pfcPart** ou **pfcAssembly** ou **pfcDrawing** ou a outra classe que represente outro tipo de modelo (**pfcDiagram**, etc.).
- **Método GetActiveServer()** -> **pfcServer**: verifica qual o servidor que está ativo na opção “Server Registry” do menu “Tools” do Pro/ENGINEER e retorna-o sob a forma de objeto da classe **pfcServer**.

Classe pfcServer

- ❖ **Propriedade ActiveWorkspace -> string:** verifica qual o *workspace* ativo que está associado ao servidor ativo na opção “Server Registry” do menu “Tools” do Pro/ENGINEER e retorna o seu nome.
- **Método IsObjectCheckedOut-> boolean:** verifica se um determinado modelo está *checked out* num determinado *workspace* e se sim retorna o valor “true”, se não retorna o valor “false”. Caso seja retornado o valor “undefined” significa que o modelo não está no *workspace*. Os argumentos **WorkspaceName** e **ObjectName** correspondem ao nome do workspace e ao nome do modelo respectivamente.

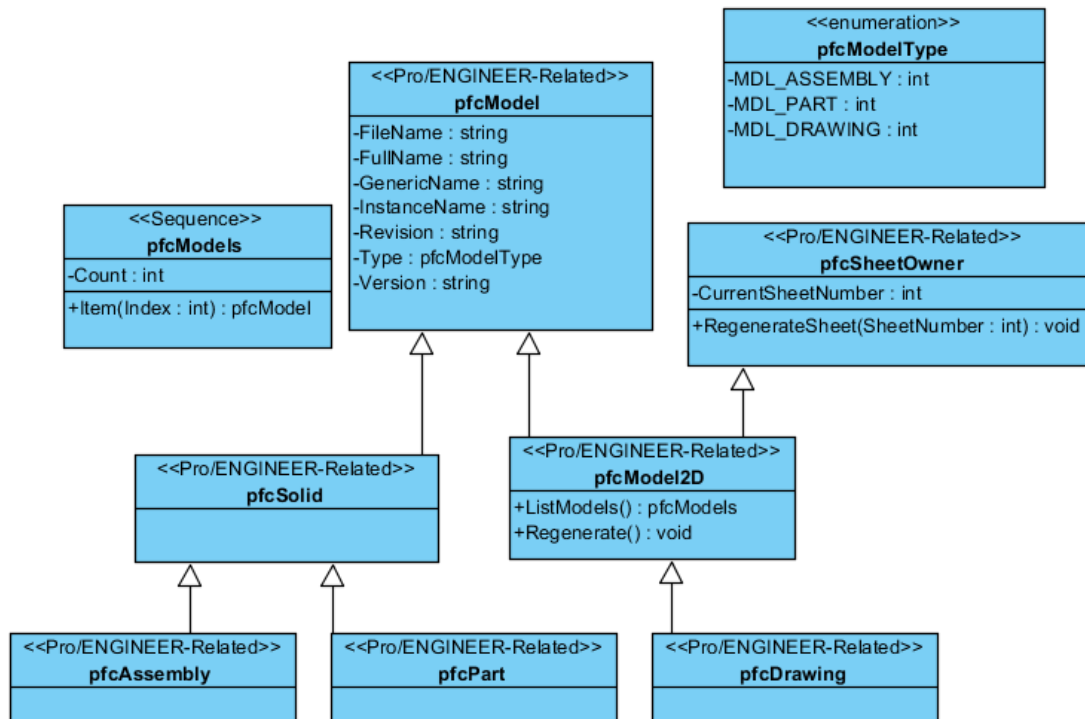


Figura 86 - ProPEditor: Diagrama de classes Pro/Web.Link 2

Classe pfcModel2D

- **Método ListModels() -> pfcModels:** retorna um array com os modelos que compõem um desenho.
- **Método Regenerate() -> void:** regenera modelos do tipo *drawing*, *report* ou *layout*.

Classe pfcModels

- ❖ **Propriedade Count** -> **integer**: retorna o número de modelos que constituem o array.
- **Método Item(integer Index)** -> **pfcModel**: retorna o modelo que está armazenado num determinado índice do array. O modelo retornado pode ser da classe **pfcPart** ou **pfcAssembly** ou de outra classe que represente outro tipo de modelo. O argumento **Index** corresponde ao índice do array que se pretende aceder.

Classe pfcModel

- ❖ **Propriedade FileName** -> **string**: retorna o nome do modelo com a indicação da extensão (“*.prt*”, “*.asm*”, etc.).
- ❖ **Propriedade FullName** -> **string**: retorna o nome do modelo no formato “instância<genérico>” ou, se não houver instância, “genérico”.
- ❖ **Propriedade GenericName** -> **string**: se o modelo for uma instância retorna o nome do modelo genérico sem a indicação da extensão; se o modelo for um genérico então retorna o valor null ou uma string vazia.
- ❖ **Propriedade InstanceName** -> **string**: retorna o nome do modelo sem a indicação da extensão.
- ❖ **Propriedade Revision** -> **string**: retorna a revisão do modelo.
- ❖ **Propriedade Type** -> **pfcModelType**: retorna o número inteiro que representa o tipo do modelo, isto é, se o modelo for do tipo *assembly* retorna o número 0 que corresponde ao elemento “MDL_ASSEMBLY” da classe *pfcModelType*.
- ❖ **Propriedade Version** -> **string**: retorna a versão do modelo.

