

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



**Universidade
do Porto**

**Faculdade de
Engenharia**

FEUP

**Sistema de Gestão Financeira para um Parque Tecnológico
utilizando um Sistema de Geração Automática de Código**

Luís Alexandre Moreira Matias

Relatório do Projecto

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Eng.º João Moreira

**Sistema de Gestão Financeira para um Parque Tecnológico
utilizando um Sistema de Geração Automática de Código**

Luís Alexandre Moreira Matias

Relatório do Projecto

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente:

Arguente:

Vogal:

20 De Março de 2009

Resumo

Este documento descreve o projecto “Sistema de Gestão Financeira para um Parque Tecnológico utilizando um Sistema de Geração Automática de Código”, desenvolvido na Quidgest – Consultoria em Gestão, SA.

De uma forma genérica, o problema apresentado foi o de desenvolver um Sistema de Gestão Financeira inovador e à medida do cliente com recurso a uma ferramenta de geração automática de código: o Génio. No mercado, existem várias aplicações de Gestão de Contabilidade. Contudo, estes produtos são estáticos e por esse motivo não se adaptam a realidades específicas. O Sistema tem de possuir automatismos que dispensem o utilizador de conhecimento funcional de contabilidade, nomeadamente um Classificador Automático de Documentos Contabilísticos. O Sistema deve ainda consolidar a contabilidade de vários organismos num só e possuir uma integração com ferramentas de Gestão Estratégica como o Balanced ScoreCard (BSC) através da construção de um sistema de indicadores financeiros.

Por fim, foi-nos proposto desenvolver uma solução de Controlo de Gestão integrada no Sistema, recorrendo a *Crystal Reports*. Deveria ser investigada a sua implementação com uma de duas tecnologias de *Rich Internet Applications (RIA's)*: *Adobe Flash* e *Microsoft Silverlight*, justificando no final uma orientação para o futuro no que diz respeito à integração de *RIA's* com aplicações de *backoffice* de Genio.

O Sistema de Gestão Financeira foi implementado com vários automatismos que simplificam a sua utilização. A Classificação Automática de Documentos Contabilísticos foi implementada recorrendo a um classificador de documentos que nos permite associar, à priori, os documentos à contabilidade geral e orçamental. É possível integrar o Sistema com o BSC através da solução BSC Quidgest. O sistema de indicadores responde a uma perspectiva meramente financeira, diminuindo assim o âmbito de utilização do BSC. A integração do Flash com o Genio é interessante e os desenvolvimentos actuais devem ser mantidos. Os desenvolvimentos de *RIA's* com Silverlight têm um grande potencial, devendo, por isso, suportar o Controlo de Gestão, bem como futuras soluções de *RIA's* cuja utilização o justifique.

Abstract

This document describes the project “Sistema de Gestão Financeira para um Parque Tecnológico utilizando um Sistema de Geração Automática de Código”, developed in Quidgest – Consultoria em Gestão, SA.

The problem is to develop an innovative software solution to support a Financial Management System. This system has to be developed with an automatic software generation tool and fully customized to a specific organization. There are several solutions to support Financial & Accounting Systems in the market. However, these solutions are static and therefore, they are not flexible to support specific needs. This system needs to have automatic tools to disable the user’s need to have know-how about Financial and Accounting such as an Accounting Documents Classifier. This system still needs to support the consolidation of different accounts of different organizations in one only and to have integration with Strategic Management tools like Balanced ScoreCard with a system of financial indicators.

At last, we have to develop a solution to support Management Control integrated in the System, using Crystal Reports. We have to research the developing of this solution using one of two RIA technologies: Adobe Flash and Microsoft Silverlight and, finally, conclude what is the best way to integrate RIA’s with Genio back office’s applications.

The Financial Management System was developed with several automatic tools that improve user usability. The Accounting Documents Classifier was developed with document classifier that allows you to associate the documents to general and budgeting accounting. It is possible to integrate the System with BSC using the solution BSC Quidgest. The system of indicators only has the financial ones. This fact reduces the BSC full potential. The actual integration of Flash with Genio is interesting and it should be maintained. The development of RIA’s with Silverlight has a large potential in the near future. The Management Control must be supported by Silverlight, like other future solutions of RIA’s, if the it’s usability needs justify it.

Agradecimentos

Quero aqui deixar os meus mais profundos agradecimentos a todos aqueles que possibilitaram o desenvolvimento deste projecto e o finalizar desta etapa.

Começo por deixar um sentido obrigado ao Professor João Mendes Moreira pela perseverança, pela paciência e pela incansável dedicação na orientação deste projecto.

Gostaria de agradecer ao Eng. João Paulo Carvalho e ao Eng. Carlos Nogueira pela porta que me abriram na Quidgest – Consultoria em Gestão, SA. para que assim pudesse aprender o que é estar no mercado de trabalho e assim, crescer profissionalmente. Vou deixar ainda um agradecimento particular a todos os que comigo trabalham e em especial, aos elementos da equipa de desenvolvimento do CNT que tornaram possível este projecto: Dra. Anna Muzalska, Eng. Filipe Garcia, Eng. Marcos Topa e Eng. João Baptista. Só a vossa dedicação extremosa e alto nível de profissionalismo me deixaram chegar aqui. Obrigado também por isso.

Os colegas. Os desconhecidos. Os que me deram a mão. Os que me voltaram a cara. Os que acreditaram em mim. Os que lutaram comigo e contra mim. A todos esses, quero agradecer por me tornarem, durante os últimos cinco anos de Faculdade, no homem em que sou hoje. Sobretudo, quero agradecer aos Amigos que nunca me deixaram cair desamparado nem a sorrir sozinho. Ao longo de tantas aventuras e desventuras, descobri vários nesta Faculdade de Engenharia da Universidade do Porto. A Vocês e à FEUP, trarei sempre no meu coração...por mais longe que pareça que estou, estarei sempre por perto. Este Projecto também é vosso.

Finalmente, quero expressar a minha profunda gratidão e afecto à minha Família. Por serem quem são. Por serem como são. Por me fazerem ser quem sou. Obrigado.

Luís Matias

Conteúdo

Capítulo 1 – Introdução.....	1
1.1 Contexto.....	1
1.2 Descrição do projecto	2
1.3 Objectivos.....	3
1.3.1 Solução Base de Gestão Financeira	4
1.3.2 Integração com BSC Quidgest.....	5
1.3.3 Integração Microsoft Silverlight com CNT	5
1.4 Motivação	6
1.5 Organização do Presente Relatório	7
Capítulo 2 - Revisão Tecnológica	8
2.1 Sistema de Gestão Financeira	9
2.1.1 Tecnologias utilizadas para a implementação da solução	10
2.2 RIA – Rich Internet Application	12
2.3 Génio.....	14
2.3.1 Especificidades Técnicas	14
2.3.2 Conceitos Proprietários	16
2.4 Crystal Reports	19
2.5 Balanced ScoreCard	19
2.6 Adobe Flash.....	20
2.7 Microsoft Silverlight	22
2.7.1 Arquitectura.....	23
2.7.2 Principais Funcionalidades	25
2.7.3 Modelo de Programação	25
2.7.4 Modelo de Utilização	26

2.7.4 Web Services e LINQ.....	27
Capítulo 3 - Descrição da Solução Base	30
3.1 CNT – Contabilidade Financeira Empresarial	30
3.2 Funcionalidades do Sistema de Gestão Financeira.....	32
3.3 Motor de Classificação Automática de Documentos	35
3.4 Conclusões e Trabalho Futuro.....	36
Capítulo 4 - Integração do Sistema de Gestão Financeira Com Balanced Scorecard	38
4.1 Metodologia Balanced ScoreCard	38
4.1.1 Mapas Estratégicos.....	39
4.2 Ferramenta proprietária “BSC Quidgest”	41
4.2.1 Definição dos Mapas Estratégicos	42
4.2.2 Definição de indicadores.....	42
4.2.3 Geração de scorecards.....	42
4.3 Integração CNT com BSC.....	44
4.3.1 Sistema de Indicadores Financeiros.....	44
4.3.2 Sistema de Integração BSC com outras aplicações	46
Capítulo 5 - Integração do Génio com Ria’s	50
5.1 Vantagens das Ria’s no Génio.....	50
5.2 Integração Actual – Flash	51
5.2.1 Flash – Visão Geral.....	53
5.2.2 Integrações existentes	54
5.3 O Problema – Controlo de Gestão	57
5.3.1 Controlo de Gestão – O que é?	57
5.3.2 Controlo de Gestão – Conceitos Básicos.....	58
5.3.3 Controlo de Gestão no Génio – Crystal Reports.....	59
5.3.4 Controlo de Gestão no Génio – Silverlight.....	65

5.4 Ria's com Balanced ScoreCard.....	68
5.5 Solução Desenvolvida – Quid Controller.....	68
5.5.1 Data Binding – Caso Geral.....	69
5.5.2 Quid Controller – Requisitos	73
5.5.3 Principais Dificuldades.....	82
5.6 Flash com Génio vs. Silverlight com Génio.....	83
5.6.1 Métricas para avaliação de RIA's.....	84
5.6.2 Características do Flash	85
5.6.3 Características do Silverlight	88
5.6.4 Silverlight e Flash – Qual utilizar?.....	93
Capítulo 6 - Conclusões e Trabalho Futuro.....	98
Anexos.....	100
Bibliografia e Referências.....	176

LISTA DE FIGURAS

Figura 1 – Tecnologias a utilizar para implementação do sistema de gestão financeira.....	11
Figura 2 – Áreas de Génio mapeadas em sql server.....	15
Figura 3 – Código típico duma rotina manual em génio.	15
Figura 4 – Relação Típica de n para n entre duas áreas em Génio. A seta espelha ligações de n para 1, no sentido da seta. Neste caso, podemos definir que as rubricas (cleco) poderão ter vários orçamentos associados (orcam) e vice-versa.	17
Figura 5 – definição duma listagem em Génio.	19
Figura 6 – Imagem do código fonte da classe cshockwaveflash. Esta classe é responsável pela integração dos objectos .swf.....	21
Figura 7 – Arquitectura do microsoft silverlight. [lmo08]	24
Figura 8 – Definição de controlos silverlight em XAML.	26
Figura 9 – Arquitectura típica duma aplicação em silverlight com ligação a uma sgbd.	29
Figura 10 – Arquitectura do cnt. Podemos verificar que apesar do utilizador efectuar pedidos a duas aplicações diferentes, estas utilizam a mesma base de dados em MS Sql Server. Dentro do cnt, podemos ter tantas aplicações quantas quisermos com as mesmas funcionalidades... 31	31
Figura 11 – Scorecard em árvore gerado a partir do balanced scorecard quidgest.	41
Figura 12 – Exemplo dum scorecard em dashboard. Aqui é Possível verificar o estado actual dos indicadores do mapa estratégico escolhido. o ponteiro azul é a meta, sendo que as diferentes cores no gráfico têm o significado do grau de cumprimento das metas.....	43
Figura 13 – Novo exemplo dum scorecard, em vista de lista.	44
Figura 14 – Arquitectura da integração entre o BSC e outras aplicações.....	47
Figura 15 – Separador do integrador no formulário de consulta/edição num indicador.....	48
Figura 16 – Arquitectura do flash [ado08].....	53
Figura 17 – Código-Fonte HTML típico para integrar uma aplicação Flash numa Página web. [w3s08]	53
Figura 18 – Código fonte duma rotina manual do sistema de gestão de workflows da quidgest.	56
Figura 19 – Mapa de crystal reports em edição.	60
Figura 20 – Modelo relacional base dos mapas.....	62
Figura 21 – Balancete de contas da contabilidade da Câmara Municipal de Aveiro.	64
Figura 22 – Formulário de parametrização do balancete geral de contas no cnt.	64
Figura 23 – Modelo relacional mapeado em classes através do LINQ TO SQL.	71
Figura 24 – Exemplo da interface da webservice wcf.....	72
Figura 25 – Declaração dum controlo datagrid em XAML.....	72
Figura 26 –Interface base da aplicação.....	74
Figura 27 – Edição dum mapa de execução de Orçamento.....	76
Figura 28 – Botões que nos permitem aceder a diferentes visões do mapa.	77
Figura 29 – Interface do Expression Blend.....	80
Figura 30 – Gráfico gerado automaticamente a partir dos dados da orçamentação.	81
Figura 31 – Arquitectura do microsoft message queuing.....	160

Figura 32 – Tabela do Plano Oficial de Contas do CNC (Comissão de Normalização Contabilística). Podemos aqui verificar as contas fundamentais para os Indicadores Financeiros Acima Referidos.	161
---	-----

LISTA DE TABELAS

Tabela 1 – Funcionalidades avançadas do cnt.....	35
Tabela 2 – Requisitos do quidcontroller.....	67
Tabela 3 – Quadro comparativo entre as tecnologias microsoft silverlight e adobe flash	94
Tabela 4 – Quadro das funcionalidades avançadas do sage next e primavera executive.....	168

ABREVIATURAS E SÍMBOLOS

.NET	Plataforma de desenvolvimento e execução de sistemas e aplicações;
ADO.NET	ActiveX Data Objects
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
ASP.NET	Active Server Page da framework .NET
ASPX	Active Server Pages
AVM	ActionScript Virtual Machine
BITMAP	Mapa de Bits
BSC	Balanced ScoreCard
CASE	Computer-Aided Software Engineering
CLR	Common Language Runtime
CNF	Contabilidade e Gestão Financeira
CNT	Contabilidade Financeira Empresarial
CODEC	Codificador/Decodificador
DOM	Document Object Model
EBIT	Earnings Before Interest and Taxes
EBITDA	Earnings Before Interest, Taxes, Depreciation and Amortization
ER	Entidade-Relação
ERP	Enterprise Resource Planning
GGD	Gestão de Workflows
GIF	Graphics Interchange Format
GTS	Gestão de Tesouraria
HTML	HyperText Markup Language
IE	Internet Explorer
IIS	Internet Information Services
IVA	Imposto sobre o Valor Acrescentado
JPEG	Joint Photographic Experts Group
LINQ	Language Integrated Query

LINQtoSQL	Language Integrated Query to Simple Query Language
MFC	Microsoft Foundation Classes
MP3	MPEG-1/2 Audio Layer 3
MSMQ	Microsoft Message Queuing
OEM	Original Equipment Manufacturer
POC	Plano Oficial de Contas
QCT	QuidController
RAD	Rapid Application Development
RIA	Rich Internet Application
ROA	Return On Assets
ROCE	Return On Capital Employed
ROI	Return On Investment
SAFT	Standard Audit File for Tax purposes
SDK	Software Development Kit
SGBD	Sistema de Gestão de Base de Dados
SQL	Simple Query Language
SWF	ShockWave Flash
TTF	True Type Fonte
UI	User Interface
UML	Unified Modeling Language
VB	Visual Basic
WEB	World Wide Web
WCF	Windows Communication Foundation
WMA	Windows Media Audio
WMV	Windows Media Video
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

CAPÍTULO 1 – INTRODUÇÃO

O presente documento tem o propósito de descrever as actividades realizadas e os produtos desenvolvidos no âmbito da disciplina de **Projecto** do Mestrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto (**MIEIC-FEUP**).

De uma forma genérica, o problema apresentado foi o de desenvolver um Sistema de Gestão Financeira para um Parque Tecnológico que se designa por CNT – Contabilidade Financeira Empresarial com uma série de automatismos que permitam que o sistema seja utilizado sem exigirem que o utilizador possua um grande conhecimento funcional de contabilidade. Por outro lado, é necessário desenvolver integrações com Sistemas de Gestão Estratégica e com o Controlo de Gestão.

1.1 CONTEXTO

No mercado, é comum encontrar várias aplicações de Gestão de Contabilidade. Produtos finais, fechados, estáticos, sem grande margem de adaptação a necessidades específicas tais como a inclusão de várias empresas/organismos sobre a alçada de um só e a consolidação das suas contas para efeitos de Controlo de Gestão. Normalmente, as aplicações permitem gerir diferentes organismos mas estes depois não podem ser consolidados num só. Isto prende-se com a dificuldade existente em definir, por exemplo, rubricas orçamentais comuns a todas as despesas.

Ex.: Se tivermos uma rubrica orçamental com a denominação “Electricidade” em aplicativos relacionados com diferentes empresas, teremos uma tendência natural para fazer a sua consolidação somando os orçamentos lançados nessa rubrica para períodos idênticos nas duas empresas. Contudo, podemos ter que numa empresa a rubrica orçamental “Electricidade” sirva para lançar orçamentos relativos a todas as despesas de electricidade e noutra, esta apenas sirva para descrever as despesas com electricidade das instalações que estão sob seu usufruto directo e não das arrendadas. Isto poderá fazer com que o Controlo de Gestão não consiga retirar uma conclusão em relação a variações do orçamentado em relação a esta rubrica: o seu aumento poderá significar um aumento das despesas da electricidade ou um crescimento das receitas através do aumento dos alugueres dos espaços próprios.

Noutra vertente, existem também algumas aplicações que nos permitem efectuar um Controlo de Gestão automático e padrão. Contudo, estas não fogem à apresentação simples de mapas estáticos com base na importação de dados de outro sistema. A sua reduzida margem de mercado prende-se com o facto de serem muitas vezes aplicações demasiado padronizadas numa disciplina que é, frequentemente, feita à medida de acordo com a Estratégia definida com a empresa. A interacção com o utilizador é praticamente nula e o aspecto gráfico é pobre. Tipicamente, estas aplicações centram-se exclusivamente numa vertente financeira da gestão, não existindo preocupação com outras métricas, tal como as organizações cada vez mais o exigem. Poderemos verificar estes factos na análise de mercado que é efectuada na secção M) de Anexos.

Todas as aplicações são normalmente dirigidas a contabilistas ou exigem que os utilizadores tenham conhecimentos aceitáveis de contabilidade, quando hoje em dia temos um grupo de actores financeiros muito mais diversificado e, consequentemente, a necessidade cada vez mais emergente de abrir a contabilidade a recursos humanos com um leque de competências mais vasto.

Em acréscimo a todos os requisitos referidos anteriormente, não existia no mercado um sistema de gestão financeira direccionado a Parques Tecnológicos. Foi o pedido duma solução feita à medida, o mais rapidamente possível, adaptável à realidade das mudanças constantes da legislação e das lógicas de negócio, direccionada a Parques Tecnológicos e que comportasse todos os requisitos definidos anteriormente que a Quidgest utilizou como mote para dar asas ao desafio de desenvolvimento do CNT.

1.2 DESCRIÇÃO DO PROJECTO

O desenvolvimento deste sistema tem como base uma ferramenta proprietária de Geração Automática de Código: o Génio. Este, além de permitir uma grande reutilização de código fonte, privilegia o desenvolvimento iterativo, possibilitando o fácil desenvolvimento de funcionalidades sobre uma solução base. Para uma melhor análise a esta ferramenta, poderemos consultar o Capítulo 2.

Para este sistema, foram definidos uma série de requisitos base do que deveria ser, para nós, as funcionalidades standard dum Sistema de Gestão Financeira. Para desenvolver estas funcionalidades, a Quidgest definiu uma equipa de desenvolvimento. Estes requisitos estão aprofundados de forma detalhada no Capítulo 3.

Contudo, essas funcionalidades base e os recursos do Génio não eram suficientes para dar resposta aos requisitos previamente definidos como desejáveis para que uma solução deste tipo possa ser colocada no mercado. Este Projecto foi criado para responder a estas necessidades. Foi então proposto ao mestrando que:

- Integrasse a equipa de desenvolvimento das funcionalidades base do CNT, para que este pudesse conhecer de forma profunda a aplicação e a sua lógica de negócio;
- Desenvolvesse, com recurso a uma série de ferramentas e tecnologias previamente definidas por opção estratégica da empresa, um conjunto de funcionalidades sobre a aplicação base que a permitisse responder aos requisitos definidos pelo cliente;

1.3 OBJECTIVOS

O conjunto de funcionalidades a desenvolver sobre os requisitos base referidos na secção acima é composto pelos seguintes conjuntos de desenvolvimentos:

- Desenvolvimento dum Motor de Classificação de Documentos Contabilísticos que permita que o utilizador não tenha de possuir conhecimentos contabilísticos para efectuar a introdução de documentos no sistema. Será necessário desenvolver automatismos que permitam, com base numa classificação prévia dos produtos/artigos, que os lançamentos contabilísticos sejam efectuados sem intervenção do utilizador;
- Desenvolvimento dum Sistema de Indicadores Financeiros com vista à integração da ferramenta proprietária “Balanced ScoreCard Quidgest” com o CNT;
- Investigação e Desenvolvimento sobre a tecnologia Microsoft Silverlight com vista à integração com o Controlo de Gestão do CNT e com a ferramenta proprietária

“Balanced ScoreCard” Quidgest e comparação face às actuais integrações do Genio com RIA’s: Flash.

A procura de soluções a estes problemas obriga a adquirir um conhecimento funcional sobre Contabilidade, Controlo de Gestão e Gestão Estratégica de topo. Este *know-how* veio enriquecer a solução base de Gestão Financeira que acaba por diferenciar o produto final dos restantes existentes no mercado na medida em que estes são deficitários ou incompletos nestas componentes.

Em seguida, vamos referir-nos sumariamente aos diversos desenvolvimentos que são necessários levar a cabo para responder ao problema proposto, e que serão descritos em detalhe nos capítulos seguintes.

1.3.1 SOLUÇÃO BASE DE GESTÃO FINANCEIRA

O Sistema de Gestão Financeira desenvolvido tem a sigla CNT adoptada como um nome curto do seguinte nome de produto: “CNT - Contabilidade Financeira Empresarial”, sendo que iremos referir-nos ao Sistema como CNT daqui para a frente.

O CNT - “Contabilidade Financeira Empresarial” distancia-se dos seus demais do mercado na forma como integra numa só aplicação a Contabilidade (Geral e Analítica) com a Contabilidade Orçamental e o Controlo de Gestão. Isto faz com que os ganhos de aprendizagem possam ser partilhados entre todos os “actores financeiros” de uma organização tais como o gestor, o contabilista, o responsável financeiro ou o operacional.

Além do desenvolvimento de todas as funcionalidades base e requisitos definidos como essenciais para o CNT, tínhamos que deveria ser desenvolvido um Motor de Classificação Automática de Documentos Contabilísticos. Apesar de ser esta a nomenclatura utilizada para definir esta funcionalidade na definição do produto, entendemos que ela poderá induzir o leitor em erro na medida em que a classificação não é automática apesar de aparentar sê-lo, uma vez que os documentos não requerem a sua definição. Contudo, como iremos analisar no Capítulo 3, esta classificação requer uma classificação prévia por parte dum contabilista. É com base nesta associação de dados prévia que são realizados os automatismos necessários a classificar o documento.

Estes requisitos devem ser desenvolvidos com recurso à ferramenta Génio para uma ou várias aplicações BackOffice (estratégia da empresa) apoiadas por rotinas manuais na linguagem de desenvolvimento C++ (estratégia da empresa). Todos os mapas e relatórios devem ser desenvolvidos com recurso à integração existente entre o Génio e a ferramenta *Crystal Reports* (estratégia da empresa).

Além de ser dirigido a um negócio muito específico (Gestão de Parques Tecnológicos), os requisitos base deste sistema também cumprem os requisitos definidos na medida em que contém funcionalidades que pretendem aumentar os tipos de utilizadores habilitados a interagir com este tipo de aplicações.

1.3.2 INTEGRAÇÃO COM BSC QUIDGEST

O BSC – Balanced ScoreCard é uma metodologia de Gestão estratégica que visa integrar as diferentes perspectivas existentes dentro duma organização na definição da sua estratégia, ao contrário das tradicionais análises de indicadores meramente financeiros. Através duma ferramenta proprietária que implementa esta metodologia (BSC Quidgest), pretende-se que seja construído um sistema de indicadores financeiros para integração dessa aplicação com a solução base do CNT. Pretende-se ainda verificar se tal é possível e se sim, como.

Esta integração permite-nos ter um Controlo de Gestão mais abrangente e interactivo, bem como fazer com que este seja mais específico em torno da estratégia que se pretende levar a cabo na organização.

1.3.3 INTEGRAÇÃO MICROSOFT SILVERLIGHT COM CNT

O Microsoft Silverlight é uma tecnologia que implementa Rich Internet Applications (RIA's). Possibilita-nos o acesso a uma série de funcionalidades avançadas de processamento gráfico como a inclusão de renderização gráfica e clips média em páginas web. No âmbito deste trabalho, foi proposto o desenvolvimento duma investigação sobre a integração de Silverlight em aplicações geradas com recurso ao Génio. Este estudo requer a compreensão das actuais integrações de tecnologias similares com o Génio para fins semelhantes (ex.: Flash) com a realização dum estudo comparativo entre a utilização das duas tecnologias. É também necessário averiguar da exequibilidade da integração do Microsoft com a ferramenta BSC Quidgest, que já está desenvolvida com recurso ao Flash,

bem como da integração do módulo de Controlo de Gestão do CNT, previamente desenvolvido em Crystal Reports, com Microsoft Silverlight.

Em relação a estes tópicos é necessário averiguar se é possível realizar as integrações, e se sim, se é ou não rentável fazê-lo e qual o esforço associado face à manutenção das integrações já existentes, no caso, com o Flash.

Aqui, é a mais valia que a Riqueza (qualidade da experiência de utilização e interacção) inerente à utilização duma aplicação deste tipo confere a funcionalidades tipicamente “cinzentas” como o Controlo de Gestão. Com base nesse desenvolvimento, é necessário efectuar um estudo comparativo entre o desenvolvimento de aplicações de backoffice com Genio, num contexto de *data binding*, recorrendo ao Flash ou ao Silverlight, concluindo assim qual das tecnologias RIA's deve ser utilizada no futuro para implementações do género.

1.4 MOTIVAÇÃO

Este projecto coloca vários aspectos motivadores para a empresa. Antes de mais, a empresa não possuía qualquer solução de Gestão Financeira vocacionada para o mercado empresarial. O desafio lançado foi encarado como a oportunidade ideal não só para entrar no mercado, mas também para o fazer duma forma diferenciadora em relação à restante concorrência.

A criação duma aplicação que possuisse uma integração plena e distinta com o Controlo de Gestão e que permitisse, através de diversos automatismos, que o utilizador não tivesse que possuir conhecimento funcional para utilizar a aplicação é, a todos os níveis, um conceito interessante. A integração do Controlo de Gestão com o recurso a RIA's permite que sejam estudadas as integrações existentes, bem como as outras possibilidades existentes. Este é especialmente relevante na medida em que permite criar *know-how* nesta área que, doutra forma, não existiria. Os automatismos que a aplicação oferece permitem que esta se distinga no mercado do ponto de vista da utilização: ao alargar o leque de utilizadores, a aplicação torna-se mais abrangente, aumentando assim o número de potenciais clientes.

1.5 ORGANIZAÇÃO DO PRESENTE RELATÓRIO

Este relatório encontra-se dividido em sete capítulos e inclui ainda um anexo.

No segundo capítulo efectuamos a revisão tecnológica, em que analisamos de forma geral as tecnologias que vamos utilizar.

No terceiro capítulo, analisamos de forma sumária a solução base desenvolvida para que possamos compreender os desenvolvimentos efectuados no sentido desta responder aos requisitos previamente definidos.

No quarto capítulo, apresentamos a metodologia Balanced ScoreCard, bem como a solução da Quidgest para a implementar. Finalmente, apresentamos um Sistema de Indicadores Financeiros que permitem integrar a aplicação com o nosso Sistema de Gestão Financeira.

No quinto capítulo, analisamos a integrações do Genio com RIA's. Começamos por analisar a integração existente com Flash, o Controlo de Gestão na solução base e o que dele se pretende para a integração com Silverlight, o estudo da possibilidade de integrar Silverlight com o Balanced ScoreCard e a solução de Controlo de Gestão desenvolvida em Silverlight. Para finalizar, apresentamos uma comparação exaustiva entre Flash e Silverlight no contexto de desenvolver aplicações com *data binding* em aplicações integradas com o Genio.

No sexto e último capítulo, apresentam-se as conclusões do projecto apresentando os principais desenvolvimentos, as opções tomadas e os resultados obtidos, para que em seguida se defina o que deveria ser feito no futuro em relação ao presente problema.

Em anexo, é apresentado algum código fonte relevante para a melhor compreensão dos desenvolvimentos efectuados, bem como alguns conteúdos que espelham alguns conceitos com mais detalhe.

CAPÍTULO 2 - REVISÃO TECNOLÓGICA

Como já verificámos na secção “1.2 Descrição do Projecto”, o problema apresentado foi o de desenvolver um Sistema de Gestão Financeira para um Parque Tecnológico que se designa por CNT – Contabilidade Financeira Empresarial. Contudo, foram propostos alguns desenvolvimentos sobre a solução base com vista à diferenciação deste produto no mercado. Neste capítulo, vamos descrever de forma sumária o que se pode esperar, de forma generalista, dum Sistema de Gestão Financeira, para que definamos em seguida as particularidades da nossa solução e as tecnologias que a vão implementar.

Sistema de Gestão Financeira

Tipicamente, um Sistema de Gestão Financeira é um sistema que se destina a apoiar a gestão de Contabilidade Geral, Analítica e Orçamental bem como a gestão de Tesouraria de uma ou mais organizações. Estes módulos podem estar, ou não, integrados na mesma aplicação. Este apoio passa pelo suporte informático e automatização de vários processos de contabilidade. A automatização por estas funcionalidade passa por identificar e processar, tanto quanto possível, os processos contabilísticos que podem ser automatizados, diminuindo a necessidade do utilizador possuir uma formação funcional de Contabilidade para poder interagir com a aplicação. Apesar de existirem no mercado várias aplicações que oferecem uma resposta diversificada a este tipo de questões, existem ainda bastantes falhas nessas respostas que são ainda demasiado generalistas para resolverem problemas de mercados muito específicos, como o Controlo de Gestão e a sua integração com a contabilidade geral.

Nesse sentido, vamos descrever em seguida a nossa solução particular dum Sistema de Gestão Financeira.

2.1 SISTEMA DE GESTÃO FINANCEIRA

O que faz este Sistema diferente dos existentes do mercado?

A maior parte das soluções existentes no mercado apenas respondem a um de dois problemas: o Problema da Gestão da Orçamentação e Contabilidade Geral e Analítica, com a visão das rubricas e das contas POC; o Problema do Controlo de Gestão, com a visão da Contabilidade Geral cruzada com grandezas funcionais (ex.: Actividades). Este sistema de Gestão Financeira responde às necessidades dos dois mundos numa só aplicação integrada, além de ter uma série de mecanismos que permitem a utilização do sistema por parte de pessoas que não tenham um conhecimento Funcional de contabilidade.

Todo o projecto é desenvolvido recorrendo a uma ferramenta proprietária de Geração Automática de Código (ferramenta CASE): o Génio. O Génio permite-nos eliminar com um elevado grau de eficiência as redundâncias normalmente encontradas no processo de desenvolvimento de software, acelerando todo o processo de produção de código-fonte. Esta permite ainda alterações na estrutura do Projecto sem que isso atrase de forma considerável o processo de desenvolvimento, fazendo com que este possa ser muito mais iterativo e incremental, envolvendo o utilizador na definição dos requisitos (requisitos evolutivos) e no teste do próprio software, fazendo com que todas as aplicações desenvolvidas com o Génio utilizem uma metodologia de desenvolvimento *RAD (Rapid Application Development)*.

Sobre esta solução base, pretende-se que seja implementada uma solução de Controlo de Gestão que, através da construção de diversos mapas em Crystal Reports a partir de dados existentes no sistema, possa constituir um sistema de apoio à decisão do Gestor. Esta solução terá de ser inovadora do ponto de vista da interactividade e da riqueza de utilização da aplicação, devendo ser desenvolvida com recurso a uma tecnologia que implementa RIA: *Microsoft Silverlight*.

Outro dos módulos que se pretende desenvolver é o Motor de Classificação Automática de Documentos Contabilísticos. Este módulo vai consistir no lançamento automático dos movimentos contabilísticos a partir duma classificação prévia dos produtos utilizados.

Pretende-se ainda desenvolver o Sistema de Construção de Indicadores Financeiros a partir do sistema de gestão financeira com vista à sua integração com a tecnologia Balanced ScoreCard Quidgest. Esta ferramenta proprietária pretende implementar uma metodologia desenvolvida por Kaplan e Norton em 1992, com vista a melhorar a gestão estratégica de topo das organizações com base não só em indicadores financeiros mas também em indicadores intangíveis, como por exemplo, o registo da assiduidade dos funcionários e o número de normas de segurança cumpridas. Preparada para ser integrada com ERPs (*Enterprise Resource Planning*) devido à sua diversidade de informação, a integração desta tecnologia com o sistema de Gestão Financeira terá ao seu dispor apenas informação de carácter financeiro, sendo tomados em conta apenas critérios de carácter tangível.

2.1.1 TECNOLOGIAS UTILIZADAS PARA A IMPLEMENTAÇÃO DA SOLUÇÃO

O estudo das tecnologias que são utilizadas na implementação da solução fica desde logo condicionado pela opção estratégica da Empresa: utilização da ferramenta proprietária Génio para desenvolvimento do produto; utilização da tecnologia Crystal Reports para construção dos relatórios e dos Mapas de Gestão com os Indicadores Financeiros; evolução do Controlo de Gestão da análise de mapas estáticos para a utilização de *RIA's* com recurso a uma tecnologia recente na área: Microsoft Silverlight. Podemos verificar a forma como estes componentes interagem na Figura 1.

Contudo, a empresa já apresenta bastantes aplicações em que o Génio é utilizado em conjunto com *Adobe Flash* (outra tecnologia que nos permite desenvolver *RIA's*). É imperativo compreender se o *Microsoft Silverlight* é uma evolução natural, uma nova possibilidade ou uma ferramenta que não é tão eficaz a resolver o mesmo problema. Outras possibilidades para desenvolver *RIA's* são as *Java Applets*, *Java Script*, Controlos *ActiveX* ou *AJAX* que, contudo, não irão ser abordadas neste relatório pois a sua implementação não constitui uma hipótese estrategicamente válida para a empresa.