Classe pfcSheetOwner

- ❖ **Propriedade CurrentSheetNumber**-> **integer**: retorna o número da página ativa de um modelo do tipo *drawing*, *Report* ou *Layout*.
- **Método RegenerateSheet(integer SheetNumber)** -> **void**: regenera uma determinada página de um modelo do tipo *drawing*, *Report* ou *Layout*. O argumento **SheetNumber** corresponde ao número da página que se pretende regenerar.

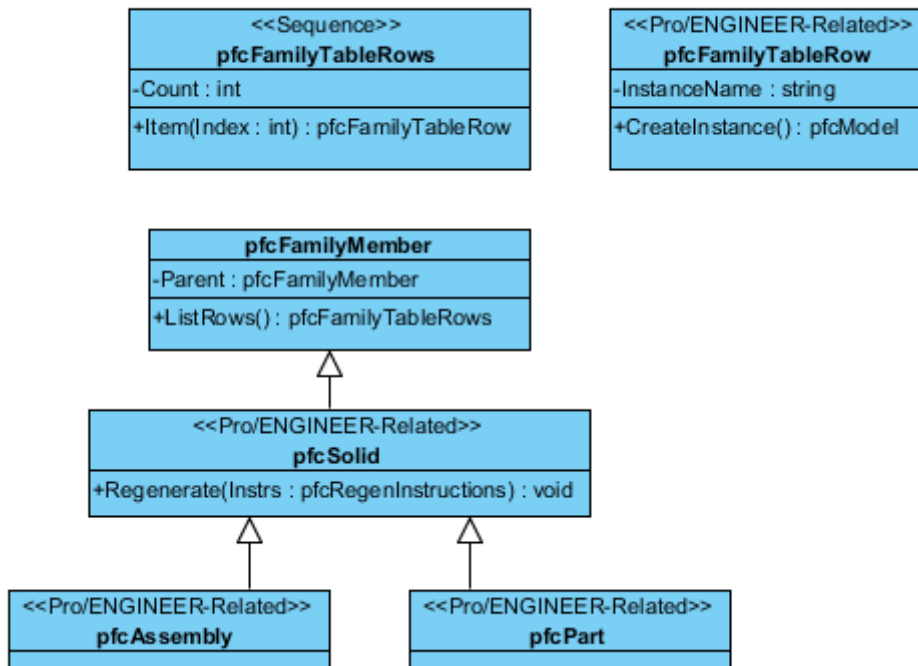


Figura 87 - ProPEditor: Diagrama de classes Pro/Web.Link 3

Classe pfcSolid

- **Método Regenerate(pfcRegenInstructions Instrs) -> void:** regenera um modelo da classe **pfcPart** ou **pfcAssembly**. Visto que na aplicação ProPEditor não é necessário fornecer instruções adicionais ao método e que o argumento Instrs é opcional então o valor utilizado neste argumento é null. A classe **pfcRegenInstructions** é detalhada na aplicação ProCatalog.

Classe pfcFamilyMember

- ❖ **Propriedade Parent -> pfcFamilyMember:** retorna o modelo genérico de um modelo. Se o modelo for uma instância retorna o modelo genérico da classe **pfcPart** ou **pfcAssembly**. Se o modelo for um genérico retorna o valor null.
- **Método ListRows() -> pfcFamilyTableRows:** retorna um array com as linhas que fazem parte da *family table*. A cada uma destas linhas corresponde uma instância.

Classe **pfcFamilyTableRows**

- ❖ **Propriedade Count** -> **integer**: retorna o número de linhas da *family table* que constituem o array.
- **Método Item(integer Index)** -> **pfcFamilyTableRow**: retorna um objeto da classe **pfcFamilyTableRow** que representa uma linha da *family table* que está armazenada num determinado índice do array. O argumento **Index** corresponde ao índice do array que se pretende aceder.

Classe **pfcFamilyTableRow**

- ❖ **Propriedade InstanceName** -> **string**: retorna o nome da instância que está situada numa determinada linha da *family table*.
- **Método CreateInstance()** -> **pfcModel**: regenera uma instância da family table, adiciona-a à sessão do Pro/ENGINEER e retorna-a sob a forma de objeto da classe **pfcPart** ou **pfcAssembly**.