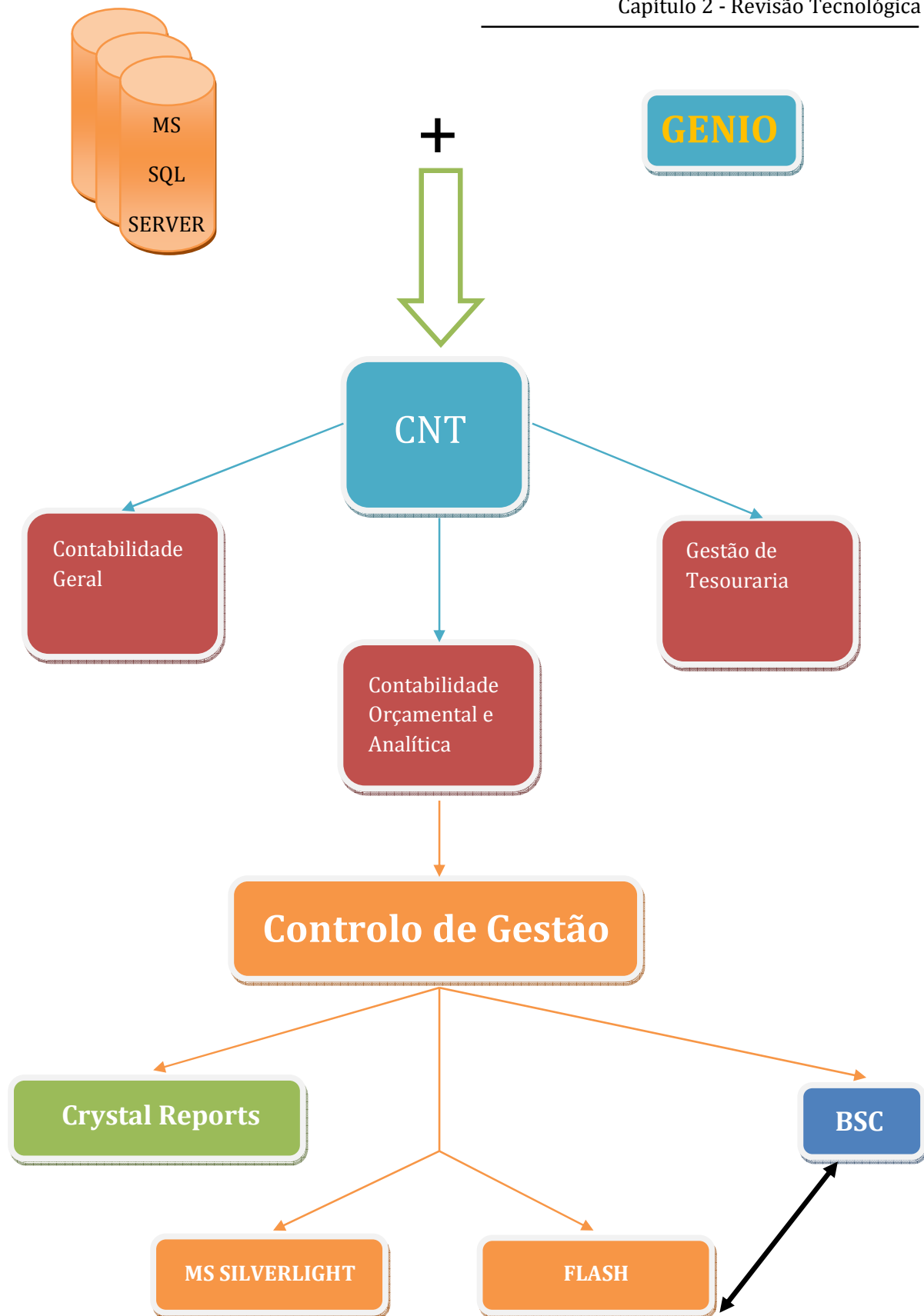


FIGURA 1 - TECNOLOGIAS A UTILIZAR PARA IMPLEMENTAÇÃO DO SISTEMA DE GESTÃO FINANCEIRA.

Apesar do Projecto ter uma abrangência superior, aqui irão ser essencialmente focados duas tecnologias desenvolvidas no Projecto por se considerarem dotadas de uma dificuldade tecnológica superior:

- O Balanced ScoreCard Quidgest, que implementa um Controlo de Gestão abrangente com recurso a mapas desenvolvidos em Flash;
- O Microsoft Silverlight, que pretende servir de base a uma aplicação de Controlo de Gestão com uma grande experiência de utilização: Rica, Interactiva e Inovadora;

Pelo facto de ambas as tecnologias (Flash e Microsoft Silverlight) implementarem *Rich Internet Applications*, descreve-se em seguida esse conceito.

2.2 RIA – RICH INTERNET APPLICATION

Uma *RIA* – Rich Internet Application é uma aplicação web em tudo semelhante às aplicações em backoffice. Contudo, uma *RIA* apresenta um modo de funcionamento em tudo diferente às tradicionais aplicações *web*. Na execução duma aplicação web, temos que esta possui, tradicionalmente, uma arquitectura cliente-servidor. Normalmente, neste tipo de aplicações, verificamos que é o servidor que se encarrega de todo o processamento e armazenamento de dados, sendo que o cliente apenas apresenta um conjunto de código *HTML* estático. Numa *RIA* isto não acontece: do lado do cliente, é utilizada uma tecnologia específica (*Adobe Flash* ou *Microsoft Silverlight*, por exemplo) permitindo assim que o processamento de dados se faça do lado do cliente. O servidor, por sua vez, funciona apenas como um repositório de dados.

Quais as vantagens da utilização de *RIA*'s face a uma aplicação backoffice tradicional?

A facilidade de migração dum sistema. Enquanto numa instalação duma aplicação backoffice tradicional temos que garantir que os sistema operativo possui um conjunto de características para que uma aplicação corra normalmente (entradas de registo, livrarias, serviços activos, etc.), tal não acontece com uma *RIA*'s, que tem como ponto de acesso qualquer computador com um *browser* e um acesso à Internet.

E quais as vantagens da utilização de *RIA's* face a uma aplicação web tradicional?

O tempo de troca de informação. Este tipo de aplicações separa o processamento dos dados, tornado assim mais eficaz a contribuição do cliente e do servidor na aplicação, sendo que diminui de forma considerável a latência antes existente. Com uma aplicação *web* tradicional existem constantes trocas de dados que, desta forma, não se verificam.

A comunicação entre cliente e servidor deixa de ser tradicionalmente síncrona para ser assíncrona, tendo grande parte do processamento do lado do servidor.

Com isto, temos um aumento do desempenho enorme que potencia também a qualidade da experiência do utilizador. Esta área ganhou ainda mais ênfase com o aparecimento das *RIA's 2.0*, ou de segunda geração. A disponibilização de tecnologias com *API's* (*Application Programming Interface*) mais poderosas do ponto de vista de interactividade do utilizador veio fazer com que seja possível, desta forma, desenvolver aplicações *web* com um aspecto em tudo semelhante a uma aplicação *backoffice*.

A utilização de *RIA's* tem desvantagens?

Sim. A segurança que está normalmente imposta nos clientes limita as funcionalidades das *RIA's* em relação às aplicações *backoffice*. Uma das funcionalidades que normalmente surge limitada prende-se com o acesso a dados armazenados localmente. O facto de a maior parte das tecnologias que suportam *RIA's* necessitarem de *scripting* e de páginas com tamanhos significativamente superiores, que provocam tempos de *loading* mais longos, são aspectos negativos que também deverão ser considerados

2.3 GÉNIO

O Génio é uma ferramenta de Geração Automática de Código proprietária.

Esta contém *templates* para gerar aplicações (*source code*) para várias linguagens padrão, entre as quais C# (web) e C++ (backoffice com MFC), utilizando vários tipos de bases de dados (ex.: *SQL Server*, *DB2* ou *Oracle*). Esta ferramenta *CASE* distingue-se das muitas presentes no mercado por não apresentar qualquer tipo de dependência de diagramas auxiliares (*ER* ou *UML*). Contudo, possui um conjunto de conceitos auxiliares que permitem apoiar não só a compreensão do sistema, como também a sua criação.

O seu sistema de criação de menus e formulários é visual, semelhante ao utilizado pelo Visual Studio e próximo do utilizador. Este conjunto de características próximo do que seria um sistema RAD faz com que o Génio gere com bastante facilidade protótipos funcionais, deixando que o utilizador participe de forma decisiva na elaboração do sistema final mais adequado às suas necessidades.

2.3.1 ESPECIFICIDADES TÉCNICAS

Cada aplicação gerada em Génio tem a si associadas duas Bases de Dados. Uma de definições e uma de dados propriamente ditos. A Base de Dados de definições (exemplo na Figura 2) guarda tudo aquilo que é parametrizável e desenvolvido no Génio, desde controlos de formulários, passando por tabelas (áreas e domínios), menus, ordens, ajudas, relações entre tabelas, entre outros. Esta Base de Dados é consultada quando é iniciado o processo de geração de código. Este processo limita-se a substituir as *tags* presentes nos *templates* pelo conteúdo da Base de Dados. No limite, uma parte dum campo dum determinado registo numa determinada tabela poderá ser uma *tag*.

Table - dbo.GENTabel					
Summary					
	NDBF	NUM	NUMALX	DESDBF	GEN
	ANO2	30	0	Anos dos exercicios económicos	o
	HMBKG	145	75	Ficheiros de Home-Banking]	o
	RECON	222	127	Reconciliações bancárias	a
	FIVA	92	56	Folhas de entrega de IVA	a
	FIV1	110	58	Folhas de entrega de IVA	a
	COUST	136	26	Centros de custos	o
	PORN2	113	65	Terceiros	o
	LCPT1	260	0	Ligação entre centros de custos ...	
	AFPC0	122	6	Esquemas de afectação a centro...	o
	MSIVA	15	98	Mensagens de isenção de IVA	a
	ACT11	100	5	Actividades	a
	FOUH2	185	61	Pagamentos e recebimentos	o

FIGURA 2 – ÁREAS DE GÉNIO MAPEADAS EM SQL SERVER.

A outra Base de Dados serve essencialmente para guardar os dados respeitantes à aplicação utilizada no momento. Normalmente, quando a aplicação está em produção, existe um, ou mais, servidores que contém esta segunda Base de Dados, sendo que a das definições (a primeira) não faz parte integrante do produto.

Para tudo aquilo que não é possível definir com Génio, existem as denominadas rotinas manuais. As rotinas manuais são blocos de código definidos em ficheiros à parte devidamente identificados e criados para o efeito onde é possível desenvolver código manualmente, tal como podemos verificar na Figura 3. Este código é depois acoplado ao restante durante a geração pois encontra-se identificado com *tags*.

```

102
103  / [MANUAL CNT ROTINA OFERTPRN] /
104  {
105      //Imprime uma oferta, a partir do form
106      CWaitCursor Wait;
107      if (!ValidaCtrls()) return;
108
109      //Grava a ficha e garante o posicionamento correcto
110      CString codencom=pDoc->encom->m_codencom;
111      pDoc->encom->DoUpdate();
112      if (!pDoc->encom->Search(NULL,codencom,"",FALSE)) return;
113      pDoc->encom->DoEdit();

```

FIGURA 3 – CÓDIGO TÍPICO DUMA ROTINA MANUAL EM GÉNIO.

Dentro do Génio, podemos definir três grandes áreas de implementação: o modelo relacional, os formulários e o sistema de menus. Normalmente, a definição da lógica de negócio existente fica “alojada” no modelo relacional, funcionando o sistema de menus e o de formulários como de mera interface com o utilizador. Contudo, podemos também associar a lógica de negócio a menus e a formulários através de uma adequada definição da política de acessos, i.e., diferenciando os objectos a que cada utilizador tem acesso em função das permissões que lhe foram atribuídas pelo administrador do sistema.

Na definição do modelo relacional, dos menus e dos formulários existe uma série de conceitos distintos dos conhecidos que serão agora denominados de conceitos proprietários. Estes conceitos dizem respeito a funcionalidades únicas do Génio e conceitos de organização do modelo relacional, dos menus, etc.. que são fundamentais estarem presentes para quem desenvolve com o Génio.

2.3.2 CONCEITOS PROPRIETÁRIOS

Numa ferramenta com estas características, existem vários conceitos proprietários. Em seguida vamos explicar sumariamente os mais importantes do ponto de vista do desenvolvimento do presente projecto.

Área O conceito de Área no ambiente de desenvolvimento Génio é um conceito que visa associar a classe de objectos com o contexto em que se insere, no seu domínio. Isto é, uma factura que paga algo num Restaurante não corresponde à mesma Área duma factura que paga algo numa Mercaria, assim como um Aeroporto de Chegada é diferente dum Aeroporto de Partida, mas as características dos dois Aeroportos são iguais. São utilizados apenas em contextos diferentes. Isto serve, por um lado, para a melhor identificação funcional das classes na Base de Dados e, por outro, para definir a estrutura da Base de Dados do sistema a gerar. Com a utilização das áreas, garantimos a manutenção da consistência, isto é, não existe no modelo relacional do sistema dois caminhos diferentes entre as mesmas tabelas/domínios.

Domínio Um Domínio é a definição da Classe de Áreas. Isto é, se tivermos duas Áreas Aeroporto de Partida e de Chegada, estas serão Áreas do Domínio Aeroporto. É no domínio

que devemos definir as características das Áreas (campos, etc..), uma vez que as áreas têm exactamente as mesmas características.

Fórmulas As fórmulas são expressões que nos permitem calcular o valor do campo em causa. Contudo, este conceito surge expandido no Génio devido às suas características únicas.

Ex.: É possível definir uma Soma Relacionada. Esta Soma Relacionada não será mais do que a soma de um conjunto de valores dum conjunto de registos numa tabela associada abaixo (N para um). Podemos também definir uma Réplica, que conterà um campo que está numa área referenciada acima. Esta Réplica irá garantir que o valor desse campo será sempre o mesmo do valor do campo replicado da área referenciada.

Modelo Relacional e Chaves Estrangeiras Para “simplificar” o conceito de Modelo Relacional, o Génio tem convencionado que a cardinalidade de todas as relações existentes é de N para 1, isto é, de baixo para cima, como podemos verificar na Figura 4. Os modelos relacionais são construídos desta forma para que se possa compreender que áreas podem ser referenciadas a partir de outras. Neste caso, serão sempre aquelas que estiverem acima na árvore de relações. As relações são definidas através da inclusão de campos do tipo Chaves Estrangeiras.

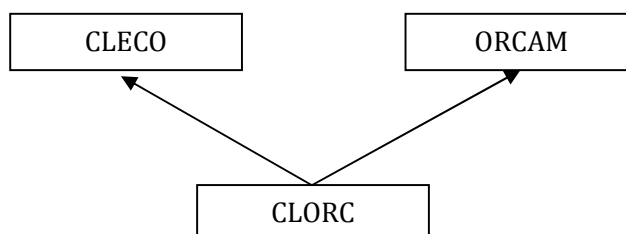


FIGURA 4 – RELAÇÃO TÍPICA DE N PARA N ENTRE DUAS ÁREAS EM GÉNIO. A SETA ESPELHA LIGAÇÕES DE N PARA 1, NO SENTIDO DA SETA. NESTE CASO, PODEMOS DEFINIR QUE AS RUBRICAS (CLECO) PODERÃO TER VÁRIOS ORÇAMENTOS ASSOCIADOS (ORCAM) E VICE-VERSA.

DBEdit O DB Edit é um controlo do Génio que é criado em forma de menu. Na prática, este controlo traduz-se num formulário totalmente parametrizável mas que inclui uma listagem dos campos numa determinada área. Podemos escolher qual o formulário de apoio para inserção e alteração de dados, bem como os botões a incluir e os campos a listar. Este formulário tem a particularidade de condicionar e ser condicionado pelas escolhas de outros

DB Edit, desde que as áreas estejam relacionadas entre si.

Ex.: Um menu cuja continuação seja um DB Edit de Pessoa e depois um de Factura, em que Factura→Pessoa, isto é, cada Factura tem a si associada uma pessoa. Nós vamos ter a oportunidade de escolher um dos vários registos da Área Pessoa mas depois, o DB Edit seguinte listará apenas as facturas com relação com o registo da Área Pessoa escolhido anteriormente.

Expõe Tabela Tal como o nome indica, este controlo é responsável por expôr uma determinada tabela/área num Formulário. Esta área terá de ter algum tipo de relação (directa ou indirecta) com a área do formulário em questão. A função deste controlo é expor o conteúdo da tabela em questão, seleccionando apenas os registos que tem associação com o registo presentemente aberto no corrente formulário. Podemos também inserir ou alterar registos dessa tabela, sendo que tais inserções e alterações estarão sempre condicionadas ao registo da Área aberta no Formulário.

Listagens Uma Listagem é aquilo que é vulgarmente conhecido como um Relatório no contexto duma Base de Dados. Esta é efectuada com recurso à ferramenta Crystal Reports que está devidamente integrada com o Génio. A diferença é que com o Génio temos a possibilidade de definir um menu do tipo listagem. Este menu terá a necessidade de incidir sobre uma determinada Área. Contudo, e quando utilizado em conjunto com outro tipo de menus (como no exemplo da Figura 5), esta pode ser uma ferramenta extremamente poderosa.

Ex.: Se tivermos inicialmente um DBEdit que nos liste Cidades, continuado dum DBEdit que tenha Restaurantes e Uma Listagem de Facturas com uma condição associada (ex.: valor>50 euros), então teremos que escolher inicialmente uma cidade. Depois poderemos escolher um dos Restaurantes dessa cidade (apenas esses serão mostrados). Finalmente, aparece uma listagem das Facturas desse Restaurante que foram superiores a 50 euros. Agora podemos também retirar o DBEdit de Restaurante e definir um intervalo de datas nas Listagens. Definimos que queremos Facturas com valor superior a 150 euros. A aplicação apenas irá listar Facturas de Restaurantes da Cidade Escolhida com valor superior a 150 euros emitidas no último mês.

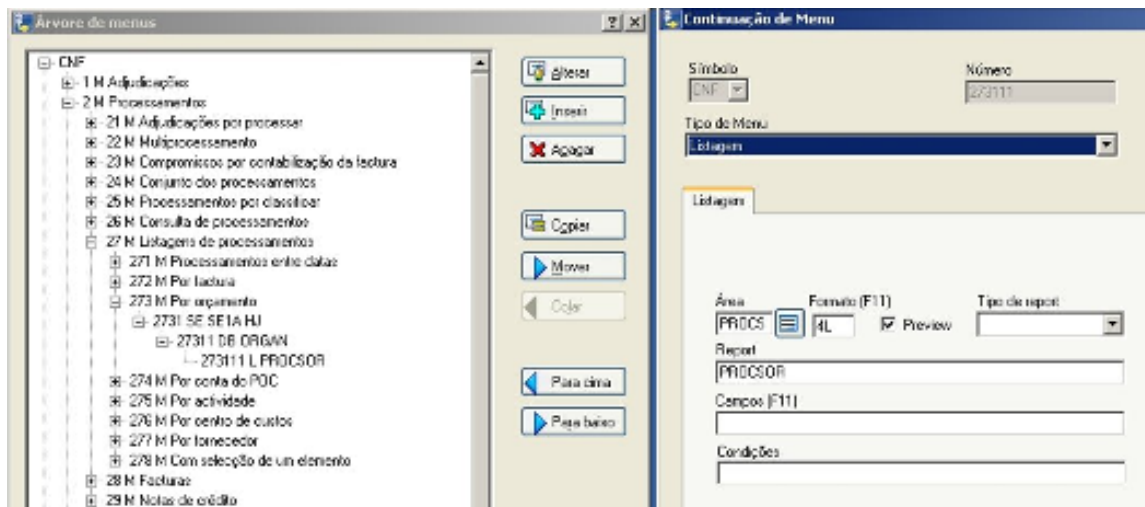


FIGURA 5 – DEFINIÇÃO DUMA LISTAGEM EM GÉNIO.

2.4 CRYSTAL REPORTS

O Crystal Reports é uma aplicação de Business Intelligence. Este tipo de aplicações é normalmente utilizado na colecção, integração, análise e apresentação de informação para sistemas de suporte à decisão. Originalmente desenvolvida pela Business Objects, esta aplicação é utilizada para desenvolver e gerar relatórios a partir de vários tipos de fontes de dados. Várias aplicações utilizam versões OEM (*Original Equipment Manufacturer*) do Crystal Reports como a sua própria ferramenta de geração de relatórios.

Neste contexto, temos que utilizar o Crystal Reports utilizando a integração existente entre este e a ferramenta proprietária de geração automática de código – o Génio.

2.5 BALANCED SCORECARD

O Balanced ScoreCard é uma metodologia disponível no mercado à mais de 15 anos que pretende ser um sistema de avaliação de desempenho e de gestão estratégica. Foi inicialmente desenvolvido por Robert Kaplan e David Norton, professores da Harvard Business School. Esta metodologia distancia-se dos restantes sistemas de gestão estratégica pelo facto de contemplar objectivos e métricas de carácter não financeiro, abrangendo várias perspectivas da organização. Implementar Balanced ScoreCard inclui os seguintes

processos: passar a visão estratégica para objectivos operacionais (mensuráveis); comunicar a visão e ligá-la ao desempenho individual dos membros da organização; definir Indicadores Estratégicos; obter o retorno e retirar daí conclusões para aprendizagem, ajustando a estratégia em conformidade.

No presente relatório, quando falamos em Balanced ScoreCard (BSC) num contexto tecnológico, falamos numa ferramenta proprietária denominada “Balanced ScoreCard Quidgest”, que não é mais que um produto que a empresa tem no mercado.

Este sistema de informação de apoio à Gestão Estratégica tem como principal objectivo integrar-se com outras aplicações (desenvolvidas com Génio ou não) para conferir toda a informação necessária a uma Gestão Estratégica de topo. Além de possuir várias funcionalidades como a construção do mapa estratégico, possui um ambiente gráfico bastante intuitivo (além dos usuais formulários) para compreender os diferentes mapas e indicadores. Este tipo de mapas é utilizado através da integração existente do Adobe Flash com as aplicações desenvolvidas em Génio, dando funcionalidades (como o *drill-down*) que doutra forma seriam difíceis de implementar numa aplicação apenas de puro *backoffice*.

2.6 ADOBE FLASH

O Adobe Flash é uma plataforma multimédia criada pela Macromedia à cerca de dez anos e mantida actualmente pela Adobe. O Flash permite adicionar apresentações com um grande nível de animação e interactividade a aplicações. As utilizações mais comuns são animações, anúncios publicitários e integração de clips de vídeo em páginas web. Mais recentemente, o Flash tem sido utilizado para desenvolver *RIA*'s. O Flash pode manipular vectores e gráficos e contém uma linguagem de *scripting* denominada de *ActionScript*. A grande vantagem dos ficheiros SWF (*ShockWave Flash*, o tipo de ficheiros que suporta o Flash) é que suporta *streaming* “bi-direcional” de áudio e vídeo, apoiando assim a interactividade do utilizador com aplicação.

Actualmente, existem várias integrações de Macromedia Flash nas aplicações geradas com Génio, tendo como objectivo aumentar tanto a interactividade com o utilizador, como a qualidade da apresentação.

Neste momento, o Génio utiliza apenas componentes fixos “*hardcoded*” em Flash. Isto é, é necessário ter um ficheiro Flash com um conjunto de características fixas que depois se torna num tipo de campo disponível para utilizar nos formulários. As características são fixas porque esse objecto SWF (*ShockWave Flash*) já está compilado e com um conjunto de informações predefinidas. Tendo em conta apenas a geração de aplicações para BackOffice, podemos definir que a comunicação com o Génio é assíncrona¹ e realizada por eventos.

Existe uma classe no Génio (Figura 6) responsável por assegurar a manutenção do controlo responsável por tratar os objectos Flash denominada *CshockwaveFlash*. Depois em cada formulário gerado com o Génio existe um método denominado *OnFlashEvent* que não é mais que um *trigger* despoletado pelo objecto *Flash* através de métodos pertencentes à classe acima referida quando um evento acontece do lado do *Flash*.

```

////////////////////////////////////
// CShockwaveFlash - classe base do controlo Shockwav

#define GSW_GANTT          0x00000001 // Diagrama d
#define GSW_ORGANIGRAMA    0x00000002 // Organigram
#define GSW_FERIAS         0x00000003 // Férias
#define GSW_WORKFLOW       0x00000004 // Workflow
#define GSW_BSCARDS        0x00000005 // BScoreCard
#define GSW_BSHISTORICO     0x00000006 // BScoreCard
#define GSW_BSMAPAESTR      0x00000007 // BScoreCard
#define GSW_BSDASHBOARD    0x00000008 // BScoreCard
#define GSW_BSRELAT        0x00000009 // BScoreCard
#define GSW_BSGAUGE         0x0000000A // BScoreCard
#define GSW_AGENDA         0x0000000B // Agenda
#define GSW_CALENDARIO      0x0000000C // Calendário

```

FIGURA 6 - IMAGEM DO CÓDIGO FONTE DA CLASSE CSHOCKWAVEFLASH. ESTA CLASSE É RESPONSÁVEL PELA INTEGRAÇÃO DOS OBJECTOS .SWF.

Os objectos SWF utilizados nos campos dos Formulários do Génio estão preparados para receberem strings do tipo XML com a informação sobre o estilo e o conteúdo dos objectos

¹ A comunicação assíncrona é uma comunicação que é efectuada sem garantia de existir uma comunicação em sentido inverso simultânea ou imediata. Um bom exemplo de comunicação síncrona vs. comunicação assíncrona é o telefone e o e-mail, respectivamente. Neste caso, tal significa que não iremos ter retorno do estado de processamento da resposta do servidor ou de quando esta irá acontecer até que a recebamos.

que devem ser mostrados no Flash. É assim que se dá a comunicação entre as apresentações Flash e a aplicação gerada pelo Génio. Temos de recorrer a uma rotina manual com uma *tau* específica para preencher a função *OnFlashEvent* de forma a tratar todos os eventos que aconteçam em Flash e preparar strings em XML que permitam alterar a apresentação em função disso. Inicialmente, a aplicação começa por despoletar um evento inicial que indica quando esta está pronta para receber XML. Depois esta função tratará o evento em conformidade, fazendo as alterações necessárias na base de dados, no Flash ou fazendo outro evento no Génio (por exemplo, chamar um formulário).

2.7 MICROSOFT SILVERLIGHT

O Silverlight é um plug-in multi-browser (Opera não suportado) e multi-plataforma (Linux não suportado²) programável que disponibiliza funcionalidades como animações, gráficos vectoriais e clips de vídeo e áudio que são típicos das RIA's. O seu modelo de programação suporta AJAX, .NET e linguagens dinâmicas como Python ou Ruby.

Perto de completar dois anos de existência, o Silverlight veio competir num mercado, as RIA's, em que o Flash foi durante muito tempo um competidor isolado.

A sua utilização em aplicações web tem crescido ao longo do tempo, nomeadamente em aplicações que privilegiem experiências de interacção com o utilizador Ricas.

Riqueza das Aplicações – Um Novo Conceito

Quando se define uma aplicação como Rica ou a própria Riqueza da aplicação, temos um conceito recente. Por Riqueza, entende-se a qualidade da experiência de utilização que a aplicação oferece ao utilizador. Experiência gráfica mas sobretudo da interacção, onde o utilizador não necessita de conhecer as limitações da aplicação para a poder definir o quão bem a pode utilizar. Um exemplo disso mesmo poderá ser, por exemplo, o website com o Memorabilia do HardRock [har08].

² Existe neste momento um projecto que utilizando uma versão do .NET Framework denominada “mono” prevê desenvolver uma tecnologia semelhante ao Silverlight. Este Projecto é denominado de “Moonlight”.

Windows Presentation Foundation – A base tecnológica

O *Windows Presentation Foundation* (WPF) é um subsistema gráfico pertencente à framework .NET a partir da sua versão 3.0. Esta tecnologia permite-nos desenvolver aplicações Ricas para desktop com recurso a desenho 2D e 3D, gráficos vectoriais, animações, *data binding*, reprodução de áudio e vídeo, etc.. Esta tecnologia é a base tecnológica do Microsoft Silverlight. Esta acaba por ser uma tecnologia mais leve e multi-plataforma que utiliza como base as potencialidades do WPF, sendo que muitas das funcionalidades se encontram reduzidas nesta plataforma.

Silverlight - Executado num *Plug-in*

Qualquer computador necessita de ter instalado um *Plug-in* do Silverlight para que este funcione. Este plug-in é de fácil instalação e inicialização. Apesar de ser multi-plataforma, este *Plug-in* está optimizado para um sistema com o sistema operativo Microsoft Windows (XP+) e o *browser* Internet Explorer (6.0+).

2.7.1 ARQUITECTURA

As principais funcionalidades do Silverlight são asseguradas por um *plug-in* para um browser que renderiza XAML – Extensible Application Markup Language. O XAML é uma ferramenta baseada em XML que é utilizada para definir os recursos visuais da aplicação. Isto inclui interfaces com o utilizador (UI – user interfaces), recursos gráficos, animações, media, controlos, etc.. Inicialmente lançado com o WPF, foi criado para diminuir a distância entre os designers e os programadores no desenvolvimento de aplicações.

A Arquitectura que suporta o Silverlight é ilustrada na Figura 7. Quando o controlo é definido no browser através de *scripting*, a principal interface de programação é Javascript DOM API, onde:

- DOM – *Document Object Model* é um modelo de objectos para representar documentos baseados em linguagens do tipo HTML e XML;
- API – *Application Programming Interface* é, neste contexto, um modelo de funcionalidades para consultar e manipular os referidos documentos;

Isto permite-nos apanhar diferentes tipos de eventos que aplicação cria como quando se clica em algo e queremos que ele execute um determinado bloco de código, por exemplo.

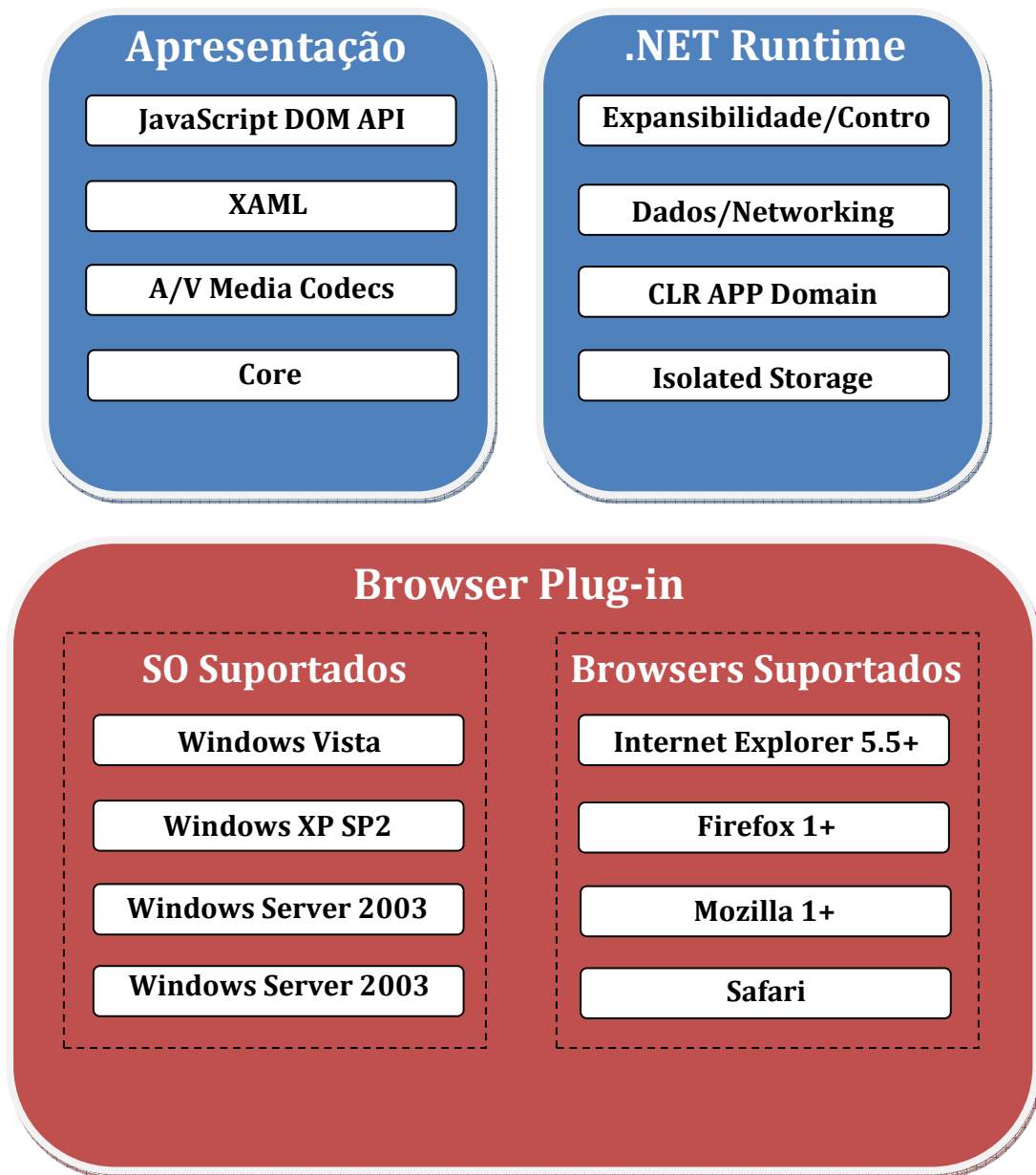


FIGURA 7 – ARQUITECTURA DO MICROSOFT SILVERLIGHT. [LM008]

Para uma experiência mais Rica, podemos programar uma aplicação com aceleração gráfica através da utilização da framework .Net CLR . Para além daquilo que é possível

fazer com uma linguagem de *scripting* como o *JavaScript*, esta contém framework muitos dos namespaces e dos controlos que vêm como parte do .NET framework para aceder a funcionalidades como a consulta de dados com o ADO.NET³ e o LINQ⁴;

2.7.2 PRINCIPAIS FUNCIONALIDADES

Algumas das principais funcionalidades do *Silverlight* são:

- Suporte para reprodução de ficheiros áudio e vídeo (incluindo WMV – Windows Media Video, WMA – Windows Media Audio e MP3 – MPEG-1 Audio Layer 3);
- Acesso controlado ao sistema de ficheiros do utilizador para arquivar dados fora da memória cache;
- Deep Zoom – Tecnologia exclusiva do Silverlight que permite a navegação em imagens de alta resolução, alterando rápida e repetidamente o seu valor de zoom, sem que seja necessário o total carregamento da imagem no Cliente;
- Mais de 30 controlos nativos para a interface;
- Suporte ao desenvolvimento a partir do Microsoft Visual Studio 2008 e Microsoft Expression Blend para diminuir as fronteiras entre os designers e os programadores.

2.7.3 MODELO DE PROGRAMAÇÃO

O Modelo de Programação em Silverlight é diferente da maior parte dos modelos programação. uma vez que se baseia em pelo menos duas linguagens quando o normal é ser em apenas uma. Uma delas, o XAML (Extensible Application Markup Language) é uma linguagem declarativa, baseada no XML (Extensible Markup Language) que é utilizada para inicializar valores estruturados e objectos. Baseia na estruturação das propriedades dos objectos e das instâncias dos controlos existentes. Uma linguagem adicional poderá ser tanto o CSharp⁵ como o VB .Net⁶. Por escolha estratégica da empresa, foi definido que a

³ - conjunto de componentes de software que podem ser utilizados por programadores para aceder a dados ou a SGBD, como o *Microsoft SQL Server*;

⁴ - *framework .NET* que nos permite realizar queries em *arrays*, XML, retornando uma colecção enumerável que pode ser acedida através dum iterador *standard* como o *foreach*;

⁵ Linguagem de Programação de alto nível introduzida na criação da plataforma .Net.

⁶ Linguagem de Programação orientada a eventos introduzida na criação da plataforma .Net.

linguagem a utilizar para o desenvolvimento do Silverlight será o CSharp, pelo que será esta a referida daqui para a frente.

O CSharp permite-nos definir tudo aquilo que acontece em *runtime*. Um exemplo clássico é uma instância dum controlo do tipo *Button*. No XAML é definido o posicionamento do botão, alinhamento, texto, cores, estilo, forma e o nome do procedimento a ser chamado quando, por exemplo, o evento “click” acontece. As operações realizadas aquando a realização desse evento (implementação do procedimento) são definidas em CSharp. É evidente que existem propriedades que apenas podem ser alteradas em *runtime*, bem como propriedades que apenas podem ser alteradas em *design*. Contudo, uma boa parte é editável nas duas visões, tornando o desenvolvimento em Silverlight bastante versátil.

O Silverlight possui um vasto conjunto de controlos que podem ser utilizados tais como o *Button* ou a *Datagrid*. Estes controlos têm de ser primariamente instanciados no *design* (XAML) para poderem ser manipulados em *runtime*.

2.7.4 MODELO DE UTILIZAÇÃO

A utilização de Silverlight faz-se recorrendo a um conjunto de tecnologias. Todavia, a base do seu design é desenvolvida em XAML. Podemos verificar na Figura 8 a declaração dum conjunto de tecnologias em XAML.

```
<TextBlock Margin="244,57,0,63" Width="57.401" TextWrapping="Wrap"
<TextBlock Height="16" HorizontalAlignment="Right" Margin="0,7,39.
<Button HorizontalAlignment="Right" Margin="0,28,39.959,0" Content
<Button HorizontalAlignment="Right" Margin="0,55,39.959,59" Conter
<Button HorizontalAlignment="Right" Margin="0,0,40.33" Content="Ak
```

FIGURA 8 – DEFINIÇÃO DE CONTROLOS SILVERLIGHT EM XAML.

Para podermos desenvolver e executar uma aplicação de Silverlight, é necessário:

- Possuir um browser suportado pelo Silverlight, tal como o Microsoft Internet Explorer;
- Ter instalado o plug-in do Silverlight para esse browser, na versão em que estamos a desenvolver;

- Possuir o Microsoft Visual Studio 2008 ou superior, com o *toolkit* de desenvolvimento para o Silverlight instalado;
- Desenvolver uma página XAML com o design e o respectivo código de implementação em *runtime*;
- Desenvolver um serviço ASP.NET numa página ASPX que possa actuar como servidor do componente Silverlight que desenvolvemos previamente;
- Possuir um servidor ASP.NET que nos permita executar convenientemente a página ASPX previamente desenvolvida;
- Se quisermos aceder/consultar dados repositórios de dados mais estruturados, tais como uma base de dados ou uma colecção de dados mais complexa, teremos sempre de recorrer a um *web service* que proceda à extracção desses dados. A comunicação desses objectos é depois feita entre o *web service* e o componente Silverlight através duma comunicação do tipo assíncrona. Os objectos enviados devem possuir a característica *Serializable* que os permita ser transmitidos através duma *network*.

O domínio da linguagem XAML e um bom conhecimento da plataforma .Net são essenciais para o desenvolvimento de aplicações em Silverlight. A grande vantagem da utilização deste tipo de parametrização é a possibilidade de integração de ferramentas distintas como o Microsoft Visual Studio e o Microsoft Expression Blend. Enquanto o Visual Studio é fortemente dirigido a programadores, o Expression Blend é dirigido principalmente para designers. As sinergias entre as duas ferramentas resultam numa experiência mais Rica para o utilizador.

2.7.4 WEB SERVICES E LINQ

Como já foi referido no Capítulo 2, o Silverlight corre numa espécie de “Caixa de Areia” que faz com que o seu acesso aos dados locais seja muito limitado. Normalmente, para que tal aconteça, o Silverlight deve comunicar com um ou mais *web services* para que lhe enviem os conteúdos. No caso dos conteúdos serem provenientes de um SGBD – Sistema de Gestão de Base de Dados, isto é mesmo fundamental.

Um *web service* é uma aplicação utilizada para a comunicação entre diferentes aplicações que são executadas, por exemplo, em diferentes plataformas. Os dados são enviados e recebidos tradicionalmente através de mensagens XML. Apesar das aplicações poderem ter as suas próprias linguagens, estas são normalmente compreendidas pelo *web service* e convertidas para XML para que as aplicações não se tenham de conhecer umas às outras mas apenas a sintaxe duma linguagem universal de comunicação.

Na versão actual do Silverlight 2.0 – Beta 2, este apenas suporta chamadas assíncronas⁷ aos *web services*. Uma vez que o Silverlight utiliza tecnologias .NET para a sua lógica de negócio, é aconselhável que se utilizem tecnologias .NET para os *web services*, nomeadamente a WCF – *Windows Communication Foundation*. A WCF é uma *framework* para construir comunicações que está embebida na *framework* .NET 3.5 para comunicações. Para que tal aconteça, é necessário recorrer a um servidor ASP.NET para executar as nossas aplicações num localmente.

Este *web service* vai-se encarregar de descriptar as mensagens recebidas do cliente e processá-las adequadamente, fazendo todos os pedidos necessários ao servidor. A *framework* .NET, na sua última versão (3.5), disponibiliza uma tecnologia integrada na linguagem de desenvolvimento para extrair informações de colecções de dados denominada LINQ – *Language Integrated Query*. O LINQ adiciona capacidades nativas de consulta de dados às linguagens .NET como CSharp e VB.Net. Daí que o *web service* a possa utilizar para questionar SGBD. Em particular, temos uma tecnologia dentro do LINQ designada LINQtoSQL, que permite ligar objectos em linguagens .NET a dados numa base de dados. A linguagem pode ir alterando os dados dos objectos assim como o seu estado, que o LINQtoSQL possibilita a submissão automática dessas alterações. Esta habilidade de tratar os dados como objectos facilita a comunicação de dados, uma vez desde que estes sejam serializáveis⁸.

⁷ - a tipologia duma chamada assíncrona duma funcionalidade define que não tenhamos qualquer control sobre e como e quando a funcionalidade é executada, qual o seu estado de execução e quando esta retornará o valor pretendido.

⁸ - Objectos serializáveis são objectos que são convertíveis numa sequência de bits para serem guardados num conjunto de dados estruturados ou ser transmitidos ao longo duma network.

Na Figura 9 temos uma arquitectura típica duma aplicação que utiliza Silverlight, WCF, LINQ e um SGBD (típicamente, SQL Server). A aplicação em Silverlight corre do lado do cliente e pede ao servidor, através dum *web service* de WCF, os conteúdos necessários à sua apresentação. O *web service* processa o pedido, ligando-se à base de dados através de LINQ, manipulando os objectos em *runtime* seja necessário. Findo o processo, o *web service* retorna os dados para o Cliente para que eles possam ser apresentados.

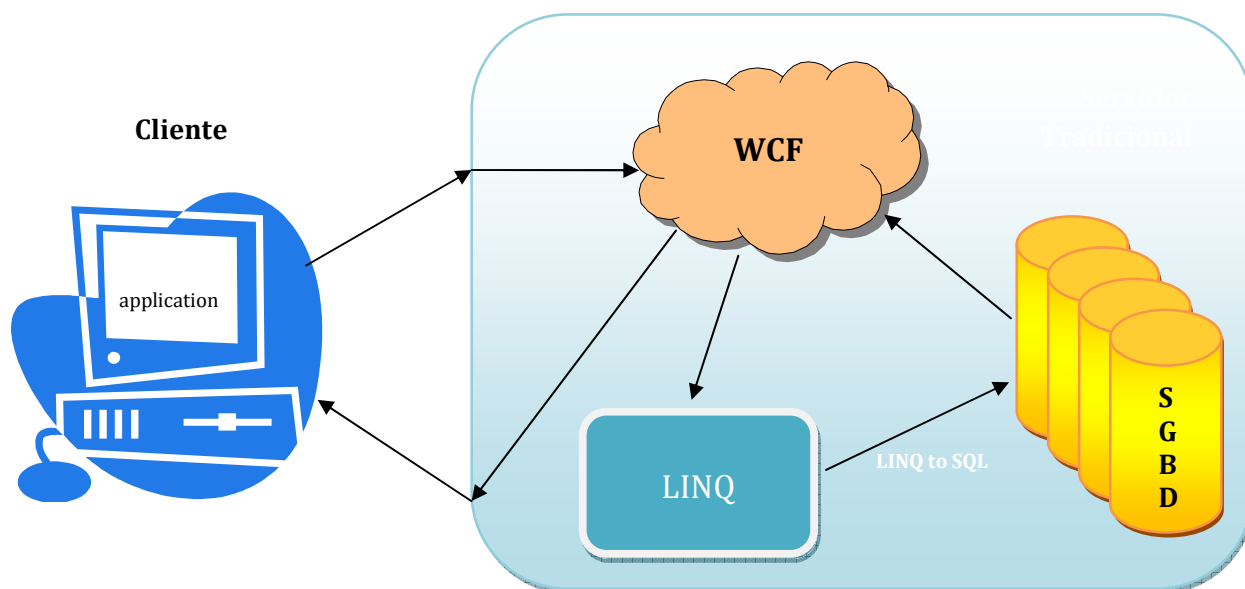


FIGURA 9 – ARQUITECTURA TÍPICA DUMA APLICAÇÃO EM SILVERLIGHT COM LIGAÇÃO A UMA SGBD.

Esta combinação de tecnologias maximiza as potencialidades do Silverlight, tornando-o uma tecnologia de construção de apresentações por excelência. Contudo, o carácter de testes de muitos dos seus controlos e funcionalidades (como a *datagrid*), confere-lhe alguma instabilidade, na medida em que torna as suas soluções pouco continuáveis ou limitadas do ponto de vista tecnológico.

CAPÍTULO 3 - DESCRIÇÃO DA SOLUÇÃO BASE

Neste Capítulo, vai-se expôr a solução base do Sistema de Gestão Financeira. Como já fora referido no Capítulo 1, o desenvolvimento dum Sistema de Gestão Financeira de raiz surgiu na sequência duma problemática que o mercado não conseguia resolver: a conciliação do Controlo de Gestão e da Contabilidade Geral e Analítica na mesma aplicação.

Procurava-se ainda um sistema fácil de utilizar, que dispensasse quase na totalidade o conhecimento funcional de contabilidade por parte dos seus utilizadores mais habituais e que fosse adaptado a uma realidade específica do mercado: a Gestão de Parques Tecnológicos.

Não iremos descrever o sistema em detalhe, uma vez que este capítulo apenas se destina a contextualizar o leitor em relação ao sistema base sobre o qual vão ser realizados os desenvolvimentos referidos no Capítulo 1. Em seguida, vamos expor uma visão geral sobre a arquitectura, o modo de funcionamento e o modelo de programação da aplicação, para explorar de forma sumária as funcionalidades base clássicas e com mais detalhe as que são diferenciadoras das restantes. Para concluir, vamos analisar um desenvolvimento em particular: o Motor de Classificação de Documentos Contabilísticos. Este destaque acontece por este ser um requisito de elevada importância do trabalho como mecanismo de criação de automatismos.

3.1 CNT – CONTABILIDADE FINANCEIRA EMPRESARIAL

O “CNT - Contabilidade Financeira Empresarial” é um sistema de gestão financeira composto por duas aplicações: o CNF e o GTS. O “CNF – Contabilidade e Gestão Financeira” é responsável por toda a Contabilidade de Gestão e Controlo de Gestão duma organização. O “GTS – Gestão de Tesouraria” é o módulo responsável pela gestão de tesouraria duma organização. Estes produtos, que têm como principal *target* de mercado empresas do sector privado, estão actualmente divididos nestes módulos devido a uma definição estratégica da empresa em relação aos seus principais clientes. Apesar de

utilizarem a mesma base de dados, o acesso às suas funcionalidade faz-se por duas aplicações diferentes. Estas definições são facilmente alteráveis com recurso à ferramenta de Geração Automática de código – o Génio, como já pudemos verificar na secção “2.3 Génio”. Os dados destas aplicações estão armazenados em bases de dados de Microsoft SQL Server 2005⁹ por opção estratégica da empresa. Podemos verificar a forma como a arquitectura da aplicação associa as diferentes tecnologias na Figura 10.

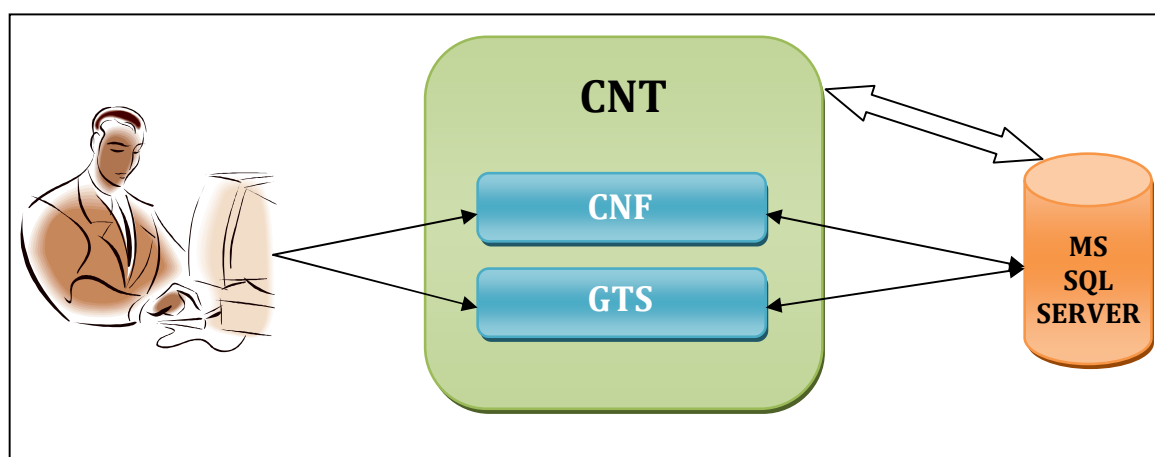


FIGURA 10 – ARQUITECTURA DO CNT. PODEMOS VERIFICAR QUE APESAR DO UTILIZADOR EFECTUAR PEDIDOS A DUAS APLICAÇÕES DIFERENTES, ESTAS UTILIZAM A MESMA BASE DE DADOS EM MS SQL SERVER. DENTRO DO CNT, PODEMOS TER TANTAS APLICAÇÕES QUANTAS QUISERMOS COM AS MESMAS FUNCIONALIDADES.

Desenvolvimento à Medida

Este sistema foi construído sobre um conjunto de requisitos que foram definidos à medida de uma determinada organização. Tal aconteceu não só por não existir no mercado aplicações vocacionadas para a Gestão de Parques Tecnológicos, mas também pelo facto de a organização necessitar de consolidar os orçamentos de todas as suas sub-organizações, numa aplicação que integrasse a contabilidade orçamental e analítica com o controlo de gestão. Na secção “5.3 O Problema – Controlo de Gestão” analisamos de forma mais detalhada esta problemática.

⁹ Sistema de Gestão de Base de Dados da Microsoft.

Parque Tecnológico - Necessidades Especiais

Um Parque Tecnológico apresenta necessidades especiais no que concerne à contabilidade. Além de possuir várias sub-empresas para tratar das diferentes vertentes do negócio, um parque tecnológico possui várias empresas instaladas nos seus edifícios e como tal, pretende um sistema que possa conciliar as informações que provêm dessas organizações. Como já fora referido no Capítulo 2, o desenvolvimento de aplicações com o Genio é iterativo. Nesta solução, o cliente foi bastante participativo no desenvolvimento da solução uma vez que ela foi à medida. A implementação de certas funcionalidades específicas tais como o suporte do multi-organismo interno e externo, por exemplo, necessitava de especificações concretas e estratégicas. As alterações que foram efectuadas no planeamento inicial das especificações ditaram alterações no plano de desenvolvimento do Projecto, o que condicionou a resposta a outros requisitos, menos prioritários e uma consequente alteração da calendarização inicial.

3.2 FUNCIONALIDADES DO SISTEMA DE GESTÃO FINANCEIRA

Neste Capítulo, vamos analisar de forma sumária as diferentes funcionalidades do Sistema de Gestão Financeira. As funcionalidades base dum Sistema de Gestão Financeira consistem nas funcionalidades que permitam processar, de forma automatizada, todos os documentos contabilísticos tais como facturas, recibos, etc.. Como tal, a esmagadora maioria dos sistemas de gestão financeira possuem esse tipo de funcionalidades e por conseguinte, relatá-los não constitui propriamente um verdadeiro acrescento de valor. O CNT possui um leque de funcionalidades bastante avançadas sobre as suas funcionalidades base e são essas que definimos em seguida na Tabela 1. Nesta tabela referem-se alguns conceitos mais técnicos de contabilidade. Podem ser encontradas definições para a maior parte destes conceitos nos Capítulos 1 e 2 de [cai03] e também nas secções Anexos M) e 5.3.2 Controlo de Gestão – Conceitos Básicos deste relatório.

Núm.	Funcionalidade	Descrição
1	Divisão Por Ano	A Base de Dados pode conter dados de vários anos, contudo, cada registo possui associado o ano respectivo.
2	MultiOrganismo	A Base de Dados pode conter registos provenientes de várias entidades/empresas: o utilizador só tem acesso à empresa à qual está associado.
3	Diários	Todos os movimentos contabilísticos estão associados a um Diário ¹⁰ . Os movimentos contabilísticos são sequenciais por diário. Suporte para numeração anual e mensal.
4	Numeração def. Movimentos Contabilísticos	A numeração atribuída a um movimento contabilístico é inalterável, independentemente de introduzirmos movimentos com datas anteriores ou posteriores.
5	Definição Contabilística e Orçamental do Artigo	No artigo encontra-se informação respectiva à classificação contabilística e orçamental, para que o utilizador não tenha de possuir conhecimentos funcionais de contabilidade para utilizar a aplicação.
6	Afectação à Contabilidade Analítica	Um valor lançado na contabilidade patrimonial pode ser redistribuído por vários centros de custo/responsabilidade.
7	Chave de Imputação de Centros de Responsabilidade	Redistribuição automática das afectações à contabilidade analítica.
8	Níveis de Acesso	Cada nível de acesso pode apenas consultar ou alterar dados com autorização
9	Estrutura em Árvore de Contas POC, Rubricas Orçamentais e Centros de Custo	Estrutura em árvore de contas POC, rubricas orçamentais e centros de custo permite criação de dependências entre os registos sem quaisquer limitações.
10	Multiorçamentos	Podem ser criados vários orçamentos.
11	Agregação de Orçamentos	Agregação de 2 ou mais orçamentos.
12	Orçamentos	Orçamento com dotação mensal. Com automatismo de redistribuição automática de dotação anual para 12 meses incluindo afectações nos centros de custos.
13	Imputação Automática de Orçamento	Permite inserir uma afectação ao centro de responsabilidade para todas as dotações mensais de um orçamento.
14	Consulta de Orçamento	Consulta de orçamento linear, que permite consultar linhas de orçamento com afectações a centros de responsabilidade e corrigi-las.
15	Ajustamentos Horizontais do Orçamento	Permite alterações horizontais no orçamento, sem mudar valor da rubrica (somas horizontais no orçamento, sem alterar o valor da rubrica).
16	Ajustamentos Verticais do Orçamento	Permite alterações verticais no orçamento, mudando o valor da rubrica (soma de alterações entre rubricas pode ser diferente de 0, mudando o valor final do orçamento).

¹⁰ Tipo de registo contabilístico que contém todos os movimentos de um determinado dia;

17	Listagens de Orçamento	Listagens de e por orçamentos, actividades e centros de custo.
18	Balancetes Orçamentais	Por orçamento, actividade e centro de responsabilidade (Controlo de Gestão)
19	Mapas de Execução de Orçamento	Semelhante ao anterior (Controlo de Gestão).
20	Classificadores	Permite associar rubricas orçamentais ou contas POC com relação n para n, permitindo a maior flexibilidade possível.
21	Lançamentos Contabilísticos Automatizados	Limitar as necessidades de conhecimento funcional de contabilidade aos mínimos.
22	Lançamentos Manuais	Permite lançamentos manuais.
23	Consulta de Movimentos	Várias vertentes: extractos de conta(s), lançamentos de uma conta, lançamentos de um tipo.
24	Operações Fim de Exercício	Operações de Fim de Exercício automatizadas.
25	Apuramento de IVA automatizado	Apuramento de IVA automatizado, apuramento mensal ou trimestral..
26	Mapas de Controlo de IVA	Declaração anual, controlo de IVA suportado, dedutível, liquidado, mapas recapitulativos, etc..
27	Regime Normal ou Pró-Rata	Opção de regime de IVA pró-rata definido por organismo.
28	Regime Misto de Pró-Rata	Regime Misto de Pró-Rata – podemos definir que algumas facturas utilizam Pró-Rata e outras não
29	Apuramento de Pró-Rata	Apuramento de Pró-Rata definitivo e automatizado.
30	Mapas Oficiais	Balanço, Balancete por Contas, Balancete Analítico, Demonstração de Resultados, Fluxo de Caixa, etc..
31	Ficheiros SAFT	Geração dos Ficheiros SAFT.
32	Importação de Encomendas	Importação de Encomendas feitas pelos clientes assim como encomendas feitas aos fornecedores.
33	Retenções na Fonte	Empresa retém valores que entrega ao estado substituindo o terceiro nessa obrigação.
34	Documentos por Classificar	Listagens de adjudicações, processamentos – facturas, antecipações de receitas, receitas e artigos que não contem toda a informação necessária para que os movimentos contabilísticos sejam efectuados.
35	Classificação da Adjudicação	Classificação Orçamental da Adjudicação automatizada baseada na informação associada ao artigo através do classificador de despesa/receita. Pode também ser efectuada manualmente.
36	Multiprocessamento	Automatiza a Classificação Orçamental da adjudicação baseando na informação incluída no artigo e data de compromisso.
37	Notas de Crédito	Automatização lançamentos contabilísticos.
38	Fundo de Maneio	Automatismo para gestão de fundos de maneio.
39	Processo Simplificado	Automatismo para processar as facturas não adjudicadas.

40	Importação de Vencimentos, Amortizações e Bens	Acesso à informação sobre Vencimentos , Amortizações e Bens em Sistemas Externos.
41	Documento de Despesa	Previsão de transacções com diferentes regimes de IVA, tipos de IVA, assim como transacções nacionais, inter comunitários e outros.
42	Registo de Receita Automático	Automatismo que abrange venda a dinheiro assim como a receita sem antecipação prévia (no segundo caso o recebimento não está criado automaticamente).
43	Notas de Crédito (aos clientes)	Automatização dos lançamentos contabilísticos.
44	Facturação	Emissão de Facturas, Venda a Dinheiro, etc.
45	Facturas pró-forma	Emissão de Facturas Pró-forma
46	Recibos	Emissão de Recibos
47	Controlo de Numeração de Documentos	Listagens que verificaram possíveis discrepâncias na numeração de documentos

TABELA 1 – FUNCIONALIDADES AVANÇADAS DO CNT

Este conjunto de funcionalidades assegura que o CNT pode-se equiparar com alguma margem em relação aos seus competidores de mercado, como podemos verificar através da análise efectuada na secção M) de Anexos. Estas funcionalidades cumprem na integridade os requisitos definidos pela empresa para o produto.

3.3 MOTOR DE CLASSIFICAÇÃO AUTOMÁTICA DE DOCUMENTOS

Neste capítulo vamos analisar de forma específica os desenvolvimentos realizados para desenvolver esta funcionalidade que, apesar de integrar a solução base, foi definida como um requisito essencial.

Como já fora referido na secção Capítulo 1, esta funcionalidade apresenta uma nomenclatura comercial que, no nosso entender, pode induzir em erro. Esta funcionalidade traduz-se num conjunto de desenvolvimentos para que sejam implementados automatismos que permitam que o utilizador não tenha que possuir conhecimentos funcionais para que possa utilizar a aplicação. Em concreto, não ter de conhecer o Plano Oficial de Contas ou a presente estrutura de Rubricas Orçamentais para que sejam efectuados os respectivos lançamentos aquando o Processamento dum documento contabilístico. Contudo, esta automatização assenta numa associação prévia dos artigos/produtos à contabilidade

orçamental e analítica efectuada por um utilizador com conhecimento funcional de contabilidade (vulgo, contabilista).

Para que esta associação seja efectuada, foi criado um classificador de despesa e de receita. Estes classificadores de despesa e de receita permitem associar cada artigo a uma determinada conta POC, rubrica orçamental e um tipo de IVA (valor, mercado, etc.). Esta associação única faz com que seja possível efectuar os lançamentos contabilísticos de forma automática. Este classificador funciona com base numa área que possui um registo por cada artigo classificado. A classificação pode ser efectuada para os documentos de receita ou de despesa de forma distinta e isolada. Os lançamentos são efectuados com base nesta informação e foram implementados através do desenvolvimento de rotinas manuais para o efeito. Estes desenvolvimentos implementam as funcionalidades 20, 21 e 39 listadas na Tabela 4 da secção anterior. Apesar do utilizador típico do sistema dispensar o conhecimento funcional de contabilidade para o utilizar, esta funcionalidade necessita que a classificação dos artigos seja efectuada à priori por um contabilista que conheça a realidade da empresa.

3.4 CONCLUSÕES E TRABALHO FUTURO

Neste capítulo verificámos que o CNT assentou o seu desenvolvimento numa Base de Dados em SQL Server e em código fonte gerado a partir do Génio. As funcionalidades mais complexas foram implementadas recorrendo a rotinas manuais.

Este sistema de gestão financeira possui várias funcionalidades avançadas que visam automatizar os processos mais redundantes e baixar os níveis de exigência de conhecimento funcional por parte do utilizador. A implementação do Motor de Classificação de Documentos Contabilísticos possui uma importância particular neste assunto pois consegue, duma forma simples, automatizar os lançamentos contabilísticos que devem ser efectuados num processamento de um determinado tipo de documento.

Para o futuro seria interessante, no nosso entender, implementar duas funcionalidades em específico:

- Débitos directos, isto é, a possibilidade de movimentar as contas bancárias de forma integrada na aplicação;
- Automatização plena dos lançamentos contabilísticos e da classificação de documentos;

Os débitos directos são importantes na medida em que são realizadas diariamente várias operações bancárias e o facto de estas poderem ser efectuadas de forma integrada na aplicação tem ganhos evidentes. A automatização plena dos lançamentos contabilísticos visa a identificação de padrões nos lançamentos contabilísticos e de classificação de documentos com o intuito de não ser necessária a intervenção do contabilista para que esta se processe automaticamente.

CAPÍTULO 4 - INTEGRAÇÃO DO SISTEMA DE GESTÃO FINANCEIRA COM BALANCED SCORECARD

Neste capítulo vamos começar por apresentar a metodologia Balanced ScoreCard, para que depois possamos analisar a solução BSC Quidgest e o sistema de indicadores financeiros que nos permite integrar esta última com o nosso Sistema de Gestão Financeira.

4.1 METODOLOGIA BALANCED SCORECARD

“The Balanced Scorecard (BSC) translates an organization’s mission and strategy into a comprehensive set of performance measures that provides the framework for a strategic measurement and management system.” [kap96]

Como já fora referido em capítulos anteriores, o Balanced ScoreCard é uma metodologia de Gestão Estratégica global aplicada às organizações. Baseia-se em indicadores específicos e grandezas mensuráveis para concluir de que forma os objectivos e metas definidas inicialmente estão a ser atingidas e, consequentemente, os resultados efectivos da estratégia da organização.

O Balanced ScoreCard ajuda a gestão de topo a obter uma visão mais compreensiva e extensível do negócio pois foca os seus indicadores não só nos inputs financeiros, mas também noutros tais como os operacionais e de marketing. Esta visão mais global apoia o processo de tomada de decisão a longo prazo. As organizações são assim direccionadas a medir, além dos tradicionais dados financeiros, o que influenciou a concretização desses dados, como, por exemplo, o desempenho dos processos, a quota e capacidade de penetração de mercado, a aprendizagem a longo-prazo e o desenvolvimento das competências existentes, etc..

“The final management process embeds the Balanced Scorecard in a strategic learning framework. We consider this process to be the most innovative and most important aspect of the entire scorecard management process.” [kap96]

Apesar de possuir várias metas intangíveis, o BSC possui as metas estabelecidas pela gestão financeira, possibilitando que os gestores analisem mensal e/ou trimestralmente os resultados financeiros. Porém, os gestores conseguem também examinar criteriosamente a forma como a unidade de negócio está a atingir os seus objectivos em relação a clientes, processos internos/ inovação e aos funcionários, sistemas e procedimentos (*Perspectivas*), assim como as suas expectativas para o futuro. Esta inovação faz com que o Balance ScoreCard seja, mais do que uma mera ferramenta de medição de indicadores, um controlador de aprendizagem que o torna ímpar em relação às outras metodologias estabelecidas no mercado.

“Financial performance measures indicate whether a company’s strategy, implementation, and execution are contributing to bottom-line improvement.” [kap96]

Devido ao carácter deste projecto (Sistema de Gestão Financeira), vamos focar-nos mais nos aspectos financeiros do Balanced Scorecard. Apesar de dar suporte a uma série de diferentes tipos de indicadores, os financeiros também são comuns e bastante utilizados para compreender a realidade das organizações. Os objectivos associados à Gestão Financeira estão normalmente relacionados com medidores de ganhos/lucros como os Resultados Operacionais, ROCE (*Return-On-Capital-Employed*) ou o Volume de Negócio. Alternativamente, os objectivos financeiros poderão ser, por exemplo, um crescimento rápido das vendas ou a geração de *cash-flow*.

4.1.1 MAPAS ESTRATÉGICOS

“The Balanced Scorecard strategy map provides a framework to illustrate how strategy links intangible assets to value-creating processes.”, [kap04]

Para a implementação desta metodologia, temos de ser capazes de definir um mapa estratégico. Segundo o livro de Mapas Estratégicos de BSC acima referido, um mapa estratégico possui uma framework que possui, através dum modo gráfico mais intuitivo, caracterizar numa determinada estratégia a ligação que existe entre os objectivos mais concretos e mensuráveis e aqueles que são conceptuais e intangíveis. Os mapas estratégicos estão normalmente divididos em quatro perspectivas: Financeira, Clientes, Processos Internos e Aprendizagem e Crescimento.

A perspectiva Financeira descreve os outputs tangíveis da estratégia em termos financeiros tradicionais. Indicadores como o ROI (*Return On Investment*), Resultados Líquidos, Margem de Lucro, Resultados Operacionais, etc. mostram-nos qual o grau de sucesso da estratégia da organização. A perspectiva dos Clientes define o que é valorizado por parte dos nossos clientes-alvo. Essa valorização é traduzida no contexto para que os recursos intangíveis criem valor. Se os clientes valorizam a qualidade do produto e o tempo nas entregas e nas produções, por exemplo, então as competências, os sistemas e os processos que produzem e entregam produtos e serviços de qualidade devem ser valorizados pela organização. Se os clientes valorizam inovação e desempenho elevados então temos de garantir a valorização das competências, sistemas e processos que criam novos produtos e serviços de qualidade.

“The financial and customer perspectives describe the desired outcomes from the strategy. Both perspectives describe the desired outcomes from the strategy. (...) The internal process perspective indentifies the critical few processes that are expected to have the greatest impact on the strategy. (...) The learning and growth perspective indentifies the intangible assets that are most important to the strategy.” [kap04]

As duas perspectivas acima referidas limitam-se a descrever o que se pretende atingir com a nossa estratégia mas não medem as formas objectivas de atingir essas metas. Como pode então a organização atingir os objectivos desejados?

As perspectivas de Processos Internos e a de Aprendizagem e Crescimento dão-nos estes indicadores. A perspectiva de Processos Internos identifica um conjunto reduzido de processos que tem um grande impacto na estratégia global. Por exemplo, uma organização pode aumentar o seu investimento em investigação e desenvolvimento para reinventar os processos de desenvolvimento do(s) seu(s) produto(s) para que possam ser oferecidos produtos de alto desempenho e de grande inovação aos seus clientes.

A perspectiva de Aprendizagem e Crescimento identifica os recursos intangíveis que são mais importantes para a sua estratégia. Os objectivos nesta perspectiva identificam que posições na estrutura de recursos humanos da empresa (*Capital Humano*), que empresas

(*Capital de Informação*) e que tipo de clima organizacional (*Capital Organizacional*) são necessários para suportar os Processos Internos de criação de Valor.

É da interligação entre estas quatro perspectivas que obtemos a chave para a criação de Valor e, por conseguinte, uma Estratégia objectiva e consistente.

4.2 FERRAMENTA PROPRIETÁRIA “BSC QUIDGEST”

O Balanced ScoreCard Quidgest é uma ferramenta proprietária que visa a implementação da metodologia Balanced Scorecard. Este sistema de informação criado, à imagem do CNT (Sistema de Gestão Financeira Empresarial), com recurso à ferramenta de geração automática de código Génio, permite-nos essencialmente gerar os Scorecards com recurso à informação existente num determinado ERP (*Enterprise Resource Planning*).

Segundo o Manual do Utilizador do BSC Quidgest, o alinhamento estratégico dos processos numa organização a partir desta aplicação está baseado na definição dum Mapa Estratégico, Perspectivas, Objectivos Estratégicos e Indicadores. Os Indicadores são a base da pirâmide exemplificada na Figura 11. Os Balanced ScoreCards são calculados de baixo para cima, tendo em conta os valores dos indicadores.

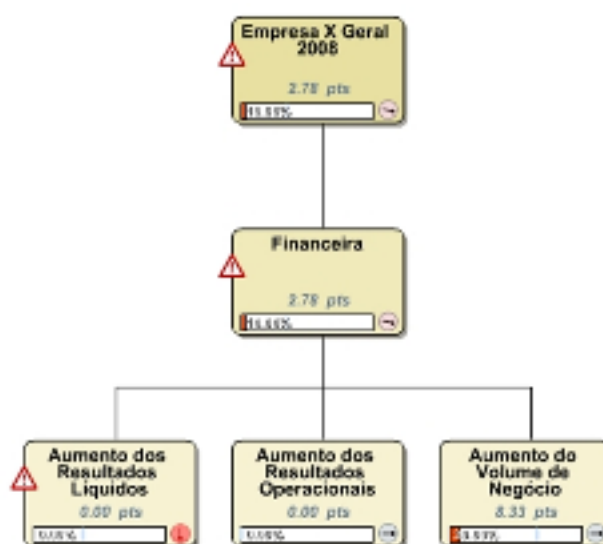


FIGURA 11 – SCORECARD EM ÁRVORE GERADO A PARTIR DO BALANCED SCORECARD QUIDGEST.