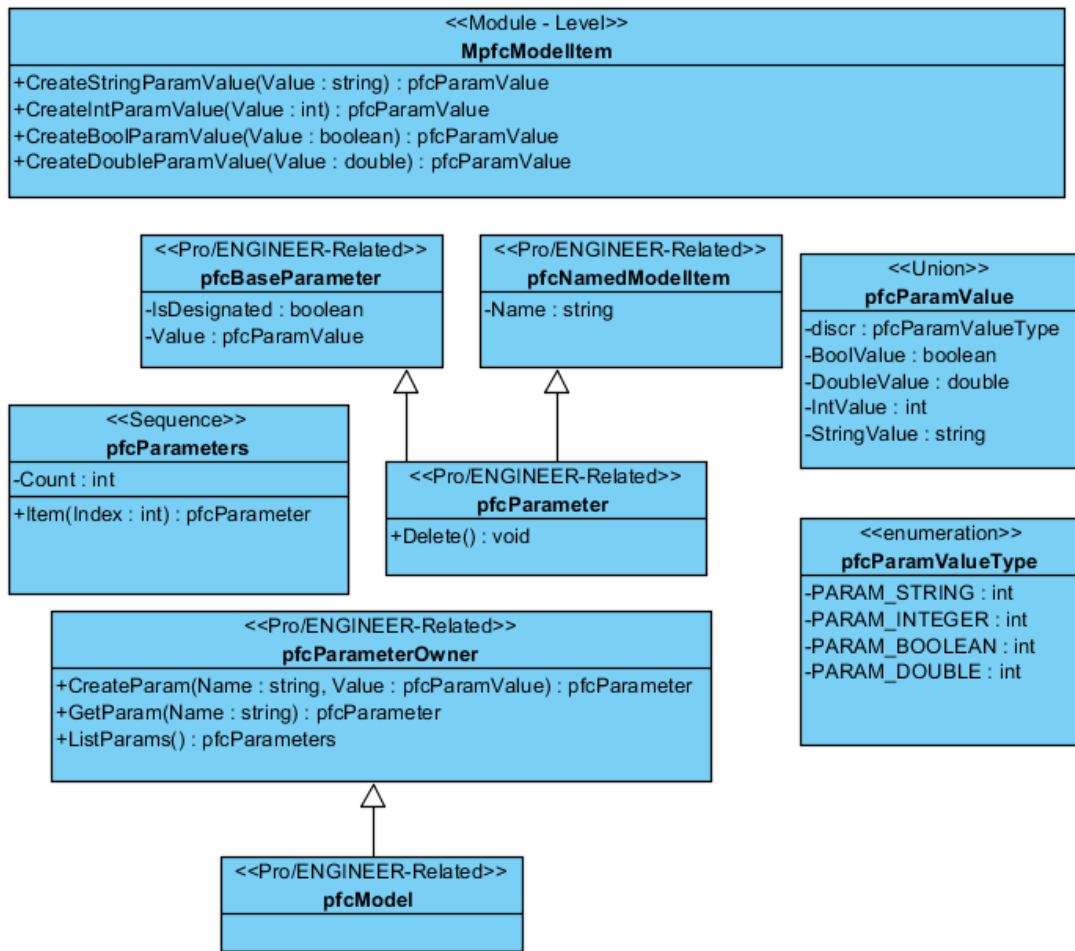


Figura 88 - ProPEditor: Diagrama de classes Pro/Web.Link 4

Classe pfcParameterOwner

- **Método CreateParam(string Name, pfcParamValue Value) -> pfcParameter:** cria um novo parâmetro e retorna-o sob a forma de objeto da classe **pfcParameter**. Para editar um parâmetro é necessário eliminá-lo e criá-lo através deste método. O argumento **Name** corresponde ao nome e o argumento **Value** corresponde ao valor que se pretende atribuir ao parâmetro. Este valor, da classe **pfcParamValue**, pode ser o valor obtido de outro parâmetro ou pode ser criado pelo programador através de um dos métodos da classe **MpcfModelItem**.
- **Método GetParam(string Name) -> pfcParameter:** retorna um determinado parâmetro de um modelo. O argumento **Name** corresponde ao nome do parâmetro que se pretende obter.

- **Método ListParams()** -> **pfcParameters**: retorna um array com os parâmetros de um modelo.

Classe pfcParameters

- ❖ **Propriedade Count** -> **integer**: retorna o número de parâmetros que constituem o array.
- **Método Item(integer Index)** -> **pfcParameter**: retorna um objeto da classe **pfcParameter** que representa um parâmetro que está armazenado num determinado índice do array. O argumento **Index** corresponde ao índice do array que se pretende aceder.

Classe pfcParameter

- **Método Delete()** -> **void**: Elimina um parâmetro do modelo.

Classe pfcBaseParameter

- ❖ **Propriedade IsDesignated** -> **boolean**: verifica se um parâmetro tem a opção “designate” selecionada e se sim retorna o valor “true”, senão retorna o valor “false”.
- **Propriedade Value** -> **pfcParamValue**: retorna um objeto da classe **pfcParamValue** que descreve o valor de um parâmetro. Para ler o valor do parâmetro é necessário utilizar a propriedade adequada da classe **pfcParamValue**.

Classe pfcNamedModelItem

- ❖ **Propriedade Name** -> **string**: retorna o nome de um parâmetro.