4.2.1 DEFINIÇÃO DOS MAPAS ESTRATÉGICOS

No BSC Quidgest, tudo começa com a definição dos Mapas Estratégicos. Este(s) mapa(s) representam a estratégia da organização. Cada mapa surge organizado por perspectivas. Estas perspectivas relacionam-se com diferentes áreas de gestão da organização e são inteiramente personalizáveis. Contudo, seguindo a metodologia BSC de base, temos as seguintes 4 perspectivas principais: Financeira, Clientes, Interna e de Aprendizagem ou Recursos. Cada perspectiva vai possuir objectivos diferentes com diferentes indicadores. Os indicadores são normalmente uma definição de algo que pode ser traduzido em números como o número de inquéritos satisfatórios ou a percentagem de clientes que não apresentaram reclamação nos estabelecimentos e/ou postos de venda.

4.2.2 DEFINIÇÃO DE INDICADORES

O indicador é um factor no qual são introduzidos dados mensuráveis que pretendem avaliar, de forma objectiva, uma realidade concreta. Estas definições são vitais para a definição do alcance dos objectivos e, consequentemente, da estratégia da Organização.

Na definição destes indicadores, podemos declarar um conjunto de características:

- Periodicidade: período de tempo em que os indicadores têm de ser reavaliados;
- Polaridade: se é melhor para a gestão estratégica (mais pontos) que aquele indicador possua um valor mais alto ou mais baixo;
- Unidade: medida que vai ser utilizada pelo indicador;
- Ponderação: do valor do indicador em relação aos outros indicadores;
- Acumulação: se o valor deve ser tomado em conta como absoluto ou como uma soma dele com as leituras anteriores no período definido;

4.2.3 GERAÇÃO DE SCORECARDS

Um Scorecard é um gráfico que nos permite, em relação a um determinado mapa estratégico, apurar se a estratégia está a sortir o efeito desejado na organização ou se não está. Este tem embebido um sistema de pontuação que mede a pontuação de cada objectivo em relação às metas propostas para cada um dos seus indicadores. Isto faz com que esta

geração se processe em árvore, começando por definir o valor das folhas (indicadores) até ao objectivo maior, como por exemplo, ser líder de mercado na península ibérica ou dominar um determinado nicho deste.

Através da análise gráfica é possível não só apurar se a estratégia está a sortir efeito bem como qual ou quais os indicadores que mais (ou menos) estão a contribuir para tal.

A integração de objectos em Flash com o Génio permite-nos utilizar vários componentes multimédia para a demonstração dos nossos gráficos Scorecard. Existem, ao todo, 4 vistas diferentes para os Scorecards: Árvore (a mais comum), Lista (Figura 13), Dashboard (Figura 12) e Histograma.

A visão em Árvore, que já vimos acima, permite-nos ter uma visão global dos objectivos e das metas definidas, assim como a percentagem de sucesso ao alcançá-los. A vista em lista é muito semelhante ao nível dos dados que apresenta. As vistas em forma de dashboard e histograma são mais centradas nos indicadores. A visão em Dashboard acaba por transmitir mais em detalhe qual o estado de cumprimento das metas definidas para cada um dos indicadores. Em contrapartida, o Histograma dá-nos uma perspectiva mais cronológica do estado do indicador ao longo do tempo, mantendo também as metas definidas para ele nos diferentes períodos.



FIGURA 12 - EXEMPLO DUM SCORECARD EM DASHBOARD. AQUI É POSSÍVEL VERIFICAR O ESTADO ACTUAL DOS INDICADORES DO MAPA ESTRATÉGICO ESCOLHIDO. O PONTEIRO AZUL É A META, SENDO QUE AS DIFERENTES CORES NO GRÁFICO TÊM O SIGNIFICADO DO GRAU DE CUMPRIMENTO DAS METAS.



FIGURA 13 – NOVO EXEMPLO DUM SCORECARD, EM VISTA DE LISTA.

Esta aplicação acaba por implementar uma metodologia bastante útil e utilizada no mercado pelos gestores, devido à riqueza de informação que oferece. Esta pode funcionar isoladamente ou integrada com uma ou várias aplicações. Os dados podem ser importados manual ou automaticamente, sendo que muitos deles fazem mais sentido serem processados de uma das maneiras referidas anteriormente.

No final, tudo depende dos Objectivos que a organização define para si, da Estratégia que esta pretende ver implementada e da avaliação que efectuar dos seus efeitos.

4.3 INTEGRAÇÃO CNT COM BSC

Este capítulo pretende descrever a solução encontrada para o problema que nos foi proposto – construção dum sistema de indicadores financeiros com vista à integração da ferramenta proprietária Balanced Scorecard Quidgest com o CNT. O CNT é o módulo de Gestão Financeira já descrito no Capítulo 3. A sua integração com o Balanced Scorecard surge no intuito de cruzar de forma automatizada e mais intuitiva os indicadores financeiros que dele resultam com outros dados da organização (ex.: ERP). Contudo e no contexto deste projecto irá ser abordada apenas a vertente financeira do Balanced Scorecard.

4.3.1 SISTEMA DE INDICADORES FINANCEIROS

Para cada organização faz sentido definir diferentes objectivos, em face do seu crescimento passado, do estado do mercado e da economia, da localização, do tipo de negócio, do capital investido, etc.. Diferentes objectivos implicam diferentes indicadores. Deste modo,

torna-se difícil senão impossível definir um conjunto de indicadores genéricos que interessem desta forma a todas as organizações. Assim, o Balanced Scorecard utiliza também o *know-how* das organizações na definição dos seus objectivos, metas, indicadores e iniciativas. Numa perspectiva estritamente financeira, os rácios financeiros são os mais utilizados para indicar os índices de crescimento numa organização. Mas esta escolha está estritamente dependente da estratégia da organização que pode ter uma forte componente da perspectiva Financeira, uma componente mediana ou nula. Mesmo que esta componente seja levada em conta de forma significativa, os indicadores e a sua polaridade não possuem valores predefinidos. Se, por exemplo, tivermos uma empresa que deseja fazer um *downsizing* através da diminuição dos Recursos Humanos na Área de Pós-Venda, poderá ser interessante que um indicador como o volume de vendas não aumente muito ou até se mantenha apenas constante.

Tendo em conta estes factores, assumimos esta definição do sistema de indicadores como uma Prova de Conceito, em que se pretende provar que é possível fazer essa integração e que estes indicadores, na perspectiva financeira, serão interessantes para uma boa parte das organizações.

Normalmente, os *ratios* financeiros mais utilizados são os *ratios* que averiguam o nível de lucro (*profitability*) da organização. Alguns dos mais comuns são os seguintes:

- Resultados Brutos;
- Resultados Operacionais;
- Margem de Lucro;
- Resultados Líquidos;
- EBIDTA;
- Volume de Negócio;
- Retorno do Investimento (ROI);
- Retorno de Recursos (ROA);
- Retorno do Capital Empregado (ROCE);

A maior parte destes indicadores podem ser determinados face a um período de tempo variável, como um trimestre, um semestre ou um ano.

Após uma análise do CNT, entendeu-se que por uma questão de tempo não vão ser implementados indicadores de BSC para todos os indicadores acima referidos. Embora

todos possuam informação diferente sobre a Gestão Financeira da organização, possuem algumas bastante redundantes e comuns a praticamente todos como o valor das Vendas, por exemplo. Contudo e como prova de conceito de que é possível e interessante integrar o BSC Quidgest com o módulo de Gestão Financeira, apresentamos três indicadores para demonstrar isto mesmo: os Resultados Líquidos, os Resultados Operacionais e o Volume de Negócios. Podemos obter mais informações sobre estes indicadores na secção L) em Anexos.

Através da escolha destes três indicadores vamos poder compreender não só a evolução da organização de forma absoluta bem como por comparação com os dados com os de outras organizações (do grupo ou não). A inclusão de indicadores que, como estes, são mais complexos permite recolher informação respeitante a vários aspectos da organização. Caso o CNT se encontre integrado num ERP, o BSC possuirá outros tipos de dados que permitirão retirar ilações concretas das razões pelas quais os indicadores financeiros variam desta ou daquela forma (correlação entre indicadores Financeiros e Não-Financeiros).

4.3.2 SISTEMA DE INTEGRAÇÃO BSC COM OUTRAS APLICAÇÕES

A ferramenta proprietária “Balanced Scorecard Quidgest” possui, na sua versão actual, opções de apoio à integração do BSC com ERPs. Tal prende-se com a importação de dados para alimentar os indicadores previamente definidos. Apesar de autónomo, o BSC Quidgest necessita dos dados presentes nas diversas componentes do negócio para alimentar os indicadores e assim compreender o quão bem sucedida está a ser a estratégia da organização. Assim sendo, está presente na ferramenta proprietária uma funcionalidade denominada de “Integração” que permite ao utilizador definir a forma como o BSC importa dados de outros sistemas. Este sistema não faz qualquer tipo de comunicação entre sistemas mas sim uma importação completamente manual dos dados da base de dados doutros sistemas.

Através da análise da Figura 14, podemos compreender a Arquitectura base da integração do BSC com outras aplicações. As aplicações (neste contexto vamos referir como ERPs uma vez que são o tipo de aplicações - ou de conjunto de aplicações – cuja integração com o BSC oferece uma perspectiva mais completa) possuem uma ou mais bases de dados associadas, assim como podem ser de diferentes tipos.

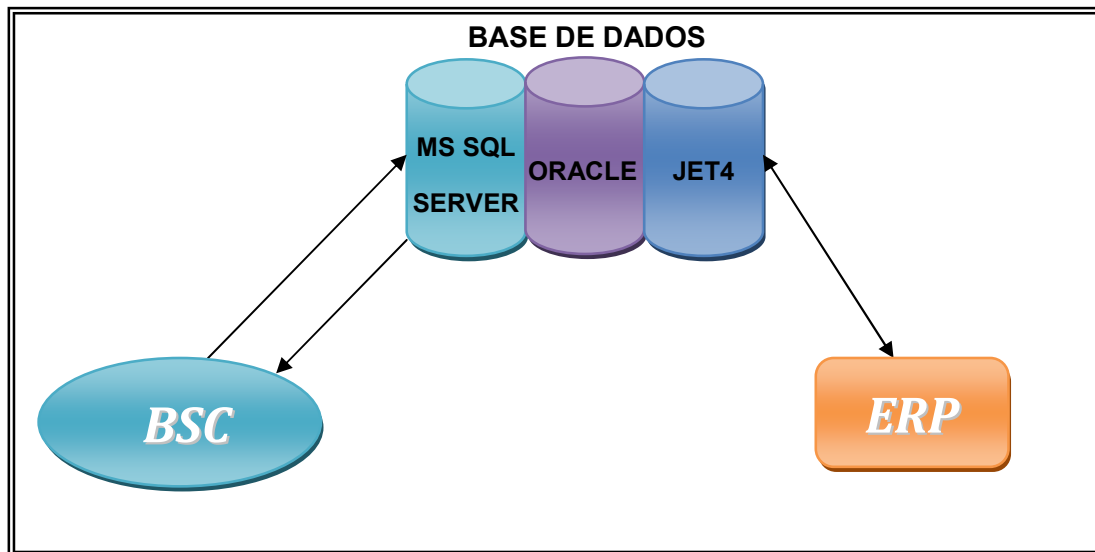


FIGURA 14 – ARQUITECTURA DA INTEGRAÇÃO ENTRE O BSC E OUTRAS APLICAÇÕES.

Esta integração apenas suporta os Sistemas de Gestão de Bases de Dados *Microsoft SQL Server*, *Oracle* e *Jet4*. Estas limitações devem-se a opções estratégicas da Quidgest em restringir os seus desenvolvimentos a Sistemas de Gestão de Bases de Dados utilizados pelos seus clientes.

Na Figura 15, podemos verificar uma integração já realizada para obter dados para um indicador. Temos as *queries* definidas do lado esquerdo, com o nome dos valores que retornam. Podemos também verificar também a data associada aos dados e um campo denominado “Fórmula de Integração”. Este campo não é mais que uma fórmula baseada nos valores retornados pelas consultas.

FIGURA 15 - SEPARADOR DO INTEGRADOR NO FORMULÁRIO DE CONSULTA/EDIÇÃO NUM INDICADOR.

As consultas presentes nas secções A), B) e C) de “Anexos” retornam sempre uma única célula contendo um único valor. Todas as *queries* de integração têm de ter este formato. O seu conteúdo é completamente parametrizável desde que aquilo que retornam seja um valor definido. As *queries* introduzidas são completamente independentes de qualquer outra parametrização efectuada no BSC ou na aplicação/servidor (se definirmos a arquitectura da integração do BSC com o ERP como cliente-servidor, temos que o BSC será o Cliente uma vez que é quem pede os dados e o servidor que contém dados que a aplicação utiliza, o Servidor), sendo da responsabilidade do utilizador garantir a lógica de negócio neste tipo de utilizações.

Esta funcionalidade está configurada para ser utilizada por administradores de sistemas. O utilizador mais comum apenas terá de definir as datas das importações de dados e importar. Este modelo torna-se particularmente eficaz quando os indicadores possuem uma periodicidade do tipo acumulado, isto é, as importações de dados são efectuadas num período definido (ex.: mensal) e os dados a ser mostrados tem um significado mais lógico quando acumulados com os dos restantes períodos (ex.: vendas). Tendo em conta o Sistema de Indicadores Financeiros desenvolvido e previamente descrito, optou-se pelo desenvolvimento de três *queries* independentes que importavam os dados relativos ao ano 2008. Este código resume-se à consulta da tabela LMOVI do CNT. Através do acesso às

diversas linhas de movimento, é possível obter-se os movimentos de débitos e créditos lançados nas diferentes contas POC. O saldo entre estes débitos e créditos (soma de todos os presentes no período e a diferença entre estas) define os valores a considerar em relação às contas POC que devemos utilizar no cálculo destes indicadores.

A utilidade desta funcionalidade de integração é evidente pois automatiza um processo que doutra forma levaria imenso tempo a concretizar manualmente. Contudo, esta apresenta algumas deficiências. Uma delas é, desde logo, o facto de não conseguirmos associar à mesma *query* períodos de tempo como parâmetros de entrada das *queries* e assim, associar à data (de importação) dos dados o período definido na *query*. O facto de termos as datas *hardcoded* nas *queries* oferece muito pouca flexibilidade ao utilizador para definir as suas importações. Seria interessante o sistema possuir uma forma de conter parâmetros de entrada específicos e outros previamente definidos, uns opcionais e outros obrigatórios, etc.. Outro aspecto menos positivo é o facto de não podermos utilizar valores de outros indicadores ou de outras *queries* nas fórmulas de integração. É muito comum verificar que alguns indicadores dependem doutros para o cálculo do seu valor, logo a importação de dados *interna* poderia assumir uma rentabilização dos recursos disponíveis. A melhoria destes dois aspectos é um trabalho a realizar futuramente no sentido de melhorar a solução já existente.

Concluimos assim que a tecnologia “BSC Quidgest” implementa a metodologia Balanced Scorecard de forma interessante e apelativa. Este é integrável com o Sistema de Gestão Financeira CNT, bem como qualquer outra aplicação que utilize bases de dados *Oracle*, *SQL Server* ou *Jet4*. Contudo, o BSC necessita de receber dados de diferentes aspectos da organização, principalmente os não-financeiros, para assim compreender a realidade estratégica global da mesma. A sua integração isolada com o CNT não maximiza o seu potencial. O Sistema de Indicadores a utilizar na integração depende totalmente da Estratégia que a organização queira implementar, sendo que as razões para definir o presente Sistema de Integração prende-se mais com a prova de conceito que foi aqui efectuada do que com indicadores duma Estratégia Global e Standard para as organizações que não é passível de ser concebida.

CAPÍTULO 5 - INTEGRAÇÃO DO GÉNIO COM RIA'S

Neste capítulo, vamos falar da integração de *Rich Internet Applications* (2.2 *RIA – Rich Internet Application*) com o Génio (secção M de “Anexos”). Como já verificámos em capítulos anteriores, o Génio pode gerar aplicações em BackOffice e em *web*. Naturalmente, as aplicações geradas para *web* têm toda a facilidade em incluir *RIA's* mas o problema abordado não é esse. Muitas aplicações são desenvolvidas em *BackOffice* por vários motivos, dois dos quais são o aumento da segurança e do desempenho em relação às aplicações *web*.

O contexto em que nos foi proposto a integração de *RIA's* no Génio foi o desenvolvimento de um Sistema de Gestão Financeira (CNT) em BackOffice e será a esta vertente que nos vamos referir ao longo deste capítulo. Em concreto, foi-nos proposto passar o módulo de Controlo de Gestão, inicialmente integrado com o CNT e cujos relatórios eram produzidos com o auxílio de mapas em Crystal Reports, para mapas construídos sobre Silverlight. Este capítulo vai abordar algumas das vantagens em possuir *RIA's* integradas com o BackOffice do Génio e a integração actual com o Flash, outra ferramenta que possibilita a criação de *RIA's*. Irão ser analisadas as possibilidades de integração, as soluções desenvolvidas para o Controlo de Gestão e para o BSC e uma proposta de integração do Controlo de Gestão do CNT com o Silverlight.

5.1 VANTAGENS DAS RIA'S NO GÉNIO

O desenvolvimento de aplicações em BackOffice utiliza presentemente MFC – *Microsoft Foundation Class*. MFC é uma livreria que utiliza partes da API de Windows e conecta-as com classes desenvolvidas em C++. Isto faz com que seja mais fácil aceder a controlos padrão do Windows como formulários, botões, *labels*, etc.. Esta integração facilita a criação de aplicações para Windows pois contém já vários *templates* para criação de janelas, formulários, menus, etc.. O Génio limita-se a utilizar estas classes para que estas possam conter funcionalidades padrão que sirvam não só como base para qualquer aplicação de backoffice em Windows como também para aplicações *Business-Oriented*.

A utilização destas classes permite desenvolver aplicações em muito menos tempo do que o que seria esperado, tendo em conta a sua complexidade. Existem também ganhos em termos de segurança, uma vez que o código desenvolvido contém já um grau bastante aceitável de fiabilidade. As desvantagens são: as limitações impostas pela utilização constante dessa tecnologia impedem a natural evolução do design e das soluções apresentadas. A interactividade existente e a inclusão de componentes de maior riqueza visual são objectivos de difícil alcance. É aqui que entram as *RIA's*.

As *RIA's* permitem ao utilizador ter uma experiência muito mais rica de interacção, desde a inclusão de *streaming* de áudio e vídeo, até à aceleração 3D em alguns casos, com aumento do desempenho, etc.. Enquanto as aplicações web tradicionais assentam sobre uma típica arquitectura de cliente-servidor, em que o servidor se encarrega de todo o processamento e armazenamento dos dados e em que o cliente limita-se a mostrar um conjunto estático de dados, como *HTML*, nas *RIA's* tal não acontece. As *RIA's* também processam os dados do lado do cliente, sendo que o servidor apenas tem que os armazenar. Isto faz com que o servidor não tenha de ser consultado em cada clique do utilizador, proporcionando uma melhor interactividade e desempenho.

As *RIA's* correm, normalmente, em *plug-ins* instalados do lado cliente sobre um *browser* de Internet. Contudo, a inclusão de componentes *web* em aplicações de *backoffice* possibilita a inclusão destas tecnologias, fazendo com que seja possível obter o melhor de dois mundos: o desempenho e estabilidade das aplicações *backoffice* e o *design* e a interactividade das aplicações *web*.

5.2 INTEGRAÇÃO ACTUAL – FLASH

Como podemos verificar na secção “2.6 Adobe Flash“, existe presentemente uma integração de uma tecnologia que implementa *RIA's* com o Génio. Essa tecnologia é o Flash.

O Flash já está presente à vários anos no mercado e tem provas dadas na sua robustez para desenvolver aplicações *business oriented*. Esta tecnologia permite-nos com alguma

simplicidade, desenvolver aplicações com uma boa apresentação gráfica, reprodução de ficheiros de media com vídeos e áudio e elevada interactividade.

Antes do aparecimento do Flash, as aplicações web eram desenvolvidas em HTML. Apesar do HTML ser uma linguagem com bastantes potencialidades para criar *websites* estáticos em que os objectos eram declarados em posições fixas, o desenvolvimento de *websites* foi requerendo uma experiência progressivamente mais completa e complexa. As animações eram conseguidas com tecnologias como *Java Applets* e animações GIF.

Apesar de realizarem animações, estas tecnologias ainda apresentavam muitas limitações no que diz respeito à interactividade com o utilizador (*Java Applets*) e mesmo à qualidade das suas apresentações (GIF só com 256 cores). Enquanto as *Java Applets* eram difíceis de criar e instáveis, podendo facilmente consumir uma elevada percentagem dos recursos existentes para apresentar aplicações bastante simples, o Flash revela-se uma aplicação bastante leve e de fácil implementação. Estas características vieram suprir as falhas criando um novo patamar na Riqueza das aplicações *web*: as *RIA*'s.

“The core component of Flash Lite 3 is the rendering engine, which composites visual elements for display on the screen. This is supplemented by the ActionScript™ component, which processes events such as key presses, enabling dynamic interactivity in the application.” [ado08]

Como podemos verificar na Figura 16, o Flash possui uma arquitectura complexa que lhe permite tratar os diversos eventos e as diferentes renderizações que este necessita fazer ao longo das suas apresentações. A linguagem base do Flash é o ActionScript que nos permite fazer o *handling* dos diversos eventos. Depois de efectuar o desenvolvimento das apresentações flash, estes são compilados em ficheiros do tipo SWF (ShockWave Flash) e tipicamente embebidos no código fonte duma página HTML (Figura 17).

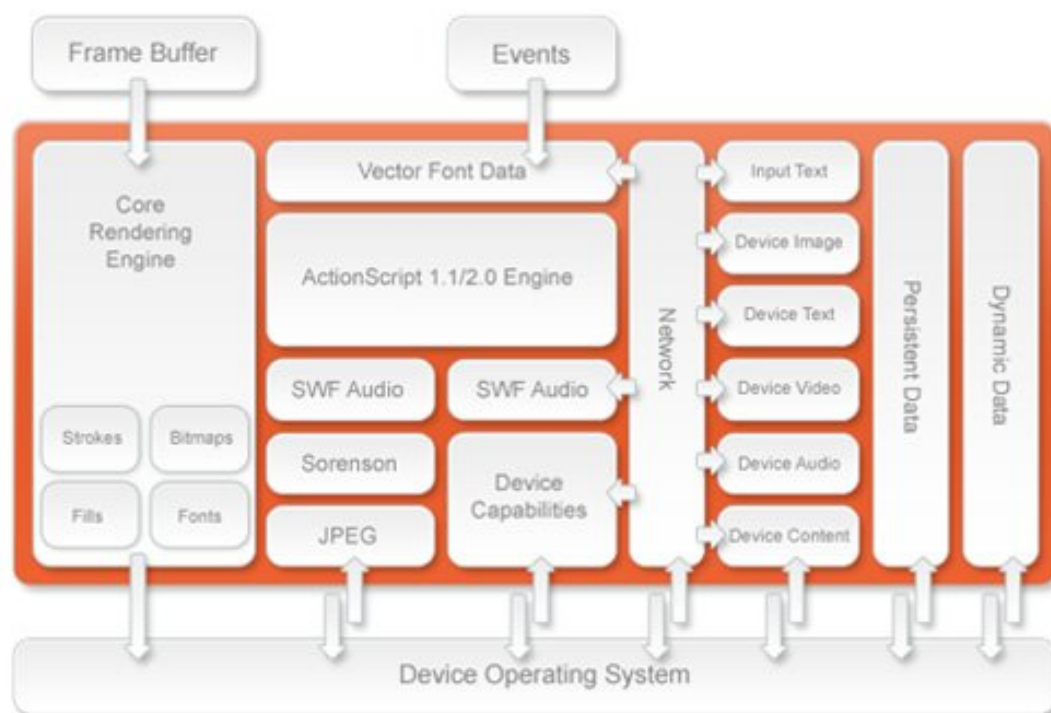


FIGURA 16 – ARQUITECTURA DO FLASH [AD008].

```
<object width="550" height="400">
  <param name="movie" value="somefilename.swf">
  <embed src="somefilename.swf" width="550" height="400">
</embed>
</object>
```

FIGURA 17 – CÓDIGO-FONTE HTML TÍPICO PARA INTEGRAR UMA APLICAÇÃO FLASH NUMA PÁGINA WEB. [W3S08]

5.2.1 FLASH – VISÃO GERAL

O Flash é uma plataforma que nos permite desenvolver *RIA*'s de forma simples. Corre num plug-in disponível para a maior parte dos *browsers* distribuídos ou num *player* isolado que existe para quase todos os sistemas operativos actualmente (Windows 9x+, Linux, MacOS, etc..). É programável com recurso ao ActionScript, uma linguagem de programação orientada a objectos que é executada numa máquina virtual: AVM – “ActionScript Virtual

Machine”. O AVM é uma plataforma que é incluída somente pelo Flash *player*. Esta linguagem acaba por ter algumas semelhanças com o *JavaScript*, outra linguagem de *scripting* muito utilizada.

O Flash disponibiliza uma grande variedade de formatos multimédia. Pode ser utilizado para criar filmes de tamanho reduzido e que sejam rapidamente carregados devido à característica compacta das suas aplicações. Facilmente vê incorporados filmes e sons, tornando a sua utilização bastante simples e desejável. Podemos utilizar o Flash para criar, por exemplo, uma aplicação interactiva para aprendizagem de alguma ferramenta num CD-ROM ou criar um anúncio publicitário para uma página web.

Contudo, as suas características únicas tais como a possibilidade de correr localmente aplicações com uma grande riqueza gráfica e a alta fiabilidade que oferecia, consumindo um nível de recursos bastante baixos, levou as empresas a desenvolver componentes e aplicações destinadas ao negócio (*Business-Oriented applications*).

5.2.2 INTEGRAÇÕES EXISTENTES

Em continuação do que foi abordado no Capítulo 2, temos então que o Flash apresenta uma integração com o Génio. O Génio é uma plataforma de geração automática de código para linguagens padrão que tem diversas versões suportadas através duma manutenção contínua da aplicação e implementação de novas funcionalidades. A integração com o Flash é uma integração *hardcoded*¹¹, isto é, não é dinâmica. Existem um conjunto de componentes predefinidos que podem ser utilizados e integrados nas aplicações geradas pelo Génio.

Como os utilizamos?

Na criação do formulário, definimos um campo que poderá ter qualquer denominação. Esse campo terá de estar forçosamente ligado a uma área (ex.:tabela, ver secção 2.3.2 Conceitos Proprietários) que seja coerente com a área daquele formulário. Existe depois no Génio uma opção nas propriedades dos campos dos formulários denominada “Tipo de Ajuda”.

¹¹ - denominação que se atribui a parametrizações que estão embuídas no código fonte da aplicação. Ex.: A definição do ano corrente é hardcoded, sendo sempre idêntica à hora do sistema. Logo, não é possível alterar o ano actual numa opção da aplicação.

Para que esta combinação funcione correctamente, é necessário que o ficheiro compilado do objecto flash utilizado (.SWF) esteja sempre presente na directoria do executável. Não é possível utilizar outros ficheiros .SWF que não aqueles correspondentes aos controlos incluídos na lista.

Como os parametrizamos?

Após incluirmos o ficheiro .SWF no formulário, temos de conseguir comunicar com este. Como o objecto é *hardcoded*, a única possibilidade que temos para comunicar com ele é enviando-lhe mensagens com strings de texto de forma assíncrona. Estas strings de texto contêm informações sobre as propriedades dos objectos a posicionar na apresentação.

Ex.: Numa agenda, temos que colocar quais os eventos já marcados e introduzidos na base de dados. Enviamos então uma string com a lista de eventos que queremos colocar.

Este tipo de strings tem que possuir um formato predefinido que o objecto Flash compreenda. Estas strings possuem normalmente o formato XML. O XML (Extensible Markup Language) é uma maneira de estruturar informação num ficheiro de texto utilizando *tags* semelhantes ao HTML. Contudo, em vez de serem estáticas (como em HTML), elas são definidas consoante o tipo de estrutura de dados que nos interessa representar. Esta combinação aumenta a versatilidade da solução, uma vez que associando as capacidades de parametrização do XML ao motor de renderização presente no Flash, temos a possibilidade de parametrizar a apresentação tendo em conta qualquer fonte de dados, desde que saibamos o formato da estrutura de dados que o nosso objecto Flash espera.

Na Figura 18, podemos verificar a definição duma rotina manual de apoio ao Sistema de Gestão de *WorkFlows* da Quidgest. É possível verificar que temos uma leitura de base de dados que é convertida posteriormente numa string XML que contém as propriedades dos objectos a representar no *Workflow*.

```

/[MANUAL GGD FLASH WORKF2]/
    CString prm, objXml;

    // assinala a entrada no workflow
    wfm._inWorkflow = TRUE;
    wfm._wfDialog = this;

    // limpa o workflow
    wfm.LimpaWorkflow();

    // preenche o ID do workflow
    wfm._wfid = pDoc->workf->m_codworkf;

    // obtém as opções de representação das actividades
    wfm._showServico = (pDoc->workf->m_servico == 1) ? TRUE : FALSE;
    wfm._showEstado = (pDoc->workf->m_estado == 1) ? TRUE : FALSE;
    wfm._showFuncionario = (pDoc->workf->m_funciona == 1) ? TRUE : FALSE;
    wfm._showDuracao = (pDoc->workf->m_duracao == 1) ? TRUE : FALSE;

    // lê workflow da BD
    wfm.ReadFromBD();

    // cria as strings XML para o controlo
    prm = wfm.GetXML();
    objXml = wfm.GetObjects();

    ExecuteFlashCmd("LoadObjXml",objXml);

```

FIGURA 18 – CÓDIGO FONTE DUMA ROTINA MANUAL DO SISTEMA DE GESTÃO DE WORKFLOWS DA QUIDGEST.

Neste caso específico, estas são as actividades respeitantes ao Workflow em causa, mas poderiam ser os veículos em stock ou as reuniões marcadas entre duas datas, por exemplo. Finalmente temos a chamada à função (*ExecuteFlashCMD*) que se encarrega de enviar para o Flash a string XML construída. Esta funcionalidade já vem incorporada no Génio e é sempre necessário invocá-la quando queremos interagir com a aplicação. A aplicação conhece qual o componente flash que está a ser invocado através da *tag* respeitante à rotina manual. Por exemplo, */[MANUAL GGD FLASH WORKF2]/* diz-nos que esta é uma rotina manual da aplicação GGD em relação ao objecto Flash WORKF2. Cada instância de objecto Flash no Génio possui uma denominação própria que é definida num campo próprio no editor de formulários. O ActionScript do Flash processa o ficheiro XML como todos os outros eventos que acontecem em Flash (tais como um “*mouse over*”, por

exemplo). Para todos estes eventos, o Flash envia respostas à aplicação também com mensagens XML, criando assim um canal de comunicação assíncrono com esta.

Onde é utilizado?

Hoje em dia, existe já cerca de uma dúzia de componentes Flash desenvolvidos com as suas próprias funcionalidades e parametrizações. Este corre em várias aplicações da empresa com sucesso comercial, casos da Gestão de Workflows e Balanced ScoreCard. Um dos componentes / controlos que é também utilizado com bastante sucesso é a Agenda Electrónica, que nos permite agendar vários tipos de eventos, com algumas semelhanças ao conhecido produto *Google Calendar*.

Conclusão

Em suma, o programador apenas tem de conhecer o formato XML que o objecto Flash aceita para poder interagir livremente com ele. Não é possível, neste momento, integrar de forma genérica objectos Flash no Génio. O Flash é utilizado de forma estável e com sucesso em várias aplicações/produtos da empresa como o Balanced ScoreCard ou a Gestão de Workflows. Podemos ainda verificar outras considerações técnicas sobre a comunicação do Génio com Flash no Anexo K) em Anexos.

5.3 O PROBLEMA – CONTROLO DE GESTÃO

Neste capítulo, vamos ter uma descrição extensiva e pormenorizada do problema que pretendemos resolver. A inclusão do controlo no CNT (Sistema de Gestão Financeira) é algo que só por si distingue este produto dos seus mais directos concorrentes no mercado. Foi proposto resolver o problema com recurso a *RIA's* e a partir daí, retirar as conclusões sobre qual a melhor ferramenta a aplicar para dar resposta a este problema.

5.3.1 CONTROLO DE GESTÃO – O QUE É?

“O Controlo de Gestão tem assumido nos últimos anos, um papel progressivamente relevante na gestão empresarial, com destaque para a vertente da gestão, medição e monitorização da performance.” [rod09]

O Controlo de Gestão é, tal como o nome indica, a disciplina que estuda o impacto da Gestão Estratégica nas organizações. Através de indicadores de diversos sectores, medidas e características da organização, é possível verificar qual o impacto actual e futuro da estratégia da organização naquilo que vão ser os seus resultados operacionais. Podemos verificar simplesmente se o nível de facturação da empresa tem aumentado, se o orçamento tem sido o suficiente ou o adequado às actividades de determinado sector ou se, por outro lado, não é adequado ao valor realizado. Podemos ter visões diferentes dos resultados da organização, cruzando várias vezes os nossos dados em matrizes bi e tridimensionais, de forma a obtermos resultados mais concretos e próximos da realidade da organização.

“De um controlo de gestão eminentemente financeiro, evoluiu-se para um controlo de gestão estratégico, operacional e financeiro.” [rod09]

Como já verificámos no Capítulo 4, as noções de controlo de gestão têm evoluído ao longo dos tempos, tendo surgido metodologias (ex.: Balanced ScoreCard) que têm contradito a noção clássica de que o Controlo de Gestão se reporta a indicadores meramente financeiros. Hoje em dia, todos os aspectos da organização são importantes, desde, por exemplo, o número de projectos novos que obteve até ao índice de pontualidade dos seus colaboradores.

Na perspectiva desta aplicação (CNT), o Controlo de Gestão que irá ser realizado tem como base mapas que provém do cruzamento de diferentes grandezas funcionais do sistema. Estes mapas são construídos segundo os requisitos do cliente para a aplicação e é sobre estes que nos vamos referir daqui em diante. Por outro lado, também nos foi proposto o desenvolvimento da integração entre a ferramenta BSC Quidgest abordada no “Capítulo 4 - Integração do Sistema de Gestão Financeira Com Balanced Scorecard” e o CNT como forma complementar de Controlo de Gestão. A integração desta ferramenta com *RIA's* também irá ser abordada ao longo deste capítulo.

5.3.2 CONTROLO DE GESTÃO – CONCEITOS BÁSICOS

Para a melhor compreensão da problemática do Controlo de Gestão, deixam-se aqui explicações sumárias para alguns conceitos importantes para a sua realização. Este sub capítulo não pretende conferir noções de Contabilidade Geral ou Analítica mas sim explicar

de forma simples a maior parte dos conceitos utilizados para que este relatório possa ser compreendido mesmo por aqueles que nunca trabalharam com contabilidade. Estas noções têm por base as definições presentes em [cai03]

Dotações Mensais – Orçamentos lançados para uma determinada rubrica orçamental num determinado período/mês.

Actividades – Grandeza funcional que permite agrupar a contabilidade pela actividade da empresa tal como imobiliária, jogo, lazer ou restauração, por exemplo

Rúbricas Orçamentais – Grandeza funcional que se destina a agrupar de forma lógica os diferentes orçamentos da organização.

Centros de Responsabilidade – Vulgarmente conhecidos como centros de custo, são grandezas funcionais que possuem afectações associadas. Estes centros de custo estão normalmente associados a outras grandezas que permitam efectuar uma análise mais operacional da organização tais como as actividades.

Ex.: É possível definirmos que o nosso orçamento de Marketing está 30% afectado ao Centro de Custo “Jogo” e 70% afectado ao Centro de Custo “Imobiliária”.

Orçamentado – valor definido antes do período como máximo a ser gasto para uma determinada rubrica orçamental;

Realizado – concretização do orçamentado; valor ganho ou gasto no período para determinada rubrica orçamental;

Antecipações de Receita/Despesa – Valor que se prevê realizar ou gastar até ao final do ano, tendo em conta os compromissos existentes.

5.3.3 CONTROLO DE GESTÃO NO GÉNIO – CRYSTAL REPORTS

A proposta inicial da solução de Controlo de Gestão , previa a realização de vários mapas recorrendo à ferramenta Crystal Reports. Estes mapas recebem algumas parametrizações iniciais (período da listagem, orçamento, centro de custos, etc..) e depois são gerados relatórios a partir de *templates* desenvolvidos em Crystal Reports com os dados previamente inseridos na aplicação.

Os Mapas de Controlo de Gestão são os seguintes:

- Mapa de Execução de Orçamento;
- Mapa de Execução de Orçamento por Actividade;
- Mapa de Execução de Orçamento por Centro de Responsabilidade;
- Balancete de Contas, Analítico;
- Balancete de Contas por Centro de Responsabilidade;

Estes mapas contém informação que normalmente não está directamente associada na Base de Dados, sendo, por isso, denominados por “dados nas nuvens”, i.e., apenas existem numa forma relativa pois, apesar de serem dados criados a partir de dados na base de dados, não estão directamente relacionados com eles. Por este motivo, o Génio não suporta uma integração com o Crystal Reports que suporte estes casos. O Génio apresenta uma integração com o Crystal Reports que lhe providencia os dados directamente tendo por base uma área do Génio ou um conjunto de áreas que esteja relacionada directamente de forma simples (isto é, que seja possível definir através dum conjunto de menus do tipo DBEdit e com recurso a condições simples). Para criar um mapa em Crystal Reports (Figura 19) é necessário inicialmente criar o *layout* do mapa. Nos casos em que o Génio não dá suporte aos mapas, é necessária criar uma rotina manual quereencha uma base de dados temporária com os dados que queremos enviar para o mapa.

Empresa: {@g_codigo} - {@g_slogan} - {@g_descorgan} Orçamento: {@g_orcament} - entre {@f_data01} e {@f_data01} {@g_titulo} Gestor Responsável: {@f_gestor}				
Receitas:	Rubrica e Descrição	Dotação Mensal	Reforço	Inscrição
{rubrica} + {descriçao}		dotmensl	resttot	inscriçao
cenas		Total		
Desconto				
cenas.r		Total		

FIGURA 19 – MAPA DE CRYSTAL REPORTS EM EDIÇÃO.

A definição das bases de dados temporárias nas rotinas manuais são constituídas essencialmente por consultas que retiram e reagrupam os dados da Base de Dados no sentido de alimentar os *templates* de Crystal Reports.

Mapa de Execução de Orçamento

O Mapa de Execução de Orçamento é um mapa que permite comparar as dotações orçamentais das diferentes rúbricas face ao realizado para as mesmas rúbricas, num determinado período de tempo. O Mapa tem ainda de incluir as informações relativas às Antecipações de Receita. As Antecipações de Receita são uma espécie de previsão da receita futura da organização. Estas são incluídas neste mapa no sentido do gestor compreender qual o estado de execução do orçamento face não só ao realizado até àquela data mas também às antecipações de receita existentes face aos compromissos.

Para listar este mapa é necessário escolher o orçamento que queremos verificar, bem como o período temporal que queremos analisar. Um dos problemas mais verificados na construção deste tipo de mapas é que a estrutura das rúbricas orçamentais é uma estrutura em árvore. Isto é, as dotações orçamentais, por exemplo, são dotações que estão associadas apenas às rúbricas movimentáveis da árvore (folhas). As rúbricas não movimentáveis têm como dotações a soma das dotações associadas às rúbricas filho e por aí adiante.

A maior parte dos mapas de Controlo de Gestão assentam sobre um Modelo Relacional semelhante. O Modelo Relacional demonstrado na Figura 20 é um modelo relacional que é normalmente utilizado como base para os modelos relacionais da maior parte dos mapas. Demonstra a relação entre as dotações mensais (orçamentado, CLOAM), as Actividades (ACTIV), os Orçamentos (ORCAM) e as rúbricas orçamentais (ORCAM). Este modelo relacional segue as normas da empresa para a sua representação, uma vez que esta facilita a sua implementação no Génio (ver secção “2.3 Genio” para mais informações).

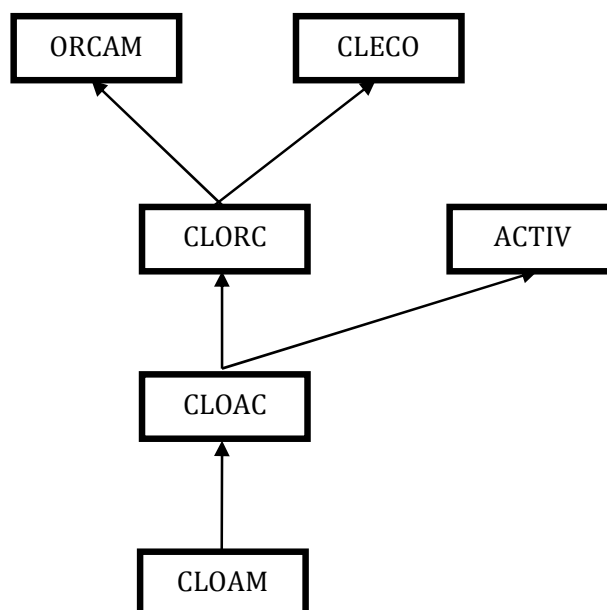


FIGURA 20 – MODELO RELACIONAL BASE DOS MAPAS.

Mapa de Execução de Orçamento por Actividade e Por Centro de Responsabilidade

Em semelhança ao Mapa de que falamos anteriormente, este Mapa contém o mesmo tipo de informação mas listada de forma diferente. Os agrupamentos dos dados não são apenas efectuados em relação à árvore de rubricas orçamentais mas também em relação às Actividades. O mapa é em tudo semelhante ao anterior mas é preciso separar as Rubricas e respectivas dotações pelas Actividades associadas. Assim, por cada actividade temos pelo menos uma página por todas as dotações que existem para uma determinada rubrica naquele período de tempo que estão associadas a determinada Actividade. Este tipo de cruzamento de dados é muito vulgar no Controlo de Gestão, que se serve de várias perspectivas da organização para formular uma opinião sobre a sua gestão estratégica.

O mapa de execução do orçamento por centro de responsabilidade. Apesar de ser semelhante ao Mapa anterior, este mapa tem a particularidade de cruzar duas estruturas em árvore. As Rubricas Orçamentais e os Centros de Responsabilidade são estruturas organizadas e relacionadas em árvore.

Além de todos os dados de entrada dos mapas anteriores, é necessário definir qual o Centro de Responsabilidade que queremos analisar. As afectações das dotações orçamentais aos Centros de Responsabilidade são apenas para os movimentáveis, em semelhança do que acontece com a árvore de rubricas orçamentais.

Caso seja indicado um Centro de Responsabilidade movimentável, os conteúdos são directos: São apenas listadas as dotações mensais com afectações àquele Centro de Responsabilidade. Contudo, se escolhermos uma visão do Mapa de Execução do Orçamento a partir de um Centro de Responsabilidade agregador (não movimentável), temos que efectuar bastantes mais passos. Temos inicialmente que construir uma árvore de Centros de Responsabilidade com base nesse Centro de Responsabilidade escolhido. Ele vai ser a raiz da nossa árvore. Depois temos em conta as dotações orçamentais que têm afectações aos Centros de Responsabilidade movimentáveis que estão presentes na árvore atrás referida, sem repetir dotações. Em relação a estas, temos também que agrupar as rubricas filho para as rubricas pai. O processo pode ser moroso e resulta num dos mapas de construção mais complexa.

Balancete de Contas

“O Balancete é, assim, um quadro recapitulativo de todas as contas de razão, onde consta a soma dos débitos e dos créditos de cada conta e os respectivos saldos (...)” [rei06]

O Balancete de Contas é um instrumento de Controlo de Gestão por excelência. Tal como é indicado pela citação acima, um Balancete de Contas contém os débitos e os créditos das contas do POC (Plano Oficial de Contabilidade), bem como os saldos daí resultantes.

Na Figura 21, temos parte de um Balancete de Contas típico. Podemos verificar as Contas do POC, com os Valores Mensais de Débito e Crédito para o mês de Julho de 2007.

No CNT, a implementação deste Balancete foi mais generalista. Devido às características únicas deste mapa, com várias necessidades de parametrização, foi criado um formulário especial que nos permite definir as várias parametrizações deste mapa.

Balancete de Contabilidade

BALANCETE DO MES

*** DETALHADO ***

Conta Inf. 1

Conta Sup. 89999999999999999999999999

Motivo A PEDIDO

			Valores Mensais		
Conta	Designação		Débito	Crédito	Saldo
1	DISPONIBILIDADES	T	905.055,53	990.699,43	85.643,90 C
11	CAIXA	T	646.653,67	646.332,12	321,55 D
111	CAIXA A	M	646.653,67	646.332,12	321,55 D
118	FUNDO PERMANENTE	T	0,00	0,00	0,00
11802	MÁRIO JOÃO DO E.SANTO ANTUNES	M	0,00	0,00	0,00
11803	PAULA MARIA DA SILVA LOPES	M	0,00	0,00	0,00

FIGURA 21 – BALANCETE DE CONTAS DA CONTABILIDADE DA CÂMARA MUNICIPAL DE AVEIRO.

Como podemos verificar na FIGURA 22, neste formulário do Balancete de Contas Analítico, podemos definir qual o intervalo de contas POC que desejamos ver listado no mapa, bem como o nível de detalhe das contas do POC (Nível POC). Podemos definir também o período desejado. Estas parametrizações permitem que o mapa seja muito mais dinâmico e confira com exactidão a informação que o gestor necessita. Caso prefiramos o Balancete de Contas por Centro de Responsabilidade, este formulário possuirá ainda informação relativa ao Nível de detalhe dos Centros de Responsabilidade desejado. Este Balancete de Contas por Centro de Responsabilidade coloca-nos um problema semelhante ao Mapa de Execução de Orçamento por Centro de Responsabilidade, uma vez que também se cruzam duas estruturas em árvore. A resolução do problema é, neste caso, em tudo semelhante ao primeiro: apenas são listadas as contas do POC cujos débitos e créditos tenham afectações ao Centro de Custo definido. A diferença é que o Mapa surge com uma opção para seleccionar o Centro de Custo desejado dentro deste, no *prevines* do Crystal Reports, em semelhança ao que acontece com o Mapa de Actividade.

Balancete Geral de Contas [?] [X]

Nível POC
 ▼

☒ Atualizar Níveis?

Datas

De Até

Conta

De Até

 **Imprimir**  **Cancelar**

FIGURA 22 – FORMULÁRIO DE PARAMETRIZAÇÃO DO BALANCETE GERAL DE CONTAS NO CNT.

Considerações Gerais

Todos os mapas acima listados encontram-se filtrados também por organismo/empresa pois o sistema possui uma estrutura que faz com que exista uma definição de cada dotação, rubrica orçamental, centro de responsabilidade, etc., associada a cada uma das empresas. Esta definição faz com que seja possível ter, numa organização com várias sub-organizações, visões particulares e consolidadas das diferentes realidades da empresa.

Esta realidade de Controlo de Gestão está adaptada segundo as necessidades actuais dos clientes da empresa, existindo outros mapas e visões de Controlo de Gestão que são aplicáveis, conforme a lógica de negócio e a estratégia da empresa no seu mercado e de acordo com a sua dimensão. A grande inovação presente nesta aplicação é a apresentação simultânea da lógica da contabilidade geral contendo o Plano Oficial de Contas, as Rúbricas Orçamentais e os Centros de Responsabilidade, (que possuem estruturas em árvore), com a lógica do Controlo de Gestão (com estrutura matricial de ordem N, sendo N o número de métricas envolvidas).

5.3.4 CONTROLO DE GESTÃO NO GÉNIO – SILVERLIGHT

Na maior parte das empresas presentes no mercado, os mapas de Controlo de Gestão são realizados com ferramentas como o Microsoft Excel, que exigem um grande trabalho manual e de intervenção do utilizador. Isto deve-se a duas razões: por um lado, estes mapas são, tipicamente, de definição matricial e dum elevado grau de complexidade que está associado e adaptado à lógica de negócio, estratégia e dimensão da empresa; por outro lado, o Controlo de Gestão acaba por ser uma disciplina de análise, distante da Contabilidade e que quer definir os seus próprios dados com uma visão mais estratégica. É normalmente com base nos dados resultantes destes mapas que se preparam Orçamentos e se define grande parte da estratégia financeira da Organização.

Se os Mapas estáticos em Crystal Reports resolvem o problema da necessidade de uma grande componente de trabalho manual na sua criação, a interactividade e a manipulação destes deixa muito a desejar. Ao contrário do Excel, não é possível interagir com o mapa, criar vistas ou mesmo alterar os valores para efeitos de Controlo de Gestão. Uma apresentação diferente, em que o utilizador não tivesse que olhar para um ecrã recheado de

números e em que pudesse rapidamente obter a informação que procura e até compará-la com outras fontes, convertendo-a em gráficos intuitivos e tornando a experiência de utilização mais Rica, é uma solução desejável.

Aqui entram as *RIA's* no Controlo de Gestão. Mais concretamente, o Microsoft Silverlight. Foi proposto que se implementasse uma versão dos actuais Mapas de Controlo de Gestão (ou pelo menos de um deles) com recurso à tecnologia de *RIA's* Microsoft Silverlight, devendo esta implementação ser muito mais do que a criação de simples mapas. Pretendia-se uma aplicação interactiva, que privilegiasse a componente gráfica, permitindo uma interacção completa na alteração dos parâmetros do mapa, no nível de detalhe, na alteração de dados do mapa para efeitos de Controlo de Gestão, etc... Deveria, ainda, permitir ao utilizador ter várias perspectivas sem ter que gerar vários mapas da forma mais familiar e intuitiva possíveis

A aplicação que daqui deve resultar foi denominada de “QuidController”. Foram então definidas as características do QuidController (QCT) como requisitos, com as prioridades de “Essencial”, “Desejável” e “Opcional”. Estes requisitos estão presentes na Tabela 2, abaixo listada.

Requisito	Prioridade	Descrição
QCT01	Essencial	Desenvolver um Mapa de Controlo de Gestão numa aplicação com recurso à tecnologia Microsoft Silverlight.
QCT02	Opcional	Desenvolver todos os Mapas de Controlo de Gestão numa aplicação com recurso à tecnologia Microsoft Silverlight.
QCT03	Essencial	Desenvolver uma aplicação com conteúdos dinâmicos.
QCT04	Essencial	Desenvolver uma aplicação com parametrizações dinâmicas <i>on-the-fly</i> para os mapas desenvolvidos.
QCT05	Desejável	Possibilitar que os valores dos Mapas possam ser editados.
QCT06	Desejável	Possibilitar que os Mapas possam ser guardados com as suas edições e recuperados mais tarde.
QCT07	Opcional	Possibilitar que os Mapas possam ser guardados e recuperados mais tarde com recurso a uma base de dados própria e independente do Controlo de Gestão.
QCT08	Essencial	Desenvolver mecanismos para que seja possível obter diferentes visões dos Mapas <i>on-the-fly</i> .
QCT09	Essencial	Integrar, de alguma forma, a aplicação desenvolvida, com o CNT e com as aplicações geradas em Génio, numa forma generalista.
QCT10	Essencial	Investigar a possibilidade de integrar RIA's no Génio com vista à integração de aplicações e componentes desenvolvidos em Microsoft Silverlight nas aplicações geradas nesta plataforma
QCT11	Desejável	Desenvolver uma aplicação de utilização fácil, com recurso a fortes componentes visuais e de aspecto altamente intuitivo e apelativo.
QCT12	Opcional	Possibilitar a geração automática de gráficos <i>on-the-fly</i> com base nos dados contidos nos mapas.
QCT13	Opcional	Possibilitar que sejam gerados vários mapas em simultâneo a partir de diferentes parametrizações e compará-los visualmente através da geração automática de gráficos.

TABELA 2 – REQUISITOS DO QUIDCONTROLLER.

Tendo em conta estes requisitos, foi desenvolvida uma solução de Controlo de Gestão e uma análise comparativa à integração de Flash e Silverlight com aplicações desenvolvidas em Génio. Estas serão descritas em detalhe nas secções seguintes.

5.4 RIA'S COM BALANCED SCORECARD

Como já verificámos no Capítulo 4, a ferramenta proprietária Balanced ScoreCard Quidgest contém uma integração razoável com o Flash, nomeadamente com recurso a componentes gráficos previamente desenvolvidos nesta tecnologia. No âmbito deste projecto, foi pedido o estudo aprofundado da tecnologia Microsoft Silverlight com vista à sua integração com Balanced ScoreCard. Esta integração significaria uma substituição dos actuais componentes gráficos em Flash por novos componentes desenvolvidos de raiz em Silverlight.

Atendendo ao facto do *Balanced ScoreCard Quidgest* ser já um produto com vários clientes no mercado, qualquer alteração da sua tecnologia base teria de ser justificada como inevitável. Os módulos desenvolvidos em Flash são essencialmente para suportar componentes gráficos que permitam demonstrar melhor o estado dos objectivos, dos indicadores, etc.. Podemos ver alguns destes mapas na Figura 12 e Figura 13 do Capítulo 4. Contudo, a componente interactiva destes é diminuta e não se vislumbra que uma utilização da tecnologia Silverlight trouxesse grandes ganhos em experiência de utilização, tendo em conta o esforço suplementar de implementação e o tipo de utilização que esta aplicação tem.

Concluimos assim não ser necessário considerar esta hipótese tendo em conta os factores acima referidos. O Balanced ScoreCard Quidgest vai manter assim o desenvolvimento em Genio, recorrendo ao Flash para os componentes gráficos.

5.5 SOLUÇÃO DESENVOLVIDA – QUID CONTROLLER

Nesta secção, vamos falar na solução desenvolvida para dar resposta ao problema colocado pelo Controlo de Gestão. Na Tabela 2, desenvolvida na secção “5.3.4 Controlo de Gestão no Genio - Silverlight”, temos os requisitos definidos para a nossa solução – o Quid Controller. Esta solução passava por integrar uma aplicação inteiramente desenvolvida em Silverlight em Génio, com boa apresentação gráfica e grande nível de interactividade. Esta definição é, só por si, um problema de complexa resolução. Em particular, leva-nos a testar ao limite esta nova tecnologia de implementação de *RIA's*.

Um dos problemas mais complexos que se coloca com Silverlight é o de *data binding*, isto é, a associação de uma fonte de dados (tal como uma Base de Dados dum SGBD). Isto acontece por duas razões:

- O Silverlight não possui quaisquer métodos que permitam efectuar a ligação a Sistemas de Gestão de Base de Dados;
- Esta tecnologia possui diversas limitações ao nível de acesso aos dados do cliente, uma vez que as aplicações de Silverlight são executadas numa espécie de “caixa de areia”, que limita os acessos aos dados do cliente (apenas é possível utilizar a *isolated storage*);

Contudo, o facto do Silverlight poder ser executado apenas num *web browser* também poderá determinar algumas limitações. Em seguida, vamos começar por definir como criamos uma aplicação básica de Silverlight que resolva um caso geral de *data binding*, para depois verificarmos como a solução definida cumpre os requisitos da aplicação, a sua forma de integração no Genio e como seria possível generalizar tal integração.

5.5.1 DATA BINDING – CASO GERAL

Para inicializar a resolução do problema que nos foi colocado, começou-se por resolver um problema mais simples. Como colocar dados de um SGBD como o *Sql Server* numa aplicação em Silverlight? A este conceito de ligação dum controlo a uma colecção de dados denominamos de *Data Binding*.

“*The datagrid control is designed to assist you in displaying data in a row/column format (...) It is bindable to a data source.*” [lmo08]

Tendo em conta que os dados a ser demonstrados seriam mapas de Controlo de Gestão, definiu-se que se iria utilizar um controlo já existente nas bibliotecas do Silverlight que responde a grande parte do nosso problema: a *datagrid*.

Para iniciar o desenvolvimento com a tecnologia, foram seguidos diversos tutoriais da página oficial do Silverlight na Internet [the08], sendo que foi seguido um em particular. Este tutorial permitia criar com recurso a WCF e LINQ, uma aplicação com *data binding* em Silverlight. [swi08]

Porquê WCF?

Devido às limitações da arquitectura do Silverlight – é executado num plug-in – e como incremento de segurança, este é executado numa espécie de “caixa de areia”. Daí que, como já fora referido na análise efectuada no Capítulo 2, não seja possível ligarmo-nos directamente a um Sistema de Gestão de Base de Dados utilizando esta tecnologia. Contudo, é possível utilizarmos um *web service* como meio de comunicação. Podemos ligarmo-nos ao *web service* e através dum simples pedido, é possível obter os dados. O *web service*, por sua vez, contém os dados de acesso à base de dados tais como a sua localização, credenciais e as consultas a efectuar. Podemos utilizar vários tipos de *web services*, contudo existe um em particular que deve ser utilizado: WCF – *Windows Communication Foundation*. Este tipo de *web services* pertence à mesma *framework* que o Silverlight e este contém o *namespace* directo para os utilizar. Para tal, basta que este esteja referenciado no *ServiceReference* para que não tenhamos de implementar a forma como ele comunica com a aplicação: é só desenvolver as funções, declará-las na *interface* e utilizá-las na aplicação.

Porquê LINQ?

O LINQ – Language Integrated Query – é uma tecnologia da *framework* .NET que nos permite utilizar colecções de dados (como tabelas) da mesma forma que utilizamos classes (Figura 23). Podemos ainda executar métodos sobre elas tais como utilizamos em SQL (*select*, *where*, *group*, *count*, etc.) ou nas linguagens de programação orientadas a objectos (*foreach*, *for*, *while*, etc..). Podemos ainda associar estas classes às tabelas físicas nos SGBD para que possamos efectuar a submissão das alterações automaticamente. Essa associação faz-se com recurso à tecnologia LINQtoSQL.

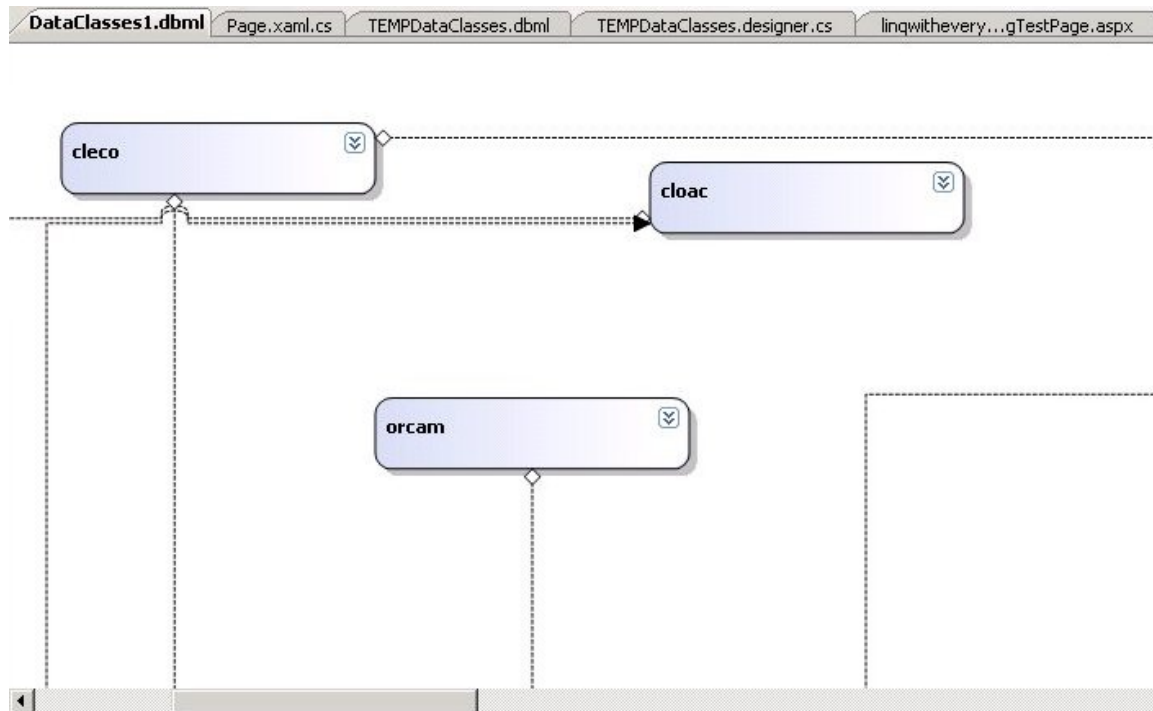


FIGURA 23 – MODELO RELACIONAL MAPEADO EM CLASSES ATRAVÉS DO LINQ TO SQL.

Uma das características mais importantes da utilização do LINQ é a possibilidade de definir as classes resultantes dos SGBDs como *serializáveis*. Esta propriedade permite que os objectos, instâncias destas classes, sejam convertíveis num conjunto de texto que pode ser transmitido através duma rede, para posteriormente ser decodificado.

Como criar a aplicação?

Depois de criarmos a aplicação em Silverlight, teremos de criar o *web service* WCF e incluir uma classe LINQtoSQL. É muito importante que criemos no novo objecto LINQtoSQL as classes respeitantes às tabelas da base de dados que desejamos utilizar. Depois temos de alterar as propriedades de serialização desta classe para “unidireccional”. É isto que fará com que os atributos da classe sejam serializáveis. Em seguida, devemos definir o ficheiro de interface do *web service* de acordo com a Figura 24.

```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    List<mapOrcarub> GetMap1(string codorcam, DateTime dt1, DateTime dt2);

    [OperationContract]
    List<orcam> GetOrcament();

    [OperationContract]
    void SaveMap1(List<mapOrcarub> mapa, string filename);

    [OperationContract]
    List<mapOrcarub> LoadMap1(string filename);
}
```

FIGURA 24 – EXEMPLO DA INTERFACE DA WEBSERVICE WCF.

Em seguida, temos de adicionar a referência ao *web service* no projecto de Silverlight. É este pequeno pormenor que criará os automatismos para que a aplicação em Silverlight e o *web service* de WCF possam comunicar. Os *operation contract* definidos na figura acima identificam a forma como as aplicações externas poderão interagir com o *web service*.

Em seguida, é necessário definir o projecto em Silverlight. Como já fora referido anteriormente na secção “2.7 Microsoft Silverlight“, um projecto em Silverlight tem duas dimensões diferentes: *design*, definido em XAML e *runtime*, definido em CSharp. Como podemos verificar na Figura 25, temos de começar por declarar um controlo do tipo *datagrid*.

```
<data:DataGrid Margin="45.299,48,151.701,17" IsEnabledChanged="dg_IsEnabledCha
CurrentCellChanged="dg_CurrentCellChanged" AutoGenerateColumns="False" Grid.R
BorderBrush="#FF034750" BorderThickness="1,1,1,1" RowBackground="#FFE5F0F2" Ba
HorizontalGridLinesBrush="#FF02494D" VerticalGridLinesBrush="{x:Null}" RenderT
```

FIGURA 25 – DECLARAÇÃO DUM CONTROLO DATAGRID EM XAML.

Depois, teremos que em *runtime* associar um *event handler* aos eventos que retornam dados a partir do *web service*. Este *event handler* destina-se ao momento em que os dados são retornados pelo *web service* no sentido de os associar, por exemplo à *datagrid*.

Um controlo *datagrid* das bibliotecas do Silverlight possui várias propriedades que não vamos analisar aqui em pormenor. Vamos analisar uma em particular pela importância

neste trabalho: *AutoGenerateColumns*. Caso esta *flag* seja definida em *design* como verdadeiro, não temos de ter qualquer preocupação com a definição do título, o tamanho ou a fonte dos conteúdos: estes são automaticamente gerados a partir da definição da propriedade *ItemsSource* da *datagrid*. Contudo, se esta *flag* estiver definida como falsa, teremos que definir manualmente todas as propriedades das colunas. Este factor reveste-se de particular importância se quisermos obter dados de forma automática e sem nos preocuparmos muito com o seu aspecto ou processamento, o que se torna particularmente importante para efectuar testes à aplicação.

Finalmente, podemos executar a nossa aplicação para a testar. Se não tivermos definidos um IIS¹², temos de possuir um servidor de desenvolvimento local de ASP.NET. A partir do momento que conseguirmos obter dados da nossa Base de Dados, podemos iniciar a definição da lógica de negócio, das animações e da interface da nossa aplicação.

5.5.2 QUID CONTROLLER – REQUISITOS

Neste capítulo, vamos começar por obter uma visão geral do interface da aplicação, analisando em seguida se e como foram cumpridos os requisitos definidos para a aplicação.

Optou-se por um interface único, simples com que o utilizador se identificasse de imediato tal como podemos verificar na Figura 26.

¹² - Internet Information Services. Web Server que podem possuir um conjunto de serviços web tais como asp.net, ftp, http, etc.



FIGURA 26 -INTERFACE BASE DA APLICAÇÃO.

Aqui podemos verificar o cumprimento de alguns requisitos base, naquela que é a implementação proposta para o mapa de execução de orçamento. Vamos em seguida analisar os requisitos previamente definidos de forma individual.

QCT01 - Este requisito definia como essencial desenvolver um Mapa de Controlo de Gestão numa aplicação com recurso à tecnologia Microsoft Silverlight. Este requisito foi

cumprido, tendo sido para tal utilizado o Mapa de Execução do Orçamento pois é um mapa que nos confere uma ideia geral de grau de execução que o orçamento definido para um determinado período cumpre.

QCT02 - Este requisito definia como opcional desenvolver todos os Mapas de Controlo de Gestão numa aplicação com recurso à tecnologia Microsoft Silverlight. Apesar de ser interessante do ponto de vista funcional implementar o conjunto total dos Mapas de Controlo de Gestão em Silverlight, este requisito nunca foi visto como prioritário pela empresa ou pelo cliente. O facto de a aposta nesta tecnologia não estar garantida por parte da Quidgest determinou que a relevância deste requisito não fosse superior. Como tal, e acrescido aos atrasos de desenvolvimento da solução base já referidos no Capítulo 3, este requisito não foi cumprido.

QCT03 - Este requisito definia como essencial desenvolver uma aplicação com conteúdos dinâmicos. A nossa aplicação cumpriu-o na totalidade. Era possível escolher na aplicação o orçamento e o período definido para que esta possa gerar o mapa. Os orçamentos listados correspondem apenas aqueles que estão listados na Base de Dados. Existe uma validação das datas e um *handle* de erros caso não existam dados consistentes na Base de Dados com a parametrização definida.

QCT04 - Este requisito definia como essencial desenvolver uma aplicação com parametrizações dinâmicas *on-the-fly* para os mapas desenvolvidos. O requisito anterior já cumpre este, uma vez que é possível alterar as parametrizações do mapa na aplicação sem ter que fazer outro período de carregamento ou deixar de visualizar o mapa actual.

QCT05 - Este requisito definia como desejável possibilitar que os valores dos Mapas possam ser editados. Como podemos verificar na Figura 27, este requisito foi cumprido. Além disso, os valores dos totais são alterados quando alteramos os valores das rubricas, assim como o valor das rubricas pai quando alteramos alguma das rubricas filho. Existe uma *checkbox* definida na aplicação que controla a opção de ser possível, ou não, editar o mapa. Assim sendo, podemos afirmar que este requisito foi superado.

Rubrica - Descrição	Orçamentação	Realizado	Antecipações	Dif. Orc. Real.
R1 - R1	361,00 €	15,57 €	288,39 €	345,43 €
R2 - R2	2649,00 €	202,83 €	60,00 €	2446,17 €
R3 - R3	120,00 €	0,00 €	0,00 €	120,00 €
R4 - R4	160,00 €	0,00 €	0,00 €	160,00 €
R5 - R5	2670,00 €	0,00 €	0,00 €	2670,00 €
R6 - R6	1200,00 €	5,01 €	78,00 €	1194,99 €
R7 - R7	245,00 €	0,00 €	0,00 €	245,00 €

FIGURA 27 – EDIÇÃO DUM MAPA DE EXECUÇÃO DE ORÇAMENTO.

QCT06 - Este requisito definia como desejável possibilitar que os Mapas possam ser guardados com as suas edições e recuperados mais tarde. Na solução desenvolvida é possível guardar mapas e recuperá-los mais tarde. Todavia, esta funcionalidade apresenta algumas limitações devido às características da tecnologia Silverlight.

É possível gravar no disco o conteúdo corrente dum determinado mapa gerado e/ou alterado. Este conteúdo é sempre gravado com um formato de ficheiro de texto, do lado do *web service*, com um nome gerado a partir do tipo de mapa e da data em que este é gravado. A pasta onde ele é gravado é também definida previamente. Para abrir estes ficheiros e efectuar o carregamento de mapas previamente gravados, recorreremos ao controlo do Windows *OpenFileDialog*.