Classe pfcParamValue

- ❖ **Propriedade discr** -> **pfcParamValueType**: retorna o número inteiro que representa o tipo do parâmetro, ou seja, se o parâmetro for do tipo string retorna o número 0 que corresponde ao elemento “PARAM_STRING” da classe **pfcParamValueType**.
- ❖ **Propriedade BoolValue** -> **boolean**: retorna o valor de um parâmetro do tipo booleano. Esta propriedade deve ser utilizada exclusivamente em parâmetros do tipo booleano pois de outro modo é retornada uma exceção. O mesmo acontece com as restantes propriedades, onde por exemplo a propriedade **DoubleValue** apenas pode ser utilizada em parâmetros do tipo double.

- ❖ **Propriedade DoubleValue -> double:** retorna o valor de um parâmetro do tipo double.
- ❖ **Propriedade IntValue -> integer:** retorna o valor de um parâmetro do tipo inteiro.
- ❖ **Propriedade StringValue -> string:** retorna o valor de um parâmetro do tipo string.

Classe MpfcModelItem

- **Método CreateStringParamValue(string Value) -> pfcParamValue:** cria um valor para um parâmetro do tipo string. Os métodos desta classe têm como propósito criar um objeto da classe **pfcParamValue** para que posteriormente se possa utilizá-lo no método **CreateParam(string Name, pfcParamValue Value)** da classe **pfcParameterOwner**.
- **Método CreateIntParamValue(integer Value) -> pfcParamValue:** cria um valor para um parâmetro do tipo inteiro.
- **Método CreateBoolParamValue(integer Value) -> pfcParamValue:** cria um valor para um parâmetro do tipo booleano.
- **Método CreateDoubleParamValue(double Value) -> pfcParamValue:** cria um valor para um parâmetro do tipo double.