Uma vez que apenas é possível recorrer a este controlo na aplicação em Silverlight (*runtime*) e não no *webservice*, o acesso ao disco do cliente é bastante limitado. Nós conseguimos que o controlo nos retorne o nome do ficheiro escolhido mas não o caminho que temos de percorrer para o encontrar no computador. Esta limitação é decorrente da utilização da *Isolated Storage* por parte do Silverlight e faz com que tenhamos de utilizar a mesma directoria/localização para abrir e gravar os nossos ficheiros de texto.

Apesar destas limitações, consideramos que o requisito foi cumprido na totalidade.

QCT07 - Este requisito definia como opcional possibilitar que os Mapas possam ser guardados e recuperados mais tarde com recurso a uma base de dados própria e independente do Controlo de Gestão. Devido ao facto do requisito anterior já responder à implementação desta funcionalidade e desta não possuir um grau de prioridade baixo, este requisito não foi cumprido.

QCT08 - Este requisito definia como essencial desenvolver mecanismos para que seja possível obter diferentes visões dos Mapas on-the-fly. Os botões presentes na Figura 28 permitem-nos alternar entre diferentes visões do mesmo. Uma vez que as rúbricas orçamentais são estruturas em árvore, estas são compostas por várias árvores. Estas árvores possuem, em norma, rúbricas orçamentais de nível 1 como raízes. São estas rúbricas que são demonstradas na visão inicial de qualquer mapa pois são aquelas que nos permitem obter rapidamente uma visão geral dos dados que desejamos visualizar. Contudo, para obtermos uma visão mais pormenorizada das rúbricas orçamentais que compõe uma determinada rubrica pai, utilizamos o botão “+” localizado na respectiva linha. Para obtermos um mapa detalhado com todas as rúbricas orçamentais, devemos utilizar o botão “Detalhes”, sendo que o botão “Normal” nos apresenta a visão inicial dos mapas.

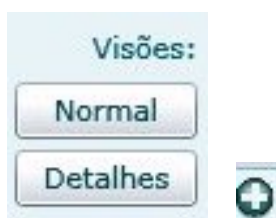


FIGURA 28 – BOTÕES QUE NOS PERMITEM ACEDER A DIFERENTES VISÕES DO MAPA.

Esta implementação não só cumpre os requisitos definidos bem como os ultrapassa na medida em que é possível obter tantas vistas do mapa quantas rubricas de primeiro nível existirem, tornando assim as vistas numa funcionalidade dinâmica.

QCT09 - Este requisito definia como essencial integrar, de alguma forma, a aplicação desenvolvida, com o CNT e com as aplicações geradas em Génio, numa forma generalista. Como poderemos verificar em seguida, este requisito foi cumprido.

Como sabemos, as aplicações em Silverlight são executadas num *plug-in* de um *browser*. Como utilizamos *web services* que necessitam de ser executados num *web server* que suporte ASP.NET, necessitamos de possuir também este *web server*. Contudo, as aplicações backoffice geradas em Genio são tipicamente executadas em formulários do *Windows* contidos na API do MFC.

Começamos por executar um browser contido num formulário numa aplicação backoffice. Uma vez que não existe qualquer ajuda de Génio disponível para backoffice que nos permita realizar esta operação, temos forçosamente que recorrer a uma rotina manual. Construímos então uma entrada de menu que nos permite chamar uma rotina manual. Dentro dessa rotina manual, temos que declarar e criar manualmente um formulário que implemente, de alguma forma, um *browser*.

Contudo, o MFC já contém na sua API as funcionalidades necessárias para declarar um *browser*. Podemos encontrar mais informação sobre esta utilização em [usi08]. Como este controlo já possui tudo o que é necessário para executar uma página de Internet dentro dum formulário de MFC, basta declarar e utilizar um formulário com estas características. Contudo, durante a investigação realizada sobre a utilização deste controlo no Génio, descobriu-se que este já era implementado na definição e utilização de rotinas manuais para *reporting services*, outra tecnologia de apoio à criação de relatórios. Assim sendo, e com apenas duas linhas de código, é possível criar um formulário que mostre uma página de Internet como um relatório dentro numa aplicação em Génio.

Para que seja possível executar a aplicação em Silverlight que pretendemos, basta possuímos o *web server* de ASP.Net que se destina a alojar a execução do nosso *web service*. Temos então duas opções:

- Possuir um IIS;
- Executar uma máquina virtual local que nos permita simular esse servidor;

Tendo em conta que as aplicações geradas tendem a ser executadas em ambientes offline, vamos optar pela segunda opção. O Microsoft Visual Studio possui uma máquina virtual de desenvolvimento associada para servir como *web server*. Com recurso ao comando do

MFC, “*shell execute*”, que nos permite executar qualquer aplicação do windows com os seus parâmetros com base numa linha de código, podemos executar uma chamada prévia ao lançamento da página web que lance o nosso web server. O código utilizado pode ser consultado nos presentes anexos.

Com isto, torna-se possível executar esta e qualquer outra aplicação de Silverlight que utilize *data binding* dentro duma aplicação em Genio. Esta funcionalidade faz com que seja cumprido na íntegra o requisito QCT09 - Integrar, de alguma forma, a aplicação desenvolvida, com o CNT e com as aplicações geradas em Génio, duma forma generalista.

QCT10 - Este requisito definia como essencial investigar a possibilidade de integrar RIA's no Génio com vista à integração de aplicações e componentes desenvolvidos em Microsoft Silverlight nas aplicações geradas nesta plataforma.

Na secção “5.2 Integração Actual - Flash“, analisámos a integração existente entre o Flash e o Genio. Como verificámos, esta assentava entre o envio de parametrizações e de eventos num formato de mensagens XML recorrendo à tecnologia MSMQ.

Para efectuar uma integração generalista, podíamos enviar as parametrizações necessárias num formato semelhante (XML) através de chamadas do mesmo *web service* que o Silverlight utiliza para aceder aos dados. Esta forma torna possível a criação duma integração generalista com qualquer aplicação de Silverlight. Podemos enviar o formato dos objectos, os dados, a localização e as credenciais da base de dados da aplicação, etc. dependendo do tipo de aplicação que estamos a utilizar.

No caso de utilizarmos uma aplicação deste tipo, podemos sempre criar uma *datagrid* generalista com várias colunas, passando depois propriedades que possamos definir em *runtime* como o título das colunas ou o seu *binding*. Desta forma simples podemos ter uma integração generalista no futuro, dependendo da forma como queremos utilizar este tipo de aplicações.

Este requisito foi, assim, cumprido na totalidade.

QCT11 - Este requisito definia como desejável desenvolver uma aplicação de utilização fácil, com recurso a fortes componentes visuais e de aspecto tão intuitivo e apelativo quanto possível. Como podemos verificar na Figura 26, a aplicação possui um interface apelativo. Por outro lado, o recurso a componentes gráficos tais como *progress bars*, *fades* e animações permite que a aplicação seja muito agradável de utilizar e bastante intuitivas.

As animações são geradas com recurso à criação de *StoryLines*. Um *StoryLine* é um controlo do Silverlight que nos permite criar animações em relação aos objectos alterando as suas propriedades ao longo duma linha temporal. Tipicamente, estas animações são desenvolvidas com recurso ao Expression Blend¹³. Podemos verificar a sua interface natural na Figura 29.

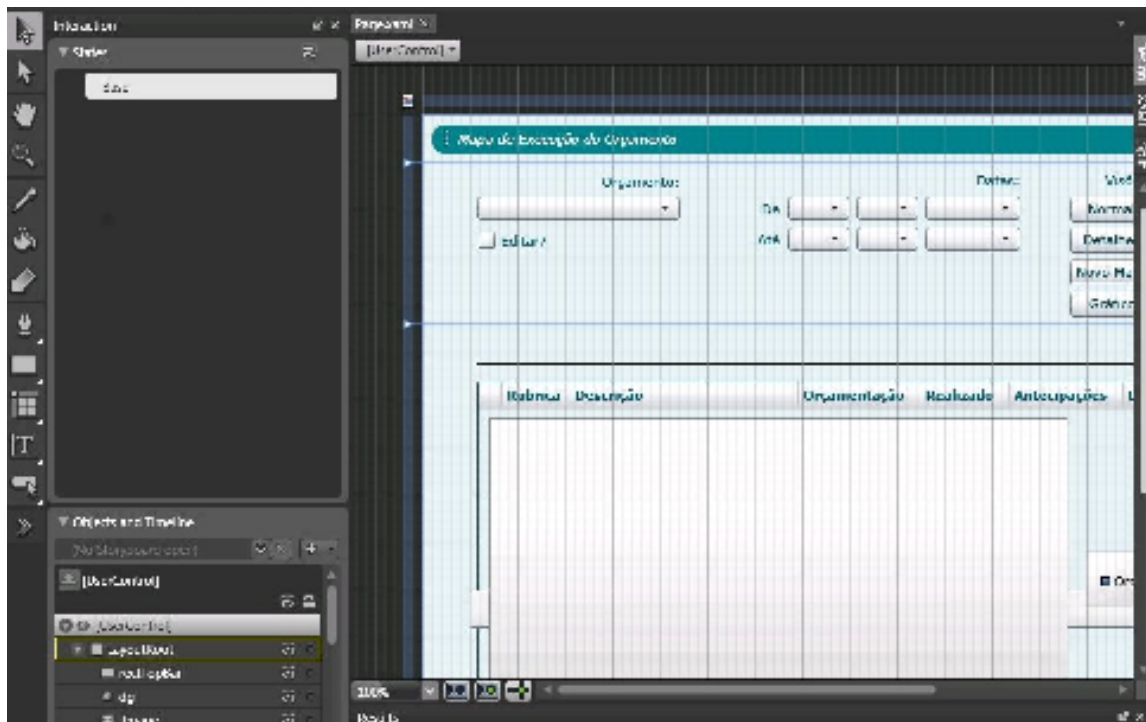


FIGURA 29 – INTERFACE DO EXPRESSION BLEND.

¹³ - Ambiente de desenvolvimento da linguagem XAML que nos permite desenvolver o design das aplicações em Silverlight. Esta aplicação é essencialmente direccionada a designers e possui um modelo de interacção semelhante a diversas aplicações que fazem edição de imagem e fotografia, tais como photoshop, paint shop pro, etc..

Devemos então definir a nossa linha temporal e o conjunto de propriedades inicial e final do nosso objecto. A renderização do Blend irá encarregar-se de gerar as matrizes de transformação necessárias para tornar a transformação o mais natural possível ao longo da linha temporal definida. Isto resulta em animações de grande qualidade definidas com grande facilidade, num espaço de tempo reduzido. Esta qualidade torna-se acrescida devido ao facto do Silverlight possuir o suporte de *Direct 3D*, tendo assim aceleração de hardware para as suas renderizações gráficas.

Considera-se assim que o requisito referido foi superado.

QCT12 - Este requisito definia como opcional possibilitar a geração automática de gráficos on-the-fly com base nos dados contidos nos mapas. Utilizando o botão “Gráfico” que existe na aplicação, esta apresenta-nos um gráfico gerado automaticamente e sem qualquer espécie de carregamento, com base nos valores orçamentados para cada uma das rubricas pai. Podemos visualizar este gráfico na Figura 30.

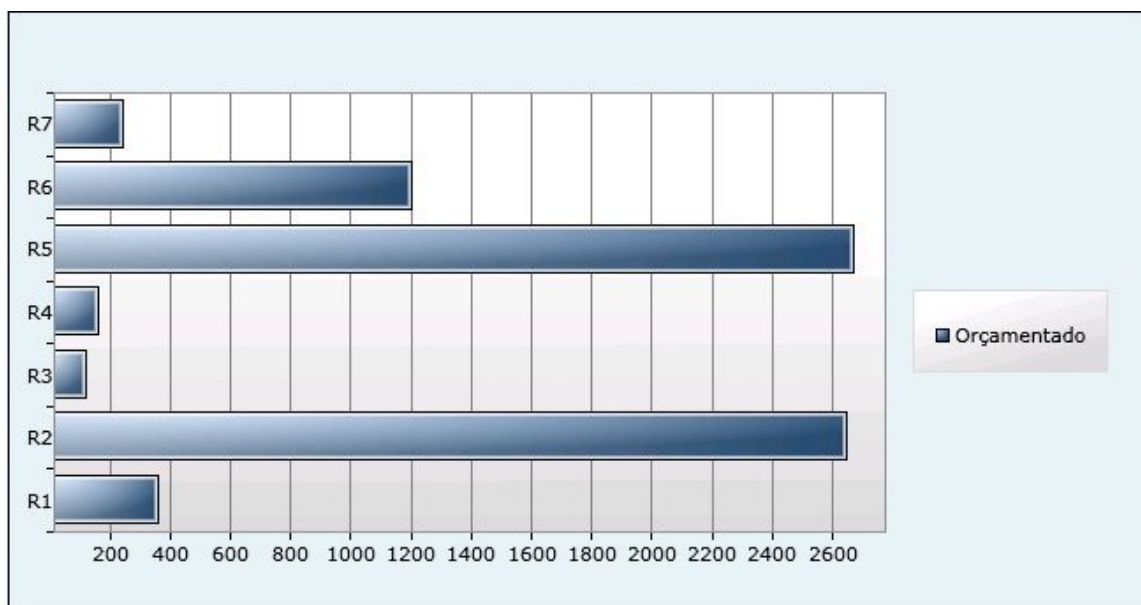


FIGURA 30 – GRÁFICO GERADO AUTOMATICAMENTE A PARTIR DOS DADOS DA ORÇAMENTAÇÃO.

Este gráfico foi construído com recurso aos controlos presentes no Silverlight Toolkit.

Silverlight Toolkit possui vários controlos que estão definidos como *charting*. Estes controlos, claramente dirigidos a aplicações *business oriented*, apresentam-nos gráficos

predefinidos e prontos a serem utilizados com uma parametrização simples. Assim sendo, temos que este requisito foi cumprido.

QCT13 - Este requisito definia como opcional possibilitar que sejam gerados vários mapas em simultâneo a partir de diferentes parametrizações e compará-los visualmente através da geração automática de gráficos. Devido aos atrasos de desenvolvimento da solução base, da dificuldade técnica colocada pelo requisito e da sua prioridade, este requisito não foi cumprido. Contudo, a forma como os controlos desta aplicação foram desenvolvidos possibilitam que uma parte significativa do trabalho esteja já desenvolvida, uma vez que as parametrizações definidas são, na sua maioria, dinâmicas e independentes dos dados utilizados.

Requisitos –Visão Geral

De uma forma geral, todos os requisitos foram cumpridos ou superados. Alguns dos requisitos de prioridade opcional não foram cumpridos devido a factores oriundos dos atrasos de desenvolvimento da solução Base, abordados em maior pormenor no Capítulo 3. Vamos em seguida abordar a integração de aplicações em Silverlight na geração de aplicações *backoffice* com recurso ao Genio.

5.5.3 PRINCIPAIS DIFICULDADES

No desenvolvimento desta solução, deparámo-nos com as seguintes dificuldades:

- Reduzido volume de documentação sobre a utilização da tecnologia neste contexto;
- Reduzida validade científica da documentação encontrada;
- Instabilidade da tecnologia, nomeadamente de alguns dos seus controlos;

O Silverlight, como tecnologia recente que é, possui poucas fontes de informação que relatem formas de resolver vários problemas que encontrámos ao longo dos presentes desenvolvimentos. Menor é ainda o número de fontes com validade científica que nos permitissem citar trabalho previamente realizado nesta área. Em específico, a utilização da *datagrid* para a resolução dum problema de *data binding* é considerado um dos problemas mais complexos que podemos resolver com recurso a esta tecnologia pela comunidade de

programadores associados. A origem desta informação, bem como muitas outras de carácter técnico, só é possível com o recurso a blogs e fóruns onde os próprios programadores que desenvolvem a tecnologia nos notificam dos seus presentes e futuros desenvolvimentos. Em particular, a *datagrid* não é um controlo estável, estando a sua última versão beta datada de Dezembro de 2008. Esta questão assume particular relevância quando funcionalidades como o “*RowDetails*” – funcionalidade que nos permite efectuar um *drill-down* automático para resolver problemas como o da listagem dos detalhes das rubricas – não poder ser ainda definida de forma individual para cada linha. Também não podemos alterar o tamanho das colunas da forma absoluta em *runtime* ou adicionar uma simples linha sem alterar a fonte dos dados. Estes problemas são citados muitas vezes como desenvolvimentos próximos deste controlo, sendo que é por vezes muito difícil compreender o limite entre aquilo que vai ser desenvolvido e aquilo que o está no presente.

5.6 FLASH COM GÉNIO VS. SILVERLIGHT COM GÉNIO

No Capítulo 2, verificámos as características base de cada uma destas tecnologias: Silverlight e Flash. Como já fora referido, a discussão sobre a integração de *RIA's* no Génio (aplicações backoffice) fica reduzida a estas duas tecnologias por opção estratégica da empresa. Neste caso, pretende-se avaliar quais as reais possibilidades de integrar aplicações em Silverlight no Génio e se sim, em que medida tal poderá ser vantajoso face às integrações existentes. Pretende-se, essencialmente, obter resposta às seguintes três questões:

1ª - É o Microsoft Silverlight uma solução viável de integração de *RIA's* no Génio? Agora ou no Futuro?

2ª - Qual a característica destas novas integrações no Génio face às existentes? Uma evolução ou um complemento?

3ª - Qual o caminho que deve ser seguido em relação ao Controlo de Gestão e BSC face aos desenvolvimentos existentes? Crystal Reports? Flash? Silverlight?

Neste capítulo, vamos compilar a informação que temos sobre as tecnologias, comparando-as directamente, bem como as suas possibilidades de integração, vantagens e desvantagens, para que possamos responder a estas três questões.

5.6.1 MÉTRICAS PARA AVALIAÇÃO DE RIA'S

O Problema que nos é aqui colocado é um problema de *RIA's* - *Rich Internet Applications*. No caso, estamos a comparar a utilização de duas tecnologias que realizam sensivelmente o mesmo tipo de aplicações para aplicá-las com um fim idêntico: a integração de aplicações desenvolvidas com recurso à plataforma de geração automática de código: Génio.

Aqui pretendem-se definir as métricas a utilizar para avaliar e comparar as duas tecnologias. Como já foi citado em capítulos anteriores, as *RIA's* são aplicações web que correm tipicamente em plug-ins de browsers. As *RIA's* possuem um conjunto de características chave:

- Suportam a utilização de renderização gráfica e a inclusão de clips media;
- Possuem uma instalação simples pois correm normalmente num plug-in que se adapta à realidade (browser, sistema operativo, etc.) dando uma experiência semelhante independentemente da plataforma em que correm;
- A manutenção dessas aplicações é simples pois os plug-ins são normalmente actualizados automaticamente;
- São aplicações mais seguras pois são pré-compiladas e utilizam, por vezes, o conceito de *sandbox*¹⁴ para o ambiente em que são executadas, o que as limita no acesso ao cliente;
- São consistentes ao nível da experiência do utilizador ao longo dos vários sistemas operativos e browsers em que são executadas;
- Melhoram no desempenho, devido ao facto de diminuírem a latência face às aplicações web tradicionais, que têm de ligar-se ao servidor constantemente para que ele processe os dados;

¹⁴ Em Segurança, uma *sandbox* é um mecanismo de segurança para executar programas de forma segura. É uma tecnologia bastante utilizada para executar código não testado ou de fontes não seguras ou certificadas.

Tendo em conta estas características e o contexto em que as pretendemos utilizar, definiu-se o seguinte conjunto de métricas para avaliar as tecnologias de *RIA* 's:

- ✓ **Experiência do Utilizador** – esta métrica pretende avaliar as capacidades das tecnologias em proporcionar uma boa experiência de utilização tais como a sua capacidade de interactividade, renderização gráfica, clips média e desempenho;
- ✓ **Segurança, Comunicação e Acesso Ao Cliente** - esta métrica pretende avaliar o nível de Segurança da tecnologia e em que medida podemos aceder e alterar os dados do Cliente e comunicar com outras aplicações;
- ✓ **Instalação e Execução** – esta métrica avalia quais as necessidades para instalar e executar uma aplicação deste tipo e quais os requisitos da aplicação cliente;
- ✓ **Manutenção e Fiabilidade** – é avaliado o esforço para manter estas aplicações funcionais assim como a fiabilidade que a tecnologia oferece;
- ✓ **Facilidade de Desenvolvimento** – define-se o quão fácil é ou não desenvolver aplicações nesta plataforma;
- ✓ **Esforço de Integração com o Génio** – tendo em conta as integrações existentes e as características das plataformas, esta métrica mede o esforço de integrar as tecnologias com o desenvolvimento de aplicações no Génio.

Estas métricas servirão como base para avaliar as tecnologias de *RIA* 's Silverlight e Flash nas secções seguintes. Vamos definir uma pontuação de 0 a 5 estrelas (🌟) para cada uma das métricas, apenas para retermos uma avaliação tangível das tecnologias.

5.6.2 CARACTERÍSTICAS DO FLASH

Tendo em conta a secção 6.2 e a análise de mercado efectuada na secção 4.3 de [mor08], apresenta-se nesta secção a análise efectuada ao Flash tendo em conta as métricas descritas na secção anterior.

Flash - Experiência do Utilizador ★★★★★

O Flash possibilita uma boa experiência de utilização graças às possibilidades de inclusão de animações renderizadas, *streaming* de áudio e de vídeo. Disponibiliza ainda uma grande quantidade de possibilidades para interacção e elevado nível de *styling*. Contudo, o Flash possui limitações, pois apesar de possuir suporte para vários tipos de ficheiros de imagem e de som, o Flash possui o seu próprio formato vídeo: o ficheiro *Flash Vídeo* (.flv). Outros formatos comerciais de ficheiros de vídeo, como o .WMV, necessitam que o codec¹⁵ esteja instalado do lado do cliente para que este seja reproduzido. O Flash assegura qualidade de reprodução independente da resolução do monitor do cliente (*vector-based*).

Flash - Segurança, Comunicação e Acesso Ao Cliente ★★★

Em termos de comunicação, o Flash não possui quaisquer funcionalidades de ligação a Sistemas de Gestão de Bases de Dados podendo, contudo, ler conteúdos e parametrizações através do acesso à leitura de ficheiros de texto (XML) numa localização remota ou mesmo do sistema de ficheiros cliente. Pode utilizar MSMQ para comunicar com outras aplicações locais e, em termos de Segurança, as limitações do Flash fazem com que seja bastante seguro executar uma apresentação deste tipo. O Flash já está no mercado à bastantes anos e com isso, desenvolveu uma relação de confiança com os actores intervenientes: utilizadores e programadores.

Flash - Instalação e Execução ★★★★★

O Flash já está à muitos anos no mercado e como tal, suporta a esmagadora maioria de sistemas operativos e browsers existentes, entre os quais o Windows 9x, Linux, Opera, IE, Mozilla Firefox, MacOS, Safari, etc.. Pode ser executado num plug-in dum browser ou num reprodutor isolado como uma aplicação de Windows. As aplicações desenvolvidas em Flash para distribuição possuem um tamanho reduzido. O arranque do plug-in é rápido.

¹⁵ - Codec é um acrónimo de Codificador/Decodificador de sinais. Neste contexto, falamos de codecs de vídeo, que nos permitem interpretar a codificação do vídeo e efectuar a sua correcta reprodução.

Flash - Manutenção e Fiabilidade ★★★★★

O facto de já existir no mercado há muitos anos confere-lhe um índice de fiabilidade superior face aos mais directos concorrentes. O plug-in do Flash é de fácil manutenção uma vez que é, normalmente, actualizado automaticamente. Apesar do facto do Flash gerar ficheiros de apresentação isolados apresentar diversas vantagens, este requer uma nova compilação de todos os seus componentes a cada alteração efectuada, o que dificulta a manutenção das aplicações.

Flash - Facilidade de Desenvolvimento ★★

Não é simples desenvolver aplicações em Flash. Flash apenas suporta a utilização duma linguagem de desenvolvimento: o *ActionScript*. Esta linguagem requer uma curva de aprendizagem elevada, uma vez que não é uma linguagem padrão nem possui outro qualquer tipo de utilização que não as aplicações Flash. Logo, o código não é reutilizável.

Não é possível utilizar quaisquer controlos do Sistema Operativo e as animações são construídas com base nas frames que vão ser demonstradas. O Flash apenas utiliza para as animações as transformações matriciais. Se quisermos, por exemplo, movimentar um objecto no espaço durante 3 segundos temos que calcular quantos frames esse número de segundos irá gastar e depois, calcular as matrizes necessárias para cada frame. Por outro lado, temos que garantir de alguma maneira que o cliente irá manter o *frame rate* da aplicação, sendo que se tal não acontecer a nossa animação poderá perfeitamente ser de 1 ou de 5 segundos (!), por exemplo.

O Flash não interpreta ficheiros de fonte (.TTF), pelo que é praticamente impossível utilizar um tipo de letra não *standard* nas suas apresentações. Esta tecnologia armazena as suas formas (*shapes*) utilizando *binary shape records*, o que necessita dum SDK¹⁶ apropriado para que se possa desenvolver as nossas definições para as formas e controlos.

O Ambiente de Desenvolvimento de Flash destina-se, tal como a sua linguagem, apenas ao desenvolvimento de aplicações Flash e requer, mais uma vez, uma curva de aprendizagem

¹⁶ SDK – Software Development Kit é um conjunto de ferramentas de desenvolvimento que permitem que o programador possa criar e alterar aplicações.

longa. Este ambiente de desenvolvimento é gráfico e por aí, mais direccionado aos designers que aos programadores.

Flash - Esforço de Integração com o Génio ★★☆☆

Apesar de já existir uma interacção com o Génio que pode ser definida como generalista, uma vez que possibilita a inclusão de N componentes desenvolvidos em Flash em aplicações de Génio geradas para *backoffice*, esta é *hardcoded*. Por cada novo componente em Flash a incluir nas aplicações, é necessário alterar as classes de interacção o que diminui em larga escala as possibilidades de integração. Existe pois, um esforço de expansão considerável na utilização do Flash, apesar do procedimento estar já completamente identificado. Por outro lado, é necessário conhecer a sintaxe do ficheiro XML que o componente Flash espera receber para que possa desenvolver algo com esse componente.

5.6.3 CARACTERÍSTICAS DO SILVERLIGHT

Tendo em conta a introdução ao Microsoft Silverlight do capítulo 1 de [lmo08] e a análise de mercado efectuada na secção 4.3 de [mor08], esta secção descreve a análise efectuada ao Silverlight de acordo com as métricas apresentadas anteriormente (secção 5.6.1 Métricas para avaliação de *RIA*'s)

Silverlight - Experiência do Utilizador ★★★★★

“Compilers and tools expose the runtime's functionality and enable you to write code that benefits from this managed execution environment(...) CLR benefits from features such as cross-language integration, cross-language exception handling, enhanced security, versioning and deployment support, a simplified model for component interaction, and debugging and profiling services.” [clr08]

O Silverlight tem um conjunto de capacidades em tudo semelhante às do Flash. Animações renderizadas, *streaming* de áudio e de vídeo, etc.. Além de suportar todos formatos media suportados pelo Flash (acrescido do WMV), este não necessita que o cliente possua quaisquer codecs de reprodução dos ficheiros vídeo, independentemente do formato. A qualidade de reprodução também é independente da resolução do monitor do Cliente. Contudo em termos de desempenho global, o Silverlight é superior pois a compilação das

suas aplicações assenta em CLR. *CLR – Common Language Runtime* é um ambiente de execução controlada de código. Isto faz com que todas as linguagens que utilizam este tipo de ambiente sejam linguagens de *managed code* – código que, ao contrário do *unmanaged code*, é executado ou pré-compilado numa máquina virtual que é, por sua vez, executada pelo CPU. Esta é uma característica de todas as linguagens .NET que são, como já verificámos no Capítulo 2, a base do *runtime* das aplicações em Silverlight.

Esta característica permite que o desempenho das aplicações em Silverlight seja superior.

Por outro lado, o Silverlight apresenta uma tecnologia que o distingue claramente da concorrência: o Deep Zoom. O Deep Zoom confere-nos a possibilidade de efectuar zoom arbitrariamente em grandes imagens com um grande desempenho. As imagens podem ser apresentadas numa escala muito grande ou muito pequena, sem que tal afecte o desempenho da aplicação em que a imagem está contida. A única propriedade que afecta o desempenho é o número de pixels que é apresentado no ecrã. [dep08] Um bom exemplo duma aplicação de Silverlight que utiliza esta tecnologia é o Hard Rock Memorabilia [har08].

O Silverlight possui ainda uma característica única: possui aceleração dos gráficos por Hardware suportada pela tecnologia *Direct 3D*.

Por este conjunto de características, considera-se que o Silverlight é superior ao Flash do ponto de vista da experiência do utilizador e, muito provavelmente, a tecnologia de RIA's que confere uma experiência mais Rica ao utilizador.

Silverlight - Segurança, Comunicação e Acesso Ao Cliente 🌟🌟🌟

Tal como o Flash, o Silverlight não possui quaisquer funcionalidade de ligação directa a Sistemas de Gestão de Bases de Dados e também pode ler conteúdos através da leitura de ficheiros/strings de texto (XML), mas tal não se pode efectuar (ao contrário do Flash) directamente do sistema de ficheiros do cliente. A forma mais simples de contornar este problema é com recurso a um *web service*. O recurso à framework .NET e ao CLR permite que se possa utilizar tecnologias como WCF, objectos ADO.NET e LINQ para aceder a Sistemas de Gestão de Base de Dados (ver Capítulo 2) Em termos de segurança, podemos

definir o Silverlight como bastante seguro. Tanto ou mais que o Flash, por exemplo. Isto deve-se ao facto do Silverlight ser executado numa *sandbox*.

Uma *sandbox* restringe tipicamente o acesso à invocação da API nativa da plataforma, controlando de forma bastante apertada o conjunto de recursos à disponibilidade da aplicação como o espaço em disco e em memória, o acesso a informações do sistema, a leitura de dispositivos de *input*, etc.. Para minimizar os efeitos destas limitações, o Silverlight recorre à tecnologia *Isolated Storage*.

“(...)When using isolated storage, applications save data to a unique data compartment that is associated with some aspect of the code's identity, such as its Web site, publisher, or signature.” [iso08]

A *Isolated Storage* é uma tecnologia que permite guardar dados provenientes duma aplicação web fora da cache do web browser. Isto faz com que os dados se mantenham no Cliente mesmo depois da cache do browser ser “limpa”. A utilização do *Isolated Storage* é independente do *browser* onde o Silverlight é executado. Esta tecnologia disponibiliza um espaço em disco muito reduzido mas é útil quando se quer guardar definições de execução no cliente.

Como já foi referido aqui e no Capítulo 2, Silverlight utiliza tecnologias da framework .Net que nos permitem comunicar com *web services* WCF. Contudo, uma das limitações do Silverlight é que esta comunicação se faça assincronamente. *“(...) All WCF service calls from Silverlight are made through asynchronous communications.” [mic08]*

Completando a afirmação anterior, podemos definir que qualquer comunicação de aplicações em Silverlight com *web service* são efectuadas através de chamadas assíncronas - a actual versão da plataforma (2.0 Beta 2) não suporta ainda chamadas síncronas.

Apesar destas funcionalidades, o Silverlight é ainda uma tecnologia pouco utilizada e relativamente recente (2 anos), encontrando-se ainda em desenvolvimento (não existe qualquer versão que não seja *Beta* ou *Release Candidate*). Consequentemente, encontra-se pouco testada, tendo uma pequena quota de mercado das *RIA*'s. Normalmente, considera-se que uma aplicação/tecnologia no mercado é segura quando esta é utilizada em várias

aplicações e/ou produtos sem que exista historial de problemas de acessos indevidos aos dados da aplicação ou que estes foram diminuindo ao longo do tempo, fazendo com a aplicação/tecnologia seja considerada perfeitamente estável. Tal não acontece com a presente tecnologia.

Silverlight - Instalação e Execução ★★ ★

É neste ponto que o Silverlight sofre a maior penalização face ao Flash. O Silverlight é executado também num *plug-in* mas não suporta muitas das plataformas standard como o sistema operativo Linux ou o browser Opera. Não suporta qualquer execução que não dependa dum *browser* nem existe qualquer forma de executar as suas aplicações num ambiente de *backoffice* puro, sem qualquer máquina virtual ou controlo *web* que simulem o ambiente típico de um *browser*. Tal como o Flash, o arranque do *plug-in* é rápido mas os seus ficheiros possuem um tamanho muito superior aos do Flash. São em média, 10 a 20 vezes superiores uma vez que não efectuam qualquer compressão dos seus ficheiros fonte. Além de serem necessários vários ficheiros com código fonte para que a aplicação seja executada, as suas dependências não se encontram embebidas na aplicação (tais como ficheiros bitmap, jpeg, icons, ficheiros áudio ou vídeo).

Silverlight - Manutenção e Fiabilidade ★★ ★ ★

Tal como o Flash, o *plug-in* é de fácil manutenção pois é actualizado automaticamente. Apesar de estar à pouco tempo no mercado, o Silverlight assenta sobre tecnologias já muito utilizadas para desenvolver aplicações *business-oriented* (.NET, C#, VB .NET, CLR, etc.), pelo que os programadores lhe conferem uma fiabilidade acima da média. Uma vez que o Silverlight possui vários ficheiros fonte, podemos facilmente identificar quais os responsáveis pelos protocolos de comunicação do *web service*, os de extracção de dados, os de *runtime*, os de design, etc.. o que facilita o *debug* das aplicações existentes e a criação de novas aplicações/funcionalidades com recurso a código já testado e existente no mercado (C#, por exemplo).

O facto de já existir no mercado há muitos anos confere-lhe um índice de fiabilidade superior face aos mais directos concorrentes. O *plug-in* do Flash é de fácil manutenção uma

vez que é, normalmente, actualizado automaticamente. Contudo, e apesar dos ficheiros Flash serem produzidos individualmente, mesmo com os ficheiros media embebidos dentro, é necessário alterar esse mesmo ficheiro e recompilá-lo de novo para qualquer alteração, por mais pequena que seja, o que complica o desenvolvimento de pequenas alterações e melhoramentos.

Silverlight - Facilidade de Desenvolvimento ★★★★★

Tal como já foi dito, o Silverlight assenta o seu desenvolvimento em tecnologias que os existentes na framework .NET: CSharp, VB, WPF, WCF, ASP.NET e LINQ. Estas linguagens são utilizadas comercialmente para desenvolver aplicações com todo o tipo de lógicas de negócio o que faz com o código existente seja completamente reutilizável.

O IDE (*Integrated Development Environment*) utilizado também é bastante conhecido – Visual Studio – e a maior parte dos programadores também apresentam grande familiaridade com a sua utilização. O Silverlight possui também o *Expression Blend* como IDE. Esta ferramenta permite-nos editar apenas o *design* das nossas aplicações e é fundamentalmente dirigido aos *designers*. A aplicação apresenta uma excelente integração com o Visual Studio, permitindo abrir os projectos tal como eles estão no seu ambiente. É através da definição do código XAML que é possível que essa integração se faça sem qualquer problema. O próprio XAML é uma linguagem com um paradigma semelhante ao HTML e de fácil compreensão.