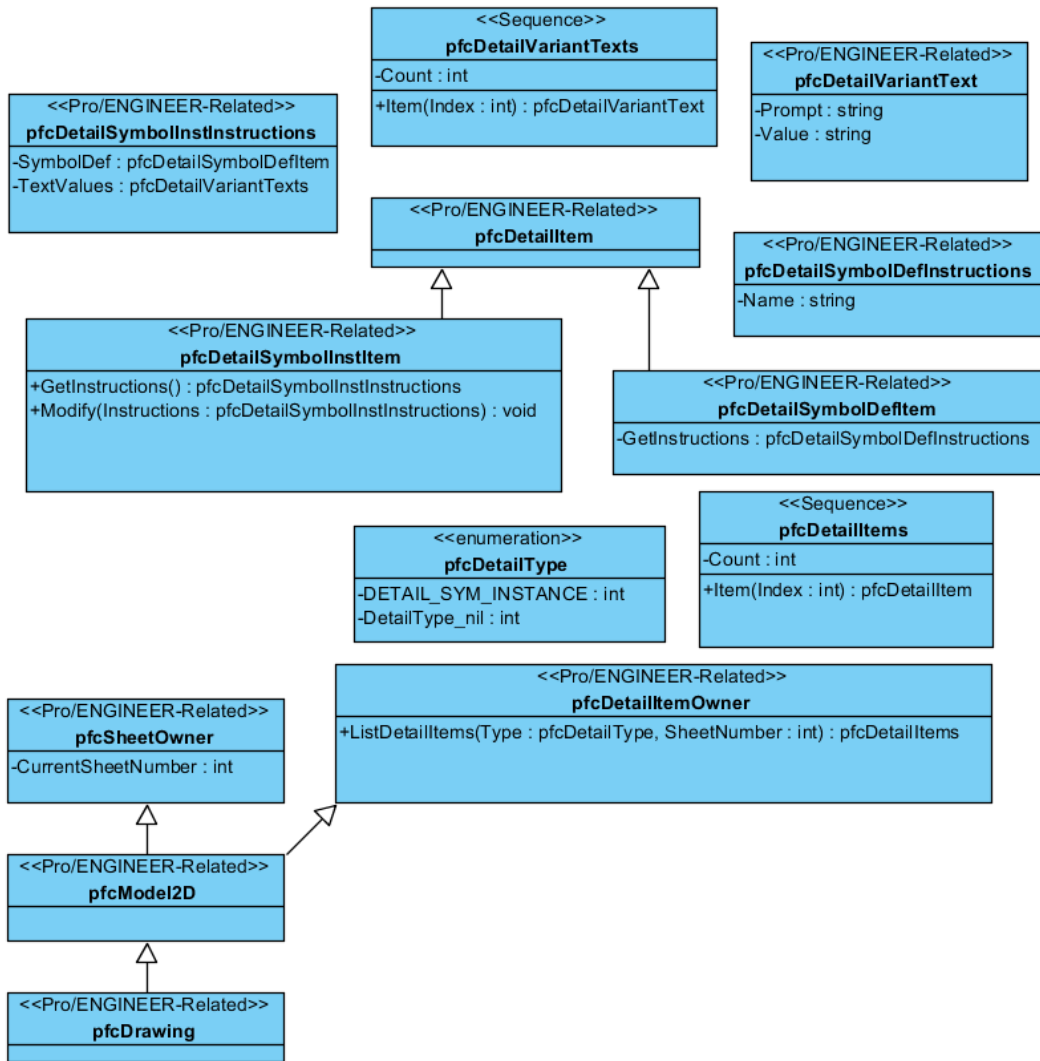


Figura 89 - ProPEditor: Diagrama de classes Pro/Web.Link 5

Classe pfcDetailItemOwner

- Método ListDetailItems(pfcDetailType Type, integer SheetNumber) -> pfcDetailItems:** retorna um array com os itens de detalhe de um modelo do tipo *drawing*, *Report* ou *Layout*. O argumento **Type** representa o tipo de item de detalhe que se pretende obter e o argumento **SheetNumber** representa o número da página do modelo. Como estes argumentos são opcionais então estes podem ter valor null. Para que o argumento **Type** possa representar o valor null é necessário utilizar o elemento **DetailType_nil** da classe **pfcDetailType**. No caso da aplicação ProPEditor pretende-se obter todas as instâncias de símbolos (elemento **DETAIL_SYM_INSTANCE** da classe **pfcDetailType**) da página ativa do desenho. Como foi referido, para obter o

número da página ativa do desenho utiliza-se a propriedade **CurrentSheetNumber** da classe **pfcSheetOwner**.

Classe **pfcDetailItems**

- ❖ **Propriedade Count** -> **integer**: retorna o número de itens de detalhe que constituem o array.
- **Método Item(integer Index)** -> **pfcDetailItem**: retorna um item de detalhe que está armazenado num determinado índice do array. O item retornado pode ser da classe **pfcDetailSymbolInstItem** ou de outra classe que represente outro tipo de item de detalhe de um modelo. O argumento **Index** corresponde ao índice do array que se pretende aceder.

Classe **pfcDetailSymbolInstItem**

- **Método GetInstructions()** -> **pfcDetailSymbolInstInstructions**: retorna um objeto da classe **pfcDetailSymbolInstInstructions** que guarda informação sobre a instância como o nome e valor dos campos da legenda, entre outros.
- **Método Modify(pfcDetailSymbolInstInstructions Instructions)** -> **void**: modifica a informação de uma instância. O argumento **Instructions** representa as novas instruções. Na aplicação ProPEditor estas instruções são uma modificação de instruções.

Classe **pfcDetailSymbolInstInstructions**

- ❖ **Propriedade SymbolDef** -> **pfcDetailSymbolDefItem**: retorna o símbolo que está associado à instância.
- ❖ **Propriedade TextValues** -> **pfcDetailVariantTexts**: retorna um array com os campos editáveis da instância (“Variable Text”).

Classe **pfcDetailSymbolDefItem**

- **Método GetInstructions()** -> **pfcDetailSymbolDefInstructions**: retorna um objeto da classe **pfcDetailSymbolDefInstructions** que contém informação sobre o símbolo, nomeadamente o nome do símbolo, entre outros.
-

Classe pfcDetailSymbolDefInstructions

- ❖ **Propriedade Name** -> **string**: retorna o nome do símbolo.

Classe pfcDetailVariantTexts

- ❖ **Propriedade Count** -> **integer**: retorna o número de campos editáveis da instância.
- **Método Item(integer Index)** -> **pfcDetailVariantText**: retorna um campo editável da instância que está armazenado num determinado índice do array. O argumento **Index** corresponde ao índice do array que se pretende aceder.

Classe pfcDetailVariantText

- ❖ **Propriedade Prompt** -> **string**: retorna o nome do campo editável da instância.
- ❖ **Propriedade Value** -> **string**: retorna o valor do campo editável da instância.

Apêndice H - ProCatalog: Classes Pro/Web.Link

Esta secção apresenta as classes e respetivos métodos e propriedades utilizados na aplicação ProCatalog. As descrições são feitas com base na API *Wizard* da Pro/Web.Link. Para evitar repetições, os métodos e propriedades descritos na aplicação ProPEditor não são aqui referidos. Quando um determinado objeto ou argumento é apresentado com cor laranja significa que a sua classe não está a ser identificada de forma explícita.

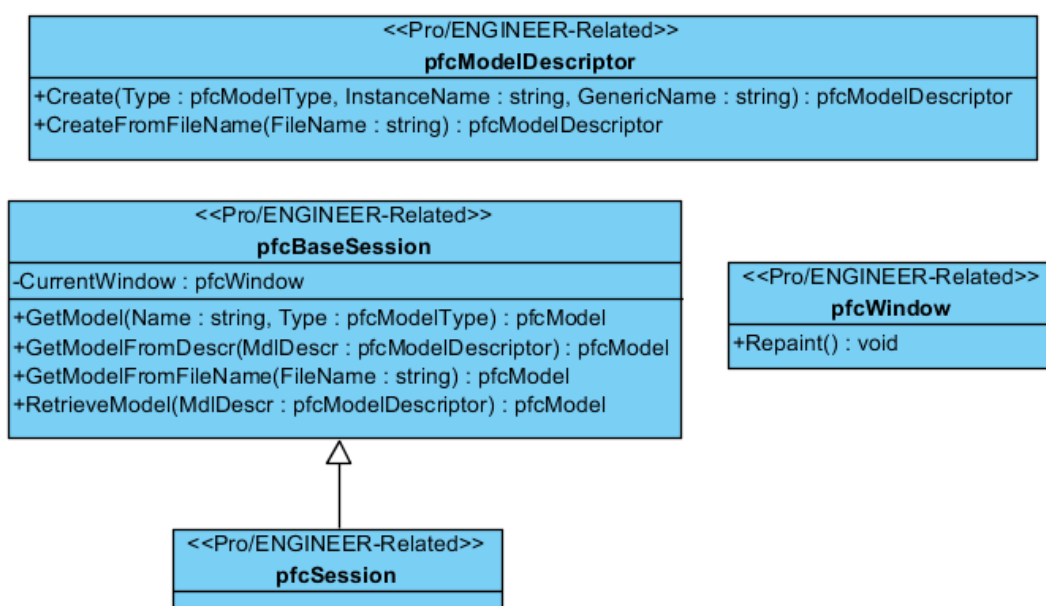


Figura 90 - ProCatalog: Diagrama de classes Pro/Web.Link 1

Classe **pfcBaseSession**

- ❖ **Propriedade CurrentWindow -> pfcWindow:** retorna a janela ativa do Pro/ENGINEER.
- **Método GetModel(string Name, pfcModelType Type) -> pfcModel:** procura por um modelo na sessão do Pro/ENGINEER através da indicação do seu nome sem a indicação da extensão (**Name**) e do seu tipo (**Type**).
- **Método GetModelFromDescr(pfcModelDescriptor MdlDescr) -> pfcModel:** procura por um modelo na sessão do Pro/ENGINEER que possua as mesmas características do model descriptor definido no argumento **MdlDescr**.

- **Método GetModelFromFileName(string FileName) -> pfcModel:** procura por um modelo na sessão do Pro/ENGINEER através da indicação do seu nome e extensão (**FileName**).
- **Método RetrieveModel(pfcModelDescriptor MdlDescr) -> pfcModel:** retorna um modelo, com as mesmas características definidas no model descriptor (**Mdl**), para a sessão do Pro/ENGINEER, não o apresentando ao utilizador. Para retornar o modelo, o Pro/E procura-o no *Workspace* (local e do lado do servidor do Windchill), no *Commonspace*, no *Working directory* e noutros directórios definidos pelo utilizador. Estes directórios podem ser definidos na opção “*search_path*” do ficheiro de configuração do Pro/E ou num ficheiro “*search.pro*” cujo directório tem de estar definido na opção “*search_path_file*” do ficheiro de configuração.

Classe pfcModelDescriptor

- **Método Create(pfcModelType Type, string InstanceName, string GenericName) -> pfcModelDescriptor:** cria um model descriptor através da identificação do tipo do modelo (**type**) e do seu nome (**InstanceName**). O argumento **GenericName** é opcional e identifica o nome do modelo genérico. Se o modelo for uma instância este argumento é igual a null ou a uma string vazia. Os objectos da classe **pfcModelDescriptor** podem ser usados para definir modelos que não estão atualmente em sessão.
- **Método CreateFromFileName(string FileName) -> pfcModelDescriptor:** cria um model descriptor através da indicação do seu nome e extensão (**FileName**).

Classe pfcWindow

- **Método Repaint() -> void:** redesenha uma janela do Pro/ENGINEER.