A criação de animações em Silverlight também é facilitada: além de ser possível definir animações com base no tempo que elas duram (ao contrário do Flash, que tem como base os frames), podemos definir apenas o estado inicial e o final – a aplicação trata de gerar os estados restantes e basta ver como fica para compreender se a aplicação faz, ou não, o esperado.

O Silverlight possui uma livreria razoável de fontes de texto. Mas se o programador quiser utilizar uma fonte externa, é possível incluir no projecto ficheiros do tipo *True Type Font* (.TTF).

Como o Silverlight tem como base WPF, é possível utilizar vários controlos do Windows sem grandes esforços de integração. Outro factor interessante é a existência do Silverlight Toolkit que é, basicamente, um conjunto de skins e controlos pré-programados e prontos a utilizar em aplicações de Silverlight. Um dos controlos tem particular relevância uma vez que é um gráfico com várias formas de apresentação o que é um factor extremamente interessante para aplicações *business oriented*.

Contudo, esta tecnologia também apresenta pontos menos positivos. O debugging da aplicação é mais simples pois é possível editar um pequeno componente da aplicação sem ter de a recompilar na totalidade. Todavia, o arranque da aplicação é bastante moroso devido ao facto de ser constantemente necessário possuímos um servidor de ASP.NET a ser executado, local ou remotamente, para que possamos executar a aplicação. Como a tecnologia é recente, esta apresenta ainda bastantes falhas de desenvolvimento e pouquíssima documentação com validade científica, pelo que foi no recurso a fóruns, blogs, etc.. que se resolveram muitos dos problemas de desenvolvimento com que nos deparámos.

Silverlight - Esforço de Integração com o Génio ★★ ★

Não existe qualquer tipo de integração na geração de aplicações com o Génio com aplicações em Silverlight. Contudo, Silverlight utiliza, tal como o Génio para aplicações *backoffice*, tecnologia da *framework* .NET. Como tal, será mais fácil a adaptação dos programadores tradicionais de Genio ao ambiente de desenvolvimento do Silverlight. Através da utilização do controlo do *web browser* contido no MFC, é possível executar uma página de Internet num formulário de Génio de forma bastante simples. Podemos comunicar com a aplicação através do simples recurso a uma mensagem ao respectivo *web service*. Com recurso a estes mecanismos, torna-se relativamente simples desenvolver automatismos de integração do Silverlight com o Genio.

5.6.4 SILVERLIGHT E FLASH – QUAL UTILIZAR?

Tendo em conta o trabalho descrito bem como a análise efectuada entre as duas tecnologias nas duas secções anteriores, apresenta-se na Tabela 3 um quadro que sintetiza o estudo feito.

	Adobe Flash	Microsoft Silverlight
Experiência do Utilizador		
Segurança, Comunicação e Acesso ao Cliente		
Instalação e Execução		
Manutenção e Fiabilidade		
Facilidade de Desenvolvimento		
Esforço de integração com o Génio		

TABELA 3 – QUADRO COMPARATIVO ENTRE AS TECNOLOGIAS MICROSOFT SILVERLIGHT E ADOBE FLASH

Como podemos verificar pela análise da tabela, em 30 pontos possíveis, as tecnologias não alcançam pontuações acima dos 75%. O facto do contexto de utilização destas tecnologias serem a sua integração em aplicações de backoffice, desvirtua claramente o ambiente onde é possível retirar o melhor partido delas – a web. Apesar de tudo, temos que alguns dos critérios definidos são mais importantes que outros.

O facto do Silverlight ter obtido mais um ponto que o Flash é importante mas está longe de ser decisivo. Do ponto de vista tecnológico, a quota de mercado que o Flash possui faz com que este seja líder do mercado destas tecnologias e seja assim, a melhor opção de utilização hoje. Contudo, devido às características próprias de cada uma das tecnologias, temos que o Silverlight tem um potencial claramente superior ao Flash e terá tudo para ser um competidor muito sério no mercado a curto prazo. O facto de ainda não existir uma *release* da tecnologia – só existem versões *beta* – é bastante limitador. O Flash é um formato testado, comercial, com grande portabilidade e adaptabilidade a diferentes plataformas e de tamanho reduzido pelo que estas características ainda o deverão manter como líder de mercado no futuro.

Contudo, num mercado em expansão como é o das RIA's, existem características que vão definir muito do que vão ser as aplicações do futuro. Nomeadamente, a utilização deste tipo de aplicações em dispositivos móveis. Neste aspecto, o Flash terá mais dificuldades em acompanhar a natural evolução do mercado. Características como:

- Aceleração gráfica 3D com base em hardware;
- Animações baseadas em vectores temporais;
- Linguagem de desenvolvimento de fácil compreensão e reutilizável noutro tipo de aplicações;
- Ambiente de desenvolvimento intuitivo.

serão fundamentais no futuro deste tipo de aplicações. Nesta base, o Silverlight apresenta-se na linha da frente para suceder ao Flash, que deverá também evoluir nestas características, sob pena de se tornar obsoleto a médio prazo.

Com base nestes dados, é possível concluirmos o capítulo com as respostas às questões colocadas inicialmente:

É o Microsoft Silverlight uma solução viável de integração de RIA's no Génio? Agora ou no Futuro?

Sim, é. O facto de basear o seu desenvolvimento em tecnologias pertencentes à framework .NET com que os programadores estão já familiarizados, bem como a facilidade de aprendizagem associada ao modo de design e de definição das suas animações fazem com que seja bastante simples integrar esta tecnologia no Génio. O controlo *web browser* do MFC permite-nos abrir uma página de Internet num formulário numa aplicação backoffice, o que faz com que seja fácil executar uma aplicação de Silverlight dentro desse ambiente.

Essas integrações podem e devem ser desenvolvidas ainda Hoje. Esta tecnologia possui bastantes potencialidades para vários tipos de aplicações e sempre que seja adequado ou se espere uma experiência de utilização acima da média, a sua utilização pode e deve ser considerada.

Qual a característica destas novas integrações no Génio face às existentes? Uma evolução ou um complemento?

Ambas. A integração do Silverlight no Génio pode ser uma revolução na experiência de utilização normal das aplicações geradas nesta plataforma. Contudo, já existem diversos componentes para aplicações backoffice em Génio desenvolvidos sobre Flash e devidamente integrados. Essas integrações devem ser mantidas e caso se façam novos desenvolvimentos, devemos compreender qual o tipo de experiência que se pretende e qual das tecnologias oferecerá um maior esforço de implementação. Caso se verifique que algo existente pode ser reaproveitado, Flash pode continuar a ser uma opção e o Silverlight apenas um complemento utilizável se e quando necessário.

Qual o caminho que deve ser seguido em relação ao Controlo de Gestão e BSC face aos desenvolvimentos existentes? Crystal Reports? Flash? Silverlight?

Como já verificámos no Capítulo 4, o BSC apresenta uma integração bastante razoável com o Flash. O tipo de utilização esperada para a aplicação não justifica a aposta numa tecnologia como o Silverlight, principalmente se tivermos em conta o esforço que teríamos de empregar para colocar a aplicação tão estável quanto ela se encontra actualmente.

Em relação ao módulo de Controlo de Gestão a resposta é mais ambígua. Como já verificámos, o Controlo de Gestão é uma disciplina que se adapta à lógica de negócio e à estratégia que deverá ser seguida pela empresa. Neste caso em concreto, a aposta no Microsoft Silverlight será, concerteza, a melhor via a seguir. Se conseguirmos identificar um conjunto de mapas e funcionalidades chave, poderá ser interessante evoluir o Controlo de Gestão para um produto que contenha funcionalidades assentes nesta tecnologia. Contudo, o recurso a mapas de Crystal Reports poderá continuar a justificar-se para muitos dos mapas padrão. Uma vez que não existe qualquer componente em Flash que responda ao tipo de solicitações que o Controlo de Gestão coloca, não faz sentido reportarmo-nos a esta tecnologia como solução neste contexto.

Concluindo, temos que o Silverlight é uma tecnologia altamente inovadora e com bastante potencial futuro, mas cuja instabilidade e baixa quota de mercado coloca ainda dúvidas quanto a uma aposta definitiva. O Flash poderá manter-se como alternativa estável, sendo

que uma aposta em desenvolvimentos futuros em Silverlight também se justifica devido ao baixo esforço de compreensão das tecnologias em que se baseia.

CAPÍTULO 6 - CONCLUSÕES E TRABALHO FUTURO

Os objectivos propostos para este projecto foram cumpridos. O Sistema de Gestão Financeira para um Parque Tecnológico foi desenvolvido no âmbito duma equipa de trabalho, tendo sido cumpridos todos os seus requisitos. O módulo do Motor de Classificação de Documentos Contabilísticos foi realizado, tal como a integração do Balanced ScoreCard e a investigação da integração de BSC e do módulo de Controlo de Gestão com a tecnologia Microsoft Silverlight. Apenas não foi considerada a investigação de software sobre *Hedge Funds*, inicialmente citada no enunciado deste projecto, por a empresa ter recuado na sua intenção estratégica de abordar este mercado.

O Sistema de Gestão Financeira (CNT) foi implementado tendo em conta as necessidades particulares dum Parque Tecnológico. Este sistema foi desenhado para integrar na mesma aplicação duas realidades distintas: a contabilidade orçamental e analítica - com as suas estruturas em árvore, tais como as rubricas orçamentais – e o Controlo de Gestão, que através do cruzamento de diferentes realidades funcionais, obtém organizações matriciais dos dados. As suas funcionalidades disponibilizam vários mecanismos que facilitam a utilização do Sistema, assim como automatismos que permitem que o utilizador não necessite de um grande nível de conhecimento funcional para que a possa utilizar. Um dos automatismos mais importantes é o Motor de Classificação Automática de Documentos Contabilísticos, uma vez que permite ao utilizador processar um determinado documento sem ter de conhecer as suas associações à contabilidade geral e orçamental ou a classificação do IVA, por exemplo. Contudo, este automatismo é baseado numa classificação prévia dos artigos utilizados, sendo que seria interessante desenvolver como trabalho futuro um conjunto de automatismos completamente autónomos.

A metodologia Balanced ScoreCard implementada pela ferramenta proprietária “BSC Quidgest” responde às necessidades que a Gestão de topo tem em possuir indicadores e métricas não-financeiras que permitam compreender e melhorar de forma mais eficaz o comportamento das organizações. Este é perfeitamente integrável com o CNT através dum sistema simples de *queries* SQL entre as duas bases de dados.

Apesar do seu carácter estático, a integração existente entre os sistemas gerados a partir do Génio e o Flash é boa e funcional. É possível criar tantos componentes em Flash quantos quisermos para integrar em sistemas gerados no Génio, sendo que a sua viabilidade é boa. Pelo tipo de experiência que o utilizador usufrui e pela estabilidade da actual solução, não se justifica que o BSC Quidgest seja desenvolvido recorrendo ao Silverlight. É possível integrarmos sistemas gerados em Génio com Microsoft Silverlight executando uma simples página de Internet dentro dum formulário numa aplicação de *backoffice*.

A solução pretendida para a integração do Controlo de Gestão com *RIA's* recorrendo à tecnologia Silverlight foi desenvolvida, tendo visto todos os seus requisitos cumpridos e/ou ultrapassados. Esta solução é interessante técnica e funcionalmente, aumentando drasticamente as potencialidades existentes de acordo com a tecnologia anteriormente utilizada para responder a este problema (Crystal Reports). A maior dificuldade sentida nestes desenvolvimentos foi a pouca documentação existente sobre esta tecnologia com validade científica. Comparando o Silverlight com o Flash no contexto de integração de *RIA's* em aplicações de *backoffice* com *data binding* geradas em Genio, o Silverlight apresentou-se como a melhor opção. Apesar do Flash ser um formato à muitos anos no mercado, testado, com grande portabilidade entre diferentes plataformas, o Silverlight tem todas as condições para o suplantar como líder de mercado. O facto de possuir características únicas tais como aceleração gráfica 3D por hardware, animações baseadas em vectores temporais, linguagem de desenvolvimento de fácil compreensão e reutilizável em diferentes aplicações bem como o seu ambiente de desenvolvimento altamente intuitivo confere-lhe um futuro altamente promissor.

Tendo em conta as integrações existentes em Flash serem bastante estáveis e funcionais, entende-se que no futuro estas se deverão, numa forma geral, manter. Para novos desenvolvimentos dever-se-à recorrer ao Silverlight.

Como trabalho futuro, seria interessante abordar a temática da criação numa integração standard de aplicações criadas em Silverlight com aplicações geradas em Genio, com vista ao desenvolvimento de aplicações nesta tecnologia.

ANEXOS

A) Query em Transact-SQL para a importação do Valor do Indicador “Volume de Negócios”

```
SELECT SUM(VALOR) AS TOTAL FROM
(SELECT (CASE WHEN LEFT(CONTA,2)='71'
              THEN VALOR*-1
              ELSE 0 END) AS VALOR
FROM
(SELECT
CASE
WHEN LEFT(conta,1) = '0'
      OR LEFT(conta,2)='25'
THEN LEFT(conta,18)
ELSE (
CASE
WHEN
(
LEFT(conta,3)='243'
OR LEFT(conta,3)='261'
OR LEFT(conta,3)='262'
OR LEFT(conta,3)='268'
OR LEFT(conta,3)='648'
OR LEFT(conta,2)='28'
)
THEN LEFT(conta,4)
ELSE (
CASE
WHEN LEFT(conta,2)='23'
THEN LEFT(conta,5)
ELSE LEFT(conta,3)
END)
END)
END CONTA,
DBO.RoundQG((SUM(ISNULL(debito,0)-ISNULL(credito,0)),2)
VALOR
FROM CNTNEW2008.dbo.CNTlmovi
WHERE tipomov <>'S1'
AND data BETWEEN CONVERT(DATETIME,'01/01/2008',101) AND
CONVERT(DATETIME,'12/31/2008',101)
GROUP BY
CONTA
) AS DADOS) AS DADOS2
```


C) Query em Transact-SQL para a importação do Valor do Indicador “Resultados Líquidos”

```

SELECT SUM(VALOR) AS TOTAL FROM
(SELECT (CASE WHEN LEFT(CONTA,2)='88'
            THEN VALOR*-1
            ELSE 0 END) AS VALOR
FROM
(SELECT
CASE
WHEN LEFT(conta,1) = '0'
      OR LEFT(conta,2)='25'
THEN LEFT(conta,18)
ELSE (
CASE
WHEN
(
LEFT(conta,3)='243'
OR LEFT(conta,3)='261'
OR LEFT(conta,3)='262'
OR LEFT(conta,3)='268'
OR LEFT(conta,3)='648'
OR LEFT(conta,2)='28'
)
THEN LEFT(conta,4)
ELSE (
CASE
WHEN LEFT(conta,2)='23'
THEN LEFT(conta,5)
ELSE LEFT(conta,3)
END)
END)
END) AS VALOR
FROM CNTNEW2008.dbo.CNTlmovi
WHERE tipomov <> 'S1'
AND data BETWEEN CONVERT(DATETIME,'01/01/2008',101) AND
CONVERT(DATETIME,'12/31/2008',101)
GROUP BY
CONTA
) AS DADOS) AS DADOS2

```

D) Page.XAML – Design da aplicação em Silverlight

```

<UserControl x:Class="linqwitheverything.Page"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Width="848" Height="712.000213623047"
xmlns:prim="clr-
namespace:System.Windows.Controls.Primitives;assembly=System.Windows.Controls.Data"
xmlns:data="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
BorderBrush="{x:Null}" Background="{x:Null}"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:charting="clr-
namespace:Microsoft.Windows.Controls.DataVisualization.Charting;assembly=Microsoft.Windows.C
ontrols.DataVisualization"

mc:Ignorable="d" xmlns:linqwitheverything="clr-namespace:linqwitheverything">

    <UserControl.Resources>

        <LinearGradientBrush x:Key="Brush1" EndPoint="370,420" StartPoint="370,0"
MappingMode="Absolute" SpreadMethod="Repeat">
            <GradientStop Color="#FFA3AEB9"/>
            <GradientStop Color="#FF055D9E" Offset="0.013000000268220902"/>
            <GradientStop Color="#FF718597" Offset="1"/>
            <GradientStop Color="#FFA7CAE4" Offset="1"/>
        </LinearGradientBrush>
        <Storyboard x:Name="datagrid">
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[2].(RotateT
ransform.Angle)">
                <SplineDoubleKeyFrame KeyTime="00:00:01" Value="-5"/>
            </DoubleAnimationUsingKeyFrames>
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[0].(ScaleTr
ansform.ScaleX)">
                <SplineDoubleKeyFrame KeyTime="00:00:01" Value="0.5"/>
                <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0.35"/>
            </DoubleAnimationUsingKeyFrames>
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[0].(ScaleTr
ansform.ScaleY)">
                <SplineDoubleKeyFrame KeyTime="00:00:01" Value="0.5"/>
                <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0.35"/>
            </DoubleAnimationUsingKeyFrames>
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[1].(SkewTra
nsform.AngleX)">
                <SplineDoubleKeyFrame KeyTime="00:00:01" Value="12"/>
                <SplineDoubleKeyFrame KeyTime="00:00:02" Value="25"/>
            </DoubleAnimationUsingKeyFrames>
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[1].(SkewTra
nsform.AngleY)">
                <SplineDoubleKeyFrame KeyTime="00:00:01" Value="-3"/>
                <SplineDoubleKeyFrame KeyTime="00:00:02" Value="5"/>
            </DoubleAnimationUsingKeyFrames>
            <PointAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransformOrigin)">
                <SplinePointKeyFrame KeyTime="00:00:01" Value="0.5,0.5"/>
            </PointAnimationUsingKeyFrames>
            <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"

```

```

Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)">
    <SplineDoubleKeyFrame KeyTime="00:00:01" Value="0"/>
    <SplineDoubleKeyFrame KeyTime="00:00:02" Value="-190"/>
</DoubleAnimationUsingKeyFrames>
<DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.Y)">
    <SplineDoubleKeyFrame KeyTime="00:00:01" Value="0"/>
    <SplineDoubleKeyFrame KeyTime="00:00:02" Value="-137"/>
</DoubleAnimationUsingKeyFrames>
<DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg" Storyboard.TargetProperty="(UIElement.Opacity)">
    <SplineDoubleKeyFrame KeyTime="00:00:01" Value="1"/>
    <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0.5"/>
</DoubleAnimationUsingKeyFrames>
<DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg2" Storyboard.TargetProperty="(UIElement.Opacity)">
    <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
    <SplineDoubleKeyFrame KeyTime="00:00:03" Value="1"/>
</DoubleAnimationUsingKeyFrames>
<ObjectAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg2" Storyboard.TargetProperty="(UIElement.Visibility)">
    <DiscreteObjectKeyFrame KeyTime="00:00:00">
        <DiscreteObjectKeyFrame.Value>
            <Visibility>Visible</Visibility>
        </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="00:00:03">
        <DiscreteObjectKeyFrame.Value>
            <Visibility>Visible</Visibility>
        </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
</ObjectAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Name="progressbar">
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Duration="00:00:00.0010000" Storyboard.TargetName="progressBar"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[0].(ScaleTransform.ScaleX)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0.983"/>
    </DoubleAnimationUsingKeyFrames>
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Duration="00:00:00.0010000" Storyboard.TargetName="progressBar"
Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="5.493"/>
    </DoubleAnimationUsingKeyFrames>
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="progressBar" Storyboard.TargetProperty="(RangeBase.Value)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:01" Value="33"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="66"/>
        <SplineDoubleKeyFrame KeyTime="00:00:03" Value="100"/>
    </DoubleAnimationUsingKeyFrames>
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="progressBar" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:03" Value="1"/>
        <SplineDoubleKeyFrame KeyTime="00:00:04" Value="0.5"/>
        <SplineDoubleKeyFrame KeyTime="00:00:05" Value="0"/>
    </DoubleAnimationUsingKeyFrames>
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00.1000000" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:01" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:03" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:04" Value="1"/>
    </DoubleAnimationUsingKeyFrames>

```

```

        <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="txtProgress" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:03" Value="0.75"/>
        <SplineDoubleKeyFrame KeyTime="00:00:04" Value="0.35"/>
        <SplineDoubleKeyFrame KeyTime="00:00:05" Value="0"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Name="animChartBar">
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="chart" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="1"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Name="animChartBarFade">
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="chart" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Name="animDGfade">
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0.99"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="0"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Name="animDGnofade">
    <DoubleAnimationUsingKeyFrames BeginTime="00:00:00"
Storyboard.TargetName="dg" Storyboard.TargetProperty="(UIElement.Opacity)">
        <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
        <SplineDoubleKeyFrame KeyTime="00:00:02" Value="1"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>

</UserControl.Resources>
<Grid x:Name="LayoutRoot" Background="#FFE5F0F2">

    <Grid.RowDefinitions>
        <RowDefinition Height="0.056*"/>
        <RowDefinition Height="0.191*"/>
        <RowDefinition Height="0.551*"/>
        <RowDefinition Height="0.09*"/>
        <RowDefinition Height="0.112*"/>
    </Grid.RowDefinitions>

    <Rectangle Fill="#FF00818E" Margin="8,8.248,128,7.752" x:Name="rectTopBar"
Stroke="{x:Null}" RadiusX="10" RadiusY="10"/>

    <data:DataGrid Margin="45.299,48,151.701,17" IsEnabledChanged="dg_IsEnabledChanged"
BeginningEdit="dg_BeginningEdit" CurrentCellChanged="dg_CurrentCellChanged"
AutoGenerateColumns="False" Grid.Row="2" x:Name="dg" BorderBrush="#FF034750"
BorderThickness="1,1,1,1" RowBackground="#FFE5F0F2" Background="#FFE5F0F2"
HorizontalGridLinesBrush="#FF02494D" VerticalGridLinesBrush="{x:Null}"
RenderTransformOrigin="0.5,0.5">
        <data:DataGrid.RenderTransform>
            <TransformGroup>
                <ScaleTransform ScaleX="1"/>
                <SkewTransform/>
                <RotateTransform/>
                <TranslateTransform/>
            </TransformGroup>
        </data:DataGrid.RenderTransform>

    <data:DataGrid.Columns>

        <data:DataGridTemplateColumn Header="" Width="Auto">

```

```

        <data:DataGridTemplateColumn.CellTemplate>
            <DataTemplate>
                <StackPanel Orientation="Horizontal">
                    <Image Source="plus_iconnnormal.png"
Height="16" Width="16" HorizontalAlignment="Center" VerticalAlignment="Center"
MouseLeftButtonUp="Image_MouseLeftButtonUp" MouseEnter="Image_MouseEnter"
MouseLeave="Image_MouseLeave" MouseLeftButtonDown="Image_MouseLeftButtonDown"/>
                </StackPanel>
            </DataTemplate>
        </data:DataGridTemplateColumn.CellTemplate>
    </data:DataGridTemplateColumn>

    <data:DataGridTextColumn Header="CodigoInterno"
        Binding="{Binding Path=codigo}" Visibility="Collapsed"/>

    <data:DataGridTextColumn Width="250"
        Binding="{Binding Path=descricao}">

        <data:DataGridTextColumn.HeaderStyle>
            <Style TargetType="prim:DataGridColumnHeader">
                <Setter Property="ContentTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="Rubrica - Descrição"
TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
        </data:DataGridTextColumn.HeaderStyle>

    </data:DataGridTextColumn>

    <data:DataGridTextColumn
        Binding="{Binding Path=orcament}" >

        <data:DataGridTextColumn.CellStyle>
            <Style TargetType="data:DataGridCell">
                <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            </Style>
            <Setter Property="Foreground" Value="#FF024850"/>
        </data:DataGridTextColumn.CellStyle>

        <data:DataGridTextColumn.HeaderStyle>
            <Style TargetType="prim:DataGridColumnHeader">
                <Setter Property="ContentTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="Orçamentação"
TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
        </data:DataGridTextColumn.HeaderStyle>

    </data:DataGridTextColumn>

    <data:DataGridTextColumn
        Binding="{Binding Path=realiza}" >

        <data:DataGridTextColumn.CellStyle>
            <Style TargetType="data:DataGridCell">
                <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            </Style>
            <Setter Property="Foreground" Value="#FF024850"/>
        </data:DataGridTextColumn.CellStyle>

```

```

        </data:DataGridTextColumn.CellStyle>

        <data:DataGridTextColumn.HeaderStyle>
            <Style TargetType="prim:DataGridColumnHeader">
                <Setter Property="ContentTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="Realizado" TextAlignment="Center"
Opacity="100" Width="Auto" Foreground="#FF02596a" />
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
        </data:DataGridTextColumn.HeaderStyle>

    </data:DataGridTextColumn>

    <data:DataGridTextColumn
        Binding="{Binding Path=antecip}" >

        <data:DataGridTextColumn.CellStyle>
            <Style TargetType="data:DataGridCell">
                <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            </Style>
        </data:DataGridTextColumn.CellStyle>

        <data:DataGridTextColumn.HeaderStyle>
            <Style TargetType="prim:DataGridColumnHeader">
                <Setter Property="ContentTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="Antecipações"
TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
        </data:DataGridTextColumn.HeaderStyle>

    </data:DataGridTextColumn>

    <data:DataGridTextColumn Header="Dif. Orc. Real."
        Binding="{Binding Path=diforcreal}" >
        <data:DataGridTextColumn.CellStyle>
            <Style TargetType="data:DataGridCell">
                <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            </Style>
        </data:DataGridTextColumn.CellStyle>

        <data:DataGridTextColumn.HeaderStyle>
            <Style TargetType="prim:DataGridColumnHeader">
                <Setter Property="ContentTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="Dif. Orc. Real."
TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
        </data:DataGridTextColumn.HeaderStyle>

    </data:DataGridTextColumn>
    <data:DataGridTextColumn Header="É Despesa?"

```

```

Binding="{Binding Path=isDespesaReceita}"
Visibility="Collapsed" />
</data:DataGrid.Columns>
<data:DataGrid.AlternatingRowBackground>
  <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
    <GradientStop Color="#FF95ABAB"/>
    <GradientStop Color="#FFB6BDCD" Offset="1"/>
  </LinearGradientBrush>
</data:DataGrid.AlternatingRowBackground>
</data:DataGrid>
<Image Height="32" HorizontalAlignment="Right" Margin="0,0,128,40"
VerticalAlignment="Bottom" Width="80" Grid.Row="4" Source="novoquid.jpg"
RenderTransformOrigin="0.5,0.5" d:LayoutOverrides="HorizontalAlignment, VerticalAlignment">
  <Image.RenderTransform>
    <TransformGroup>
      <ScaleTransform ScaleX="1" ScaleY="1"/>
      <SkewTransform/>
      <RotateTransform/>
      <TranslateTransform/>
    </TransformGroup>
  </Image.RenderTransform>
</Image>
<TextBlock OpacityMask="{x:Null}" Height="Auto" HorizontalAlignment="Stretch"
Margin="30,11.753,518,12.247" x:Name="maptitle" Foreground="#FFFFFF" Text="Mapa de
Execução do Orçamento" TextWrapping="Wrap" d:LayoutOverrides="HorizontalAlignment, Height"
FontStyle="Italic" FontFamily="Portable User Interface"/>

<Image HorizontalAlignment="Left" Margin="15.5,8.248,0,7.752" x:Name="topbar"
Width="23.257" Source="topbar.png" d:LayoutOverrides="Width"/>
<Button HorizontalAlignment="Right" Margin="0,0,232.15,86"
VerticalAlignment="Bottom" Grid.Row="1" Content="Normal" RenderTransformOrigin="4.349,0.636"
x:Name="btnDefault" Foreground="#FF024850" Click="btnDefault_Click"
d:LayoutOverrides="VerticalAlignment, Width" Width="69.85" Height="22"/>
<Button HorizontalAlignment="Right" Margin="0,54,231.959,60" Content="Detalhes"
RenderTransformOrigin="4.349,0.636" x:Name="btnAll" Foreground="#FF024850"
Click="btnAll_Click" Grid.Row="1" d:LayoutOverrides="Width" Width="69.943"/>

<data:DataGrid Margin="45.943,7.708,157.057,0" AutoGenerateColumns="False"
x:Name="dgTotais" BorderBrush="#FF034750" BorderThickness="1,1,1,1"
RowBackground="#FFE5F0F2" Background="#FFE5F0F2" HorizontalGridLinesBrush="#FF02494D"
VerticalGridLinesBrush="{x:Null}" Height="Auto" CanUserReorderColumns="False"
CanUserResizeColumns="False" CanUserSortColumns="False" VerticalAlignment="Top"
Grid.Row="3">

  <data:DataGrid.Columns>

    <data:DataGridTemplateColumn Header="" >
      <data:DataGridTemplateColumn.CellTemplate>
        <DataTemplate>
          <StackPanel Orientation="Horizontal">

            </StackPanel>
          </DataTemplate>
        </data:DataGridTemplateColumn.CellTemplate>

        <data:DataGridTemplateColumn.HeaderStyle>
          <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="Visibility" Value="Collapsed"/>
            <Setter Property="Height" Value="0"/>
          </Style>
        </data:DataGridTemplateColumn.HeaderStyle>
      </data:DataGridTemplateColumn>

      <data:DataGridTextColumn Binding="{Binding Path=descricao}" Header="Rubrica-
Descrição" >

        <data:DataGridTextColumn.CellStyle>
          <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment" Value="Right"/>
            <Setter Property="Foreground" Value="#FF024850"/>
          </Style>
        </data:DataGridTextColumn.CellStyle>
      </data:DataGridTextColumn>
    </data:DataGrid.Columns>
  </data:DataGrid>

```

```

        <Setter Property="FontSize" Value="14"/>
    </Style>
</data:DataGridTextColumn.CellStyle>
<data:DataGridTextColumn.HeaderStyle>
    <Style TargetType="prim:DataGridColumnHeader">
        <Setter Property="Visibility" Value="Collapsed"/>
        <Setter Property="Height" Value="0"/>
    </Style>
</data:DataGridTextColumn.HeaderStyle>
</data:DataGridTextColumn>
<data:DataGridTextColumn
    Binding="{Binding Path=orcament}" Header="Orçamentação" >
    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>
    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="Visibility"
Value="Collapsed"/>
            <Setter Property="Height" Value="0"/>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>
</data:DataGridTextColumn>
<data:DataGridTextColumn
    Binding="{Binding Path=realiza}" Header="Realizado">
    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>
    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="Visibility" Value="Collapsed"/>
            <Setter Property="Height" Value="0"/>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>
</data:DataGridTextColumn>
<data:DataGridTextColumn
    Binding="{Binding Path=antecip}" Header="Antecipações" >
    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>

```

```

<data:DataGridTextColumn.HeaderStyle>
  <Style TargetType="prim:DataGridColumnHeader">
    <Setter Property="Visibility" Value="Collapsed"/>
    <Setter Property="Height" Value="0"/>
  </Style>
</data:DataGridTextColumn.HeaderStyle>
</data:DataGridTextColumn>

<data:DataGridTextColumn
  Binding="{Binding Path=diforcreal}" Header="Dif. Orc. Real." >
  <data:DataGridTextColumn.CellStyle>
    <Style TargetType="data:DataGridCell">
      <Setter Property="HorizontalContentAlignment"
Value="Right"/>
      <Setter Property="Foreground"
Value="#FF024850"/>
    </Style>
  </data:DataGridTextColumn.CellStyle>

  <data:DataGridTextColumn.HeaderStyle>
    <Style TargetType="prim:DataGridColumnHeader">
      <Setter Property="Visibility" Value="Collapsed"/>
      <Setter Property="Height" Value="0"/>
    </Style>
  </data:DataGridTextColumn.HeaderStyle>
</data:DataGridTextColumn>

</data:DataGrid.Columns>
<data:DataGrid.AlternatingRowBackground>
  <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
    <GradientStop Color="#FF95ABAB"/>
    <GradientStop Color="#FFB6BDCD" Offset="1"/>
  </LinearGradientBrush>
</data:DataGrid.AlternatingRowBackground>
</data:DataGrid>
  <ComboBox HorizontalAlignment="Left" Margin="45.5,28,0,0" VerticalAlignment="Top"
Grid.Row="1" Width="170.5" x:Name="cmbOrcament" d:LayoutOverrides="VerticalAlignment"/>
  <TextBlock Height="16" HorizontalAlignment="Left" Margin="45.943,8,0,0"
VerticalAlignment="Top" Width="170.557" Grid.Row="1" Text="Orcamento:" TextWrapping="Wrap"
FontWeight="Normal" FontFamily="Portable User Interface" FontStyle="Normal" FontSize="11"
TextAlignment="Right" Foreground="#FF02596A" x:Name="lblOrcament"
d:LayoutOverrides="Width"/>
  <TextBlock Height="16" HorizontalAlignment="Right" Margin="0,7,232,0"
x:Name="lblViews" VerticalAlignment="Top" Width="70.134" FontFamily="Portable User
Interface" FontSize="11" FontStyle="Normal" FontWeight="Normal" Foreground="#FF02596A"
Text="Visões:" TextAlignment="Right" TextWrapping="Wrap" Grid.Row="1"
d:LayoutOverrides="Width"/>
  <ComboBox Margin="307.566,28,0,0" VerticalAlignment="Top" Grid.Row="1" Width="52"
d:LayoutOverrides="VerticalAlignment, Width" x:Name="cmbFromDay" HorizontalAlignment="Left"
/>
  <ComboBox Margin="365.566,28,0,0" Grid.Row="1" d:LayoutOverrides="Height"
VerticalAlignment="Top" x:Name="cmbFromMonth"
SelectionChanged="cmbFromMonth_SelectionChanged" HorizontalAlignment="Left" Width="52"/>
  <ComboBox Margin="423.783,28,344.217,0" Grid.Row="1"
d:LayoutOverrides="HorizontalAlignment, Height" VerticalAlignment="Top" x:Name="cmbFromYear"
SelectionChanged="cmbFromYear_SelectionChanged"/>
  <ComboBox Margin="308,54,0,62" x:Name="cmbToDay" Width="52" Grid.Row="1"
d:LayoutOverrides="Width, Height" HorizontalAlignment="Left" />
  <ComboBox Margin="366,54,0,62" x:Name="cmbToMonth" Grid.Row="1"
d:LayoutOverrides="Height" SelectionChanged="cmbToMonth_SelectionChanged"
HorizontalAlignment="Left" Width="52"/>
  <ComboBox Margin="0,54,344.217,62" x:Name="cmbToYear" Width="79.783"
HorizontalAlignment="Right" Grid.Row="1" d:LayoutOverrides="HorizontalAlignment, Width,
Height" SelectionChanged="cmbToYear_SelectionChanged"/>
  <TextBlock Height="16" Margin="333.16,7,344.283,0" VerticalAlignment="Top"
Width="170.557" Text="Datas:" TextWrapping="Wrap" FontWeight="Normal" FontFamily="Portable
User Interface" FontStyle="Normal" FontSize="11" TextAlignment="Right"
Foreground="#FF02596A" x:Name="lblDatas" Grid.Row="1"/>