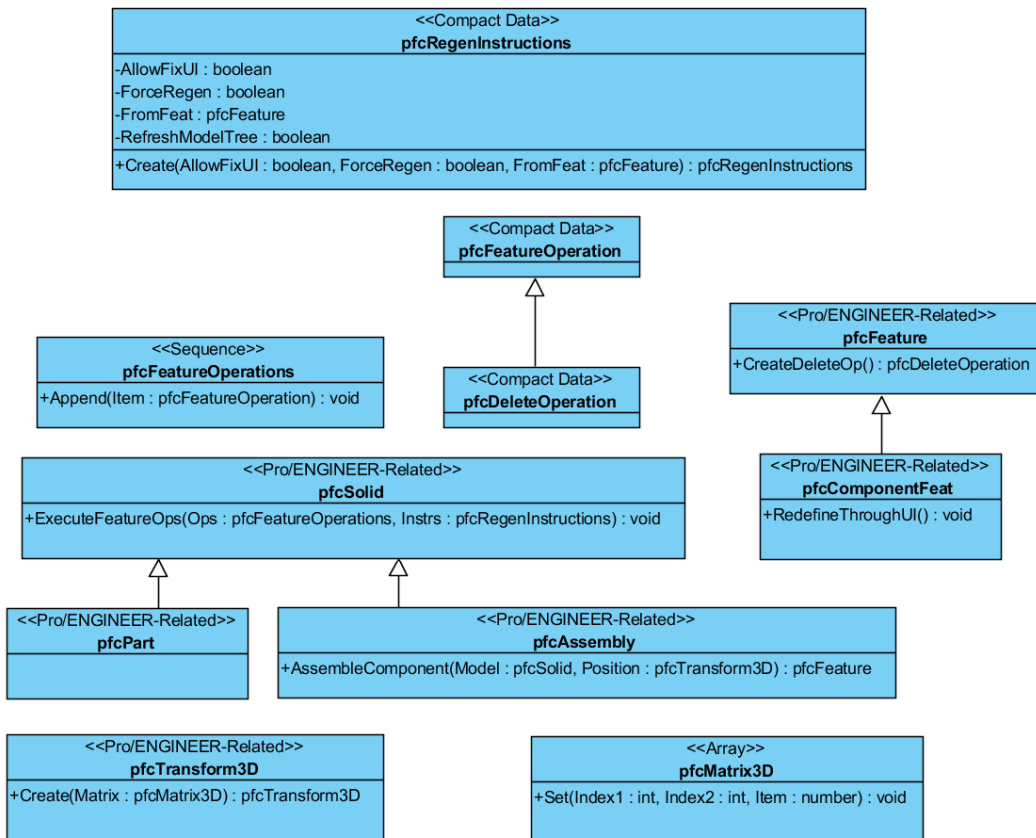


Figura 91 - ProCatalog: Diagrama de classes Pro/Web.Link 2

Classe pfcAssembly

- **Método AssembleComponent(pfcSolid Model, pfcTransform3D Position) -> pfcFeature:** coloca um modelo (**Model**) numa determinada posição do *assembly* (**Position**). O modelo pode ser da classe **pfcAssembly** ou **pfcPart**. No caso da aplicação ProCatalog, após a execução deste método é retornado um objeto da classe **pfcComponentFeat**.


Classe pfcTransform3D

- **Método Create(pfcMatrix3D Matrix) -> pfcTransform3D:** cria um objeto da classe **pfcTransform3D**, que fornece informação sobre o sistema de coordenadas, incluindo informação sobre a matriz de transformação, os eixos e a origem. O argumento **Matrix** representa a matriz de transformação.

Classe pfcMatrix3D

- **Método Set(integer Index1, integer Index2, number Item) -> void:** atribui um valor a um determinado índice do array **pfcMatrix3D**. Este array é bidimensional de 4X4 e armazena uma matriz de transformação tridimensional.

Classe pfcComponentFeat

- **Método RedefineThroughUI() -> void:** invoca a caixa de diálogo do Pro/ENGINEER para que o utilizador possa definir as restrições entre o novo modelo e o *assembly*.
Com este método, se o utilizador selecionar o ícone  para cancelar a colocação do novo modelo no *assembly*, o novo modelo não é eliminado da *model tree* e ocorre uma exceção. Como tal é necessário fazer um *try catch* ao método **RedefineThroughUI()** e invocar dentro do bloco *catch* e sobre o *assembly* o método **ExecuteFeatureOps(pfcFeatureOperations Ops, pfcRegenInstructions Instrs)** para eliminar o novo modelo adicionado e atualizar a *model tree*.

Classe pfcSolid

- **Método ExecuteFeatureOps(pfcFeatureOperations Ops, pfcRegenInstructions Instrs) -> void:** executa uma determinada operação num modelo do tipo *part* ou *assembly*. O argumento **Ops** define a operação e o argumento **Instrs** define algumas instruções.

Classe pfcFeatureOperations

- **Método Append(pfcFeatureOperation Item) -> void:** armazena uma determinada operação (**Item**) no array da classe **pfcFeatureOperation**. O argumento **Item** aceita um objeto de qualquer classe que represente uma operação, como por exemplo, das classes **pfcDeleteOperation** ou **pfcSuppressOperation**.

Classe pfcFeature

- **Método CreateDeleteOp() -> pfcDeleteOperation:** cria uma operação de eliminação de uma *feature*.

Classe pfcRegenInstructions

- ❖ **Propriedade AllowFixUI -> boolean:** quando esta propriedade é colocada a “true”, se houver um erro na execução das instruções de regeneração é apresentada a interface do Pro/E para corrigir um determinado modelo.
- ❖ **Propriedade RefreshModelTree -> boolean:** se esta propriedade for colocada com o valor “true”, a *model tree* é atualizada após a execução das instruções de regeneração.

- **Método Create(boolea AllowFixUI, boolea ForceReg, boolea FromFeat) -> pfcRegenInstructions:** cria um novo objeto da classe **pfcRegenInstructions**. Os argumentos **ForceReg** e **FromFeat** não podem ser usados nesta versão da API Pro/Web.Link, estando reservados pela PTC para uso futuro.

Apêndice I - MassProp: Classes Pro/Web.Link

Esta secção apresenta as classes e respetivos métodos e propriedades utilizados na aplicação MassProp. As descrições são feitas com base na API *Wizard* da Pro/Web.Link. Para evitar repetições, os métodos e propriedades descritos na aplicação ProPEditor e ProCatalog não são aqui citados. Quando um determinado objeto ou argumento é apresentado com cor laranja significa que a sua classe não está a ser identificada de forma explícita.

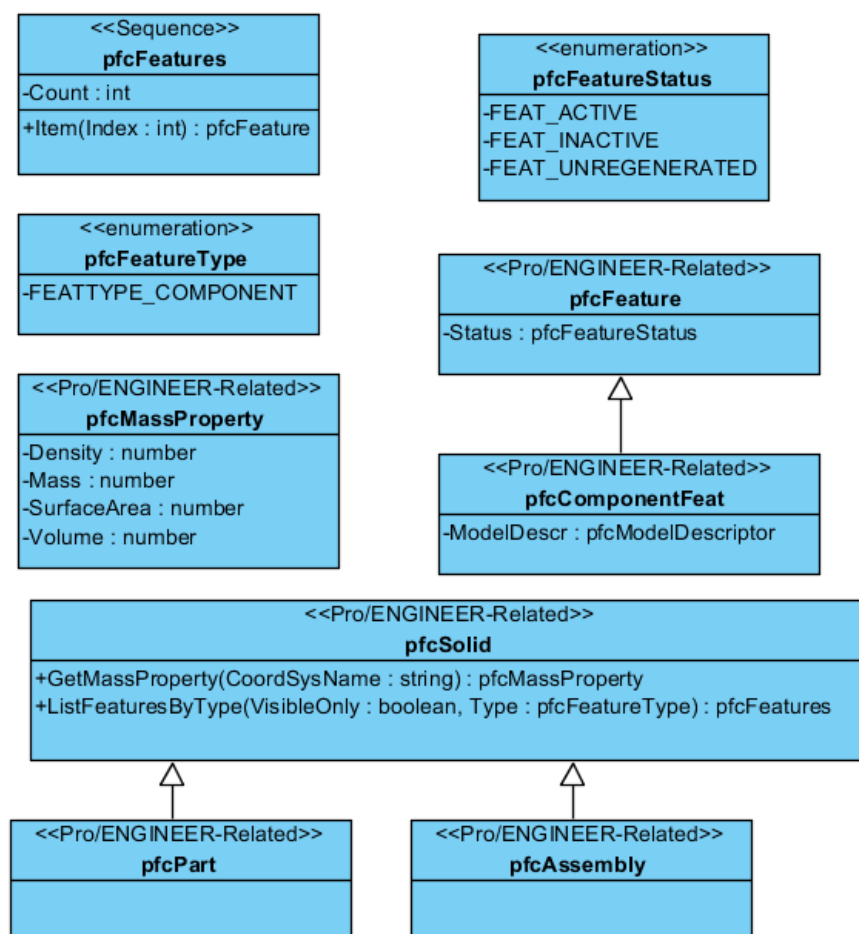


Figura 92 - MassProp: Diagrama de classes Pro/Web.Link 1

Classe pfcSolid

- **Método GetMassProperty (string CoordSysName) -> pfcMassProperty:** obtém um objeto da classe **pfcMassProperty**, que contém as propriedades de massa de um sólido (*part* ou *assembly*). O argumento **CoordSysName** representa o nome do sistema de coordenadas que calcula as propriedades de massa. Se definir o valor null para este argumento, o Pro/E vai utilizar o sistema de coordenadas padrão.
- **Método ListFeaturesByType(boolean VisibleOnly, pfcFeatureType Type) -> pfcFeatures:** lista as *features* que compõem um determinado sólido. O argumento **VisibleOnly** define se a feature está ou não visível e o argumento **Type** define qual o tipo de *feature* a obter. No contexto desta aplicação, pretende-se obter de um modelo do tipo *assembly* todas as *features* do tipo componente (FEATTYPE_COMPONENT).

Classe pfcMassProperty

- ❖ **Propriedade Density-> number:** retorna a densidade de um modelo.
- ❖ **Propriedade Mass-> number:** retorna a massa de um modelo.
- ❖ **Propriedade SurfaceArea-> number:** retorna a área da superfície de um modelo.
- ❖ **Propriedade Volume-> number:** retorna o volume de um modelo.

Classe pfcFeatures

- ❖ **Propriedade Count -> integer:** retorna o número de *features* que compõem um determinado sólido.
- **Método Item(integer Index) -> pfcFeature:** retorna um objeto que representa uma determinada *feature* de um sólido. Nesta aplicação as *features* retornadas são da classe **pfcComponentFeat**, que representam os componentes.

Classe pfcComponentFeat

- ❖ **Propriedade ModelDescr-> pfcModelDescriptor:** retorna o *model descriptor* do componente *part* ou *assembly* da classe **pfcComponentFeat**. As propriedades de massa podem ser usadas unicamente sobre objetos da classe **pfcPart** ou **pfcAssembly**. A única forma de obter estes objetos a partir dos objetos modelo da classe **pfcComponentFeat** é através da utilização do método **GetModelFromDescriptor(pfcModelDescriptor MdIDescr)** da classe **pfcBaseSession**. Isto porque não há uma propriedade da classe **pfcComponentFeat** para obter o nome

- **Método GetPropertyValue(pfcMaterialPropertyType Type) -> pfcMaterialProperty:** retorna uma determinada propriedade do material. Nesta aplicação o objetivo é obter a densidade do material (valor **MTL_PROP_MASS_DENSITY** da classe **pfcMaterialPropertyType**) para depois obter as suas unidades através da propriedade **Units** da classe **pfcMaterialProperty**.

Classe pfcMaterialProperty

- ❖ **Propriedade Units-> string:** retorna a unidade de uma determinada propriedade do material.