```

```

        <TextBlock Height="16" Margin="244,31,0,0" VerticalAlignment="Top" Width="57.401"
Text="De" TextWrapping="Wrap" FontWeight="Normal" FontFamily="Portable User Interface"
FontStyle="Normal" FontSize="11" TextAlignment="Right" Foreground="#FF02596A"
x:Name="lblDatasDe" Grid.Row="1" HorizontalAlignment="Left" d:LayoutOverrides="Width"/>
        <TextBlock Margin="244,57,0,63" Width="57.401" TextWrapping="Wrap"
FontWeight="Normal" FontFamily="Portable User Interface" FontStyle="Normal" FontSize="11"
TextAlignment="Right" Foreground="#FF02596A" x:Name="lblDatasAté" HorizontalAlignment="Left"
Grid.Row="1" Text="Até" d:LayoutOverrides="Width"/>
        <TextBlock Height="16" HorizontalAlignment="Right" Margin="0,7,151.701,0"
x:Name="lblFunctions" VerticalAlignment="Top" Width="70.134" FontFamily="Portable User
Interface" FontSize="11" FontStyle="Normal" FontWeight="Normal" Foreground="#FF02596A"
Text="Funções:" TextAlignment="Right" TextWrapping="Wrap" Grid.Row="1"
d:LayoutOverrides="Width"/>
        <Button HorizontalAlignment="Right" Margin="0,28,151.959,0" Content="Atualizar"
RenderTransformOrigin="4.349,0.636" x:Name="btnUpdate" Foreground="#FF024850" Width="69.943"
Grid.Row="1" d:LayoutOverrides="Width, Height" Height="22" Click="btnUpdate_Click"
VerticalAlignment="Top"/>
        <Button HorizontalAlignment="Right" Margin="0,55,151.959,58.992" Content="Gravar"
RenderTransformOrigin="4.349,0.636" x:Name="btnSave" Foreground="#FF024850" Width="69.943"
Height="22" Click="btnSave_Click" Grid.Row="1" d:LayoutOverrides="Width, Height"/>
        <Button HorizontalAlignment="Right" Margin="0,0,152,33" Content="Abrir"
RenderTransformOrigin="4.349,0.636" x:Name="btnOpen" Foreground="#FF024850" Width="69.943"
Height="22" Click="btnOpen_Click" VerticalAlignment="Bottom" Grid.Row="1"
d:LayoutOverrides="Width, Height"/>
        <Button HorizontalAlignment="Right" Margin="0,0,232,33" Content="Novo Mapa"
RenderTransformOrigin="4.349,0.636" x:Name="btnNewMap" Foreground="#FF024850" Width="69.943"
Height="22" VerticalAlignment="Bottom" Grid.Row="1" d:LayoutOverrides="Width, Height"
Click="btnNewMap_Click"/>
        <Button HorizontalAlignment="Right" Margin="0,0,152,6" Content="Gerar"
RenderTransformOrigin="4.349,0.636" x:Name="btnGerar" Foreground="#FF024850" Width="69.943"
Height="22" VerticalAlignment="Bottom" Grid.Row="1" d:LayoutOverrides="Width, Height"
Visibility="Collapsed" Click="btnGerar_Click"/>
        <Rectangle Height="23" HorizontalAlignment="Left" Margin="45.5,13,0,0"
x:Name="rectFilename" VerticalAlignment="Top" Width="159.057" Opacity="1" Grid.Row="2"
RadiusX="10" RadiusY="20" Fill="{x:Null}" Stroke="#FF02596A" Visibility="Collapsed"
d:LayoutOverrides="Width"/>
        <TextBlock Height="17" Margin="60.587,16,0,0" VerticalAlignment="Top" Grid.Row="2"
Text="Mapa Em Edição: " TextWrapping="Wrap" Foreground="#FF02596A" FontStyle="Normal"
FontFamily="Portable User Interface" x:Name="txtMapLabel" Width="Auto"
HorizontalAlignment="Left" FontSize="10" Visibility="Collapsed"/>
        <TextBlock Height="17" HorizontalAlignment="Left" Margin="148.557,17,0,0"
x:Name="txtFilename" VerticalAlignment="Top" Width="55.557" FontFamily="Portable User
Interface" FontStyle="Italic" Foreground="#FF02596A" Text=""ficheiro""
TextWrapping="Wrap" Grid.Row="2" FontSize="9" d:LayoutOverrides="Width"
Visibility="Collapsed"/>
        <CheckBox HorizontalAlignment="Left" Margin="46,0,0,62" VerticalAlignment="Bottom"
Grid.Row="1" Content="Editar?" d:LayoutOverrides="VerticalAlignment" Foreground="#FF02596A"
x:Name="chkEdit" Checked="chkEdit_Checked" Unchecked="chkEdit_Unchecked"/>
        <data:DataGrid Margin="45.712,48,151.288,17" IsEnabledChanged="dg_IsEnabledChanged"
BeginningEdit="dg_BeginningEdit" CurrentCellChanged="dg_CurrentCellChanged"
AutoGenerateColumns="False" x:Name="dg2" BorderBrush="#FF034750" BorderThickness="1,1,1,1"
RowBackground="#FFE5F0F2" Background="#FFE5F0F2" HorizontalGridLinesBrush="#FF02494D"
VerticalGridLinesBrush="{x:Null}" RenderTransformOrigin="0.5,0.5" Grid.Row="2">

                <data:DataGrid.Columns>

                        <data:DataGridTemplateColumn Header="" Width="Auto">
                                <data:DataGridTemplateColumn.CellTemplate>
                                        <DataTemplate>
                                                <StackPanel Orientation="Horizontal">

                                                        <Image Source="plus_Iconnormal.png"
Height="16" Width="16" HorizontalAlignment="Center" VerticalAlignment="Center"
MouseLeftButtonUp="Image_MouseLeftButtonUp" MouseEnter="Image_MouseEnter"
MouseLeave="Image_MouseLeave" MouseLeftButtonDown="Image_MouseLeftButtonDown"/>

                                                        </StackPanel>
                                                </DataTemplate>
                                </data:DataGridTemplateColumn.CellTemplate>
                        </data:DataGridTemplateColumn>

```

```

<data:DataGridTextColumn Header="CodigoInterno"
    Binding="{Binding Path=codigo}" Visibility="Collapsed"/>

<data:DataGridTextColumn Width="250"
    Binding="{Binding Path=descricao}">

    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="ContentTemplate">
                <Setter.Value>
                    <DataTemplate>
                        <TextBlock Text="Rubrica
- Descrição" TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                    </DataTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

<data:DataGridTextColumn
    Binding="{Binding Path=orcament}" >

    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>

    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="ContentTemplate">
                <Setter.Value>
                    <DataTemplate>
                        <TextBlock
Text="Orçamentação" TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a"
/>
                    </DataTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

<data:DataGridTextColumn
    Binding="{Binding Path=realiza}" >

    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>

    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="ContentTemplate">
                <Setter.Value>
                    <DataTemplate>
                        <TextBlock
Text="Realizado" TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                    </DataTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

```

```

        </Setter>
    </Style>
</data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

<data:DataGridTextColumn
    Binding="{Binding Path=antecip}" >

    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>

    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="ContentTemplate">
                <Setter.Value>
                    <DataTemplate>
                        <TextBlock
Text="Antecipações" TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a"
/>
                    </DataTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

<data:DataGridTextColumn Header="Dif. Orc. Real."
    Binding="{Binding Path=diforcreal}" >
    <data:DataGridTextColumn.CellStyle>
        <Style TargetType="data:DataGridCell">
            <Setter Property="HorizontalContentAlignment"
Value="Right"/>
            <Setter Property="Foreground"
Value="#FF024850"/>
        </Style>
    </data:DataGridTextColumn.CellStyle>

    <data:DataGridTextColumn.HeaderStyle>
        <Style TargetType="prim:DataGridColumnHeader">
            <Setter Property="ContentTemplate">
                <Setter.Value>
                    <DataTemplate>
                        <TextBlock Text="Dif.
Orc. Real." TextAlignment="Center" Opacity="100" Width="Auto" Foreground="#FF02596a" />
                    </DataTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </data:DataGridTextColumn.HeaderStyle>

</data:DataGridTextColumn>

<data:DataGridTextColumn Header="É Despesa?"
    Binding="{Binding Path=isDespesaReceita}"
Visibility="Collapsed" />
</data:DataGrid.Columns>
<data:DataGrid.AlternatingRowBackground>
    <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
        <GradientStop Color="#FF95ABAB"/>
        <GradientStop Color="#FFB6BDCD" Offset="1"/>
    </LinearGradientBrush>
</data:DataGrid.AlternatingRowBackground>

```

```

<data:DataGrid.RenderTransform>
    <TransformGroup>
        <ScaleTransform ScaleX="1"/>
        <SkewTransform/>
        <RotateTransform/>
        <TranslateTransform/>
    </TransformGroup>
</data:DataGrid.RenderTransform>
</data:DataGrid>
<ProgressBar Height="32" Margin="38.5,0,152,136" VerticalAlignment="Bottom"
Grid.Row="2" RenderTransformOrigin="0.5,0.5" x:Name="progressBar"
ValueChanged="progressBar_ValueChanged">
    <ProgressBar.RenderTransform>
        <TransformGroup>
            <ScaleTransform/>
            <SkewTransform/>
            <RotateTransform/>
            <TranslateTransform/>
        </TransformGroup>
    </ProgressBar.RenderTransform>
</ProgressBar>
<TextBlock Height="32" Margin="322,0,416,136" VerticalAlignment="Bottom"
Grid.Row="2" Text="0 %" TextWrapping="Wrap" TextAlignment="Center" FontSize="18"
Opacity="0.75" x:Name="txtProgress"/>

<charting:Chart x:Name="chart" Grid.Row="2" Margin="45.299,33,151.288,17"
RenderTransformOrigin="0.5,0.5">
    <charting:Chart.RenderTransform>
        <TransformGroup>
            <ScaleTransform ScaleY="1"/>
            <SkewTransform/>
            <RotateTransform/>
            <TranslateTransform/>
        </TransformGroup>
    </charting:Chart.RenderTransform>
    <charting:Chart.Series>
        <charting:BarSeries DependentValueBinding="{Binding
Path=Value}"
                                Title="Orçamentado" IndependentValueBinding="{Binding
Path=Key}"/>
    </charting:Chart.Series>
</charting:Chart>
<Button HorizontalAlignment="Right" Margin="0,0,232.15,6" Content="Gráfico"
RenderTransformOrigin="4.349,0.636" x:Name="btnGraph" Foreground="#FF024850" Width="69.943"
Height="22" VerticalAlignment="Bottom" Click="btnGraph_Click" Grid.Row="1"
d:LayoutOverrides="Height"/>

</Grid>
</UserControl>

```

E) Page.XAML.CS – RunTime da aplicação em Silverlight

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using System.IO;
using System.Threading;
using System.Collections.ObjectModel;
using Microsoft.Windows.Controls.DataVisualization.Charting;

using linqwitheverything.ServiceReference1;

namespace linqwitheverything
{

    public partial class Page : UserControl
    {
        private ServiceReference1.Service1Client client;
        private ObservableCollection<mapOrcarub> myItems;
        private ObservableCollection<mapOrcarub> ltTemp;
        private ObservableCollection<orcam> listOrcam;
        private List<int> days;
        private List<int> months;
        private List<int> years;
        private bool errorState = true;
        private bool generateState = true;
        private string filename;
        private bool inEdit;
        private int dgActiv;

        private void setFromDate(int d, int m, int y)
        {
            cmbFromDay.SelectedItem = d;
            cmbFromMonth.SelectedItem = m;
            cmbFromYear.SelectedItem = y;
        }

        private void setToDate(int d, int m, int y)
        {
            cmbToDay.SelectedItem = d;
            cmbToMonth.SelectedItem = m;
            cmbToYear.SelectedItem = y;
        }

        private void inicializaValores()
        {
            int i;
            years = new List<int>();
            months = new List<int>();
            days = new List<int>();
            inEdit = false;
            txtProgress.Visibility = Visibility.Collapsed;
            chart.Visibility = Visibility.Collapsed;
            dgActiv = 1;
            DateTime dt = DateTime.Now;
            for (i = dt.Year; i > 1899; i--)
                years.Add(i);
        }
    }
}

```

```

        cmbToYear.ItemsSource = years;
        cmbFromYear.ItemsSource = years;

        dg2.Visibility=Visibility.Collapsed;
        progressBar.Visibility = Visibility.Collapsed;
        for (i = 1; i < 13; i++)
            months.Add(i);

        cmbToMonth.ItemsSource = months;
        cmbFromMonth.ItemsSource = months;

        for (i = 1; i < 32; i++)
            days.Add(i);

        cmbFromDay.ItemsSource = days;
        cmbToDay.ItemsSource = days;

        setFromDate(1, 1, dt.Year);
        setToDate(31, 12, dt.Year);

        dgTotais.Visibility = Visibility.Collapsed;
        getActivDataGrid(). MaxHeight = 350;
        getActivDataGrid().Visibility = Visibility.Collapsed;
    }

    public Page()
    {

        InitializeComponent();
        inicializaValores();

        //Ligação ao Webservice
        client = new Service1Client();

        //Bloqueio dos controlos
        setControlsState(false);

        //inicialização da combobox dos orçamentos

        client.GetOrcamentCompleted += new
        EventHandler<GetOrcamentCompletedEventArgs>(client_getOrcamentCompleted);
        client.GetOrcamentAsync();

        //Declaração do handler para leitura do mapa
        client.GetMap1Completed += new
        EventHandler<GetMap1CompletedEventArgs>(client_GetMap1Completed);

        client.SaveMap1Completed += new
        EventHandler<System.ComponentModel.AsyncCompletedEventArgs>(client_SaveMap1Completed);
        //Comunicação Assíncrona com o Webservice

        client.LoadMap1Completed +=new
        EventHandler<LoadMap1CompletedEventArgs>(client_LoadMap1Completed);

        progressbar.Completed +=new EventHandler(progressbar_Completed);

        animDGfade.Completed +=new EventHandler(animDGfade_Completed);

        animChartBar.Completed +=new EventHandler(animChartBar_Completed);

        animChartBarFade.Completed +=new EventHandler(animChartBarFade_Completed);

        animDGnofade.Completed += new EventHandler(animDGnofade_Completed);
    }

    void animDGnofade_Completed(object sender, EventArgs e)
    {

```

```

        setControlsState(true);
        dgTotais.Visibility = Visibility.Visible;
    }
    void animChartBarFade_Completed(object sender, EventArgs e)
    {
        chart.Visibility = Visibility.Collapsed;
        animDGnofade.Begin();
    }
    void animChartBar_Completed(object sender, EventArgs e)
    {
        btnGraph.IsEnabled = true;
    }
    void animDGfade_Completed(object sender, EventArgs e)
    {
        animChartBar.Begin();
        chart.Visibility = Visibility.Visible;
    }
    void progressbar_Completed(object sender, EventArgs e)
    {
        TimeSpan s = new TimeSpan(0, 0, 3);
        if (progressbar.GetCurrentTime() == s)
        {
            progressbar.Pause();
        }

        if (progressbar.GetCurrentTime() > s)
            progressBar.Visibility = Visibility.Collapsed;
    }
    /*
    * Handler para o evento de efectuar o Loading directo dum Mapa
    */
    private void setGenerateState(bool state)
    {
        bool changed;
        changed = generateState != state;
        generateState = state;
        if (state == false)
        {
            setControlsState(false);
            btnSave.IsEnabled = true;
            btnOpen.IsEnabled = true;
            setDataGridState(true);

            btnGerar.Content = "Gerar";
            getActivDataGrid().Columns[0].Visibility = Visibility.Visible;
            dgTotais.Columns[0].Visibility = Visibility.Visible;

            txtMapLabel.Visibility = Visibility.Visible;
            txtFilename.Visibility = Visibility.Visible;
            rectFilename.Visibility = Visibility.Visible;
        }
        else
        {
            btnGerar.Content = "Voltar";
            setControlsState(true);
            if (changed)
                setDataGridState(false);

            txtMapLabel.Visibility = Visibility.Collapsed;
            txtFilename.Visibility = Visibility.Collapsed;
            rectFilename.Visibility = Visibility.Collapsed;
        }
    }

```

```

}
void client_LoadMap1Completed(object sender, LoadMap1CompletedEventArgs e)
{
    try
    {
        myItems = e.Result;
        errorState = false;
    }
    catch (Exception error)
    {
        myItems = itemListError("Dados não encontrados");
    }

    ObservableCollection<mapOrcarub> mainItems = getItems(myItems, 1);
    setDataGrid(mainItems);
    setGenerateState(false);

    setBtnGerarState(true);
}

private void setBtnGerarState(bool state)
{
    if (state == true)
    {
        btnGerar.Visibility = Visibility.Visible;
        txtMapLabel.Visibility = Visibility.Visible;
        txtFilename.Text = "\"" + filename + "\"";

        txtFilename.Visibility = Visibility.Visible;
        txtFilename.Width = filename.Length * 6 + 12;
        txtFilename.UpdateLayout();

        double d = (double) txtFilename.Width;
        d = d + 50 + (double) txtMapLabel.ActualWidth;

        rectFilename.Width = d;
        rectFilename.Visibility = Visibility.Visible;
    }
    else
    {
        btnGerar.Visibility = Visibility.Collapsed;
        txtMapLabel.Visibility = Visibility.Collapsed;
        txtFilename.Visibility = Visibility.Collapsed;
        rectFilename.Visibility = Visibility.Collapsed;
    }
}
/*
 * Coloca as datagrids a enabled ou disabled
 */
private void setDataGridState(bool state)
{
    btnAll.IsEnabled = state;
    btnDefault.IsEnabled = state;
    getActivDataGrid().IsEnabled = state;
    dgTotais.IsEnabled = state;
}
/*
 * Handler para o final da operação de gravação do mapa
 */
void client_SaveMap1Completed(object sender,
System.ComponentModel.AsyncCompletedEventArgs e)
{
    System.Windows.Browser.HtmlPage.Window.Alert("OK!");
    setControlsState(true);
    setDataGridState(true);
}

```

```

    }

    /*
    * Retorna o nível duma rubrica, dado o seu objecto composto
    *
    */
    private int getNivel(mapOrcarub item)
    {
        string s = item.descrica;
        int nivel;
        if (s[0] != ' ')
            nivel = 0;
        else
            nivel = (s.Length - s.TrimStart(' ').Length) / 4; //numero de espaços
        return nivel+1;
    }

    /*
    * retorna as rubricas de um determinado nível, dado uma lista de items
    *
    */
    private ObservableCollection<mapOrcarub> getItems(ObservableCollection<mapOrcarub>
items, int nivel)
    {
        ObservableCollection<mapOrcarub> lt = new ObservableCollection<mapOrcarub>();
        var query = from r in items
                    where getNivel(r) == 1
                    select r;

        query.OrderBy(r => r.descrica);

        foreach (mapOrcarub m in query)
            lt.Add(m);

        return lt;
    }

    /*
    * Formatação de valores double para a string a ser mostrada na DataGrid
    *
    */
    private string formatFloatValue(double d)
    {
        string s = Math.Round(d, 2).ToString();
        if (s.Contains(',') == false) s += ",00";
        return s;
    }

    /*
    * Definição dos totais dos valores da datagrid
    *
    */
    private ObservableCollection<mapOrcarub> getTotals(ObservableCollection<mapOrcarub>
items)
    {
        ObservableCollection<mapOrcarub> mainItems = getItems(items, 1);

        mapOrcarub m=new mapOrcarub();
        string currency=mainItems[0].antecip.Substring(mainItems[0].antecip.Length-2);
        double antecip=0,orcament=0,realiza=0,diforcreal=0;
        foreach(mapOrcarub item in mainItems)
        {
            antecip+=Double.Parse((string) item.antecip.Substring(0,item.antecip.Length-
2));
            orcament+=Double.Parse((string)
item.orcament.Substring(0,item.orcament.Length-2));
            realiza+=Double.Parse((string) item.realiza.Substring(0,item.realiza.Length-
2));
            diforcreal+=Double.Parse((string)
item.diforcreal.Substring(0,item.diforcreal.Length-2));
        }
    }

```

```

    }

    m.anticip = formatFloatValue(anticip) + currency;
    m.orcament = formatFloatValue(orcament) + currency;
    m.realiza = formatFloatValue(realiza) + currency;
    m.diforcereal = formatFloatValue(diforcereal) + currency;
    m.descrica = "Totais:";
    mainItems=new ObservableCollection<mapOrcarub>();
    mainItems.Add(m);
    return mainItems;
}

/*
 * Atribuição de nova source de dados à datagrid
 *
 */
private void setDataGrid(ObservableCollection<mapOrcarub> items)
{
    DataGrid mydg;
    if (dgActiv == 1) mydg = dg;
    else mydg = dg2;
    int rowNum = items.Count();
    int height = (int)Math.Round((rowNum + 1) * 24.75, 0);

    setControlsState(false);

    getActivDataGrid().Height = Math.Min(height, getActivDataGrid().MaxHeight);
    if (!inEdit)
        getActivDataGrid().ItemsSource = items;

    ObservableCollection<mapOrcarub> sumItems = getTotals(items);

    //dgTotais.Height = 26;
    dgTotais.ItemsSource = sumItems;

    dgTotais.Columns[0].Width = getActivDataGrid().Columns[1].Width;
    dgTotais.Columns[1].Width = getActivDataGrid().Columns[2].Width;
    dgTotais.Columns[2].Width = getActivDataGrid().Columns[3].Width;
    dgTotais.Columns[3].Width = getActivDataGrid().Columns[4].Width;
    dgTotais.Columns[4].Width = getActivDataGrid().Columns[5].Width;
    dgTotais.Columns[5].Width = getActivDataGrid().Columns[6].Width;

    dgTotais.VerticalScrollBarVisibility =
System.Windows.Controls.ScrollBarVisibility.Hidden;
    dgTotais.HorizontalScrollBarVisibility =
System.Windows.Controls.ScrollBarVisibility.Hidden;
    dgTotais.Columns[0].Visibility = System.Windows.Visibility.Visible;

    dgTotais.UpdateLayout();
    dgTotais.IsReadOnly = true;
    if (errorState == false)
    {
        dgTotais.Visibility = Visibility.Visible;
        getActivDataGrid().Visibility = Visibility.Visible;
    }
    setControlsState(true);

    if (dgActiv == 1)
        getActivDataGrid().Visibility = Visibility.Visible;
    else
        datagrid.Begin();

    dgTotais.Visibility = Visibility.Visible;
    if (dgActiv == 1)
        getActivDataGrid().IsEnabled = true;
}

/*
 *
 *
 * Handler do evento que acontece quando acaba de ler os dados da base de dados

```

```

*
*/
void client_GetMap1Completed(object sender, GetMap1CompletedEventArgs e)
{
    //preenchimento da datagrid

    try
    {
        myItems = e.Result;
        errorState = false;
    }
    catch (Exception error)
    {
        myItems = itemsListError("Dados não encontrados");
    }

    ObservableCollection<mapOrcarub> mainItems = getItems(myItems, 1);
    setDataGrid(mainItems);

    getActivDataGrid().Columns[0].Visibility = Visibility.Visible;
    loadingMapEnd();
}

/*
*
* Handler do evento que acontece quando acaba de ler os orçamentos da base de dados
*
*/
void client_getOrcamentCompleted(object sender, GetOrcamentCompletedEventArgs e)
{
    listOrcam = e.Result;
    setComboBox(listOrcam);
}

/*
*
* Loading dos dados da ComboBox de Orçamento
*
*/
private void setComboBox (ObservableCollection<orcam> items)
{
    cmbOrcament.DisplayMemberPath = "ORCAMENT";
    cmbOrcament.ItemsSource = items;
    cmbOrcament.SelectedIndex = 0;

    orcam o = (orcam)cmbOrcament.SelectedItem;
    string cod = o.CODORCAM;

    updateDataGrid();
    cmbOrcament.SelectionChanged += new
SelectionChangedEventHandler(cb_SelectionChanged);
}

/*
* Handler para a alteração do valor da comboBox do Orçamento
*
*/
void cb_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    setControlsState(false);
    updateDataGrid();
}

/*
* Procedimento de actualização da datagrid, que lê os dados da base de dados tendo
em conta a parametrização executada.
*
*/

```

```

private void updateDataGrid()
{
    setBtnGerarState(false);
    generateState = true;

    getActivDataGrid().Visibility = Visibility.Collapsed;
    dgTotais.Visibility = Visibility.Collapsed;

    orcam o = (orcam)cmbOrcament.SelectedItem;
    string cod = o.CODORCAM;

    DateTime dt1,dt2;
    try
    {
        dt1 = new DateTime((int)cmbFromYear.SelectedItem,
(int)cmbFromMonth.SelectedItem, (int)cmbFromDay.SelectedItem);
        dt2 = new DateTime((int)cmbToYear.SelectedItem,
(int)cmbToMonth.SelectedItem, (int)cmbToDay.SelectedItem);
        if (dt1 > dt2)
        {
            myItems = itemsListError("Erro: Definição de datas incorrecta!");
            setDataGrid(myItems);
        }

        loadingMapBegin(cod, dt1, dt2);

    }
    catch (ArgumentOutOfRangeException e)
    {
        myItems = itemsListError("Erro: Definição de datas incorrecta!");
        setDataGrid(myItems);
    }
}

private void loadingMapBegin(string cod, DateTime dt1, DateTime dt2)
{
    client.GetMap1Async(cod, dt1, dt2);
    progressBar.Visibility = Visibility.Visible;
    progressBar.Value = 0;
    txtProgress.Visibility = Visibility.Visible;
    TimeSpan s = new TimeSpan(0, 0, 3);
    progressBar.Duration=s;
    progressBar.Begin();

    //progressbar.Pause();
}

private void loadingMapEnd()
{
    TimeSpan s = new TimeSpan(0, 0, 3);
    progressBar.Seek(s);
    TimeSpan n = new TimeSpan(0, 0, 10);
    progressBar.Duration = n;
    progressBar.Resume();
    progressBar.Visibility = Visibility.Collapsed;
    txtProgress.Visibility = Visibility.Collapsed;
    RoutedEventArgs e=new RoutedEventArgs();
    if (chkEdit.IsChecked == true)
        chkEdit_Checked(chkEdit, e);
    else
        chkEdit_Unchecked(chkEdit, e);
}

private DataGrid getActivDataGrid()

```

```

{
    if (dgActiv == 1) return dg;
    return dg2;
}

/*
 * Handler para o evento que acontece quando se levanta o left button numa imagem
 *
 */
private void Image_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    Image im = (Image)e.OriginalSource;
    changeImageSource(im, "plus_iconmouseover.png");

    mapOrcarub m = (mapOrcarub) dgActivDataGrid().SelectedItem;

    ObservableCollection<mapOrcarub> childItems = getChildItems(myItems, m);

    if (childItems.Count() > 1)
    {
        setDataGrid(childItems);
        dgActivDataGrid().Columns[0].Visibility = Visibility.Collapsed;
        dgTotais.Columns[0].Visibility = Visibility.Collapsed;
    }

}

/*
 *
 * Procedimento semelhante ao getChildItems, com a diferença que recebe a lista onde
 deve acumular as rubricas filho
 *
 */
private void getChilds(ObservableCollection<mapOrcarub> mainMap,
ObservableCollection<mapOrcarub> newMap, mapOrcarub element)
{
    newMap.Add(element);
    var query = from item in mainMap
                where item.dependen == element.rubrica
                select item;

    if (query.Count() > 0)
        foreach (mapOrcarub e in query)
        {
            getChilds(mainMap, newMap, e);
        }
}

/*
 * Procedimento que retorna os filhos presentes num mapa, dado uma determinada
rubrica
 *
 */
private ObservableCollection<mapOrcarub>
getChildItems(ObservableCollection<mapOrcarub> mainMap, mapOrcarub map)
{
    ltTemp = new ObservableCollection<mapOrcarub>();

    getChilds(mainMap, ltTemp, map);
    return ltTemp;
}

/*
 *
 * Handler para o procedimento de clique no botão Normal
 *
 */
private void btnDefault_Click(object sender, RoutedEventArgs e)

```

```

{
    getActivDataGrid().Columns[0].Visibility = Visibility.Visible;
    ObservableCollection<mapOrcarub> mainItems = getItems(myItems, 1);
    setDataGrid(mainItems);
    if (generateState == false) setGenerateState(false);
    else setGenerateState(true);
}

/*
 *
 * Handler para o evento do clique no botão de detalhes
 *
 */
private void btnAll_Click(object sender, RoutedEventArgs e)
{
    if (generateState == false) setGenerateState(false);
    else setGenerateState(true);

    getActivDataGrid().Columns[0].Visibility = Visibility.Collapsed;
    setDataGrid(myItems);
    dgTotais.Columns[0].Visibility = System.Windows.Visibility.Collapsed;
}

/*
 * Define o estado dos controlos dado um certo estado de Enabled
 *
 */
private void setControlsState(bool state)
{
    cmbOrcament.IsEnabled = state;
    cmbFromDay.IsEnabled = state;
    cmbFromMonth.IsEnabled = state;
    cmbFromYear.IsEnabled = state;
    cmbToDay.IsEnabled = state;
    cmbToMonth.IsEnabled = state;
    cmbToYear.IsEnabled = state;
    btnDefault.IsEnabled = state;
    btnAll.IsEnabled = state;
    btnUpdate.IsEnabled = state;
    btnOpen.IsEnabled = state;
    btnSave.IsEnabled = state;
    btnNewMap.IsEnabled = state;
    btnGraph.IsEnabled = state;
}

/*
 * Coloca um estado erróneo na datagrid com base numa mensagem a ser mostrada ao
utilizador
 *
 */
private ObservableCollection<mapOrcarub> itemsListError(string msg)
{
    ObservableCollection<mapOrcarub> items = new ObservableCollection<mapOrcarub>();
    mapOrcarub m = new mapOrcarub();

    m.descrica = msg;
    m.antecip = "0.00 €";
    m.orcament = "0.00 €";
    m.realiza = "0.00 €";
    m.diforcreal = "0.00 €";

    items.Add(m);
    return items;
}

/*private int getMaxDay(int m, int y)
{
    if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12)
        return 31;
}

```

```

        if (m == 4 || m == 6 || m == 9 || m == 11)
            return 30;
        if (y % 400 == 0)
            return 29;
        if (y % 4 == 0 && y % 100 != 0)
            return 29;
        return 28;
    }*/

    /*private void atualizaDias(char combo,int m, int y)
    {
        int  actualDay, max,newDay;

        max = getMaxDay(m, y);
        if (combo=='f')
            actualDay = (int) cmbFromDay.SelectedItem;
        else
            actualDay = (int) cmbToDay.SelectedItem;

        days = new List<int>();
        for (int i = 1; i < max + 1; i++)
            days.Add(i);

        if (max < actualDay)
            newDay = max;
        else
            newDay = actualDay;

        if (combo == 'f')
        {
            cmbFromDay.ItemsSource = days;
            cmbFromDay.SelectedItem = newDay;
        }
        else
        {
            cmbToDay.ItemsSource = days;
            cmbToDay.SelectedItem = newDay;
        }
    }*/

e) private void cmbFromMonth_SelectionChanged(object sender, SelectionChangedEventArgs
    {
    }

e) private void cmbFromYear_SelectionChanged(object sender, SelectionChangedEventArgs
    {
    }

private void cmbToMonth_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}

private void cmbToYear_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}

/*
 * Handler para o clique no botão de atualizar
 */
*/
private void btnUpdate_Click(object sender, RoutedEventArgs e)
{
    setControlsState(false);
    updateDataGrid();
}

```

```

    }

    private void changeImageSource(Image im, string filename)
    {
        System.Windows.Media.Imaging.BitmapImage bi3 = new
System.Windows.Media.Imaging.BitmapImage();
        // bi3.BeginInit();

        bi3.UriSource = new Uri(filename, UriKind.Relative);
        //bi3.EndInit();
        im.Stretch = Stretch.Fill;
        im.Source = bi3;
        im.UpdateLayout();

    }

    /*
    * Handler para o procedimento de entrada do rato dentro da superfície da imagem
    *
    */
    private void Image_MouseEnter(object sender, MouseEventArgs e)
    {
        Image im = (Image) e.OriginalSource;
        changeImageSource(im, "plus_iconmouseover.png");

    }

    /*
    * Handler para o procedimento de saída do rato dentro da superfície da imagem
    *
    */
    private void Image_MouseLeave(object sender, MouseEventArgs e)
    {
        Image im = (Image)e.OriginalSource;
        changeImageSource(im, "plus_iconnormal.png");

    }

    /*
    *
    * Handler para o procedimento que acontece quando se clica no botão de Abrir
    * Necessita de trabalhar sempre na mesma directoria. Ele lança uma excepção sobre a
    directoria.
    *
    */
    private void btnOpen_Click(object sender, RoutedEventArgs e)
    {
        setControlsState(false);

        OpenFileDialog dlg = new OpenFileDialog();
        dlg.Filter = "Mapa do Controlo de Gestão (*.mcg)|*.mcg";

        dlg.ShowDialog();
        if (dlg.File == null)
            setControlsState(true);
        else
        {
            FileInfo file = dlg.File;
            filename = file.Name;

            client.LoadMap1Async(filename);

        }

    }

    /*
    * Cria um objecto do tipo List a partir duma ObservableCollection
    * Utilizado na gravação do mapa presente
    *
    */
    private ObservableCollection<mapOrcarub> criaLista()

```

```

    {
        ObservableCollection<mapOrcarub> novoMapa = new
ObservableCollection<mapOrcarub>();
        ObservableCollection<mapOrcarub> mTotais = (ObservableCollection<mapOrcarub>)
dgTotais.ItemsSource;
        novoMapa.Add(mTotais[0]);

        foreach (mapOrcarub m in myItems)
            novoMapa.Add(m);
        return novoMapa;
    }

/*
 * btn_SaveClick
 *
 * Handler para o procedimento que é executado quando o botão de gravar é clicado
 *
 */
private void btnSave_Click(object sender, RoutedEventArgs e)
{
    setControlsState(false);
    setDataGridState(false);
    if (generateState) filename = null;
    client.SaveMap1Async(criaLista(), filename);
}

private void Image_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
{
    Image im = (Image)e.OriginalSource;
    changeImageSource(im, "plus_iconcompressed.png");
}

private void btnGerar_Click(object sender, RoutedEventArgs e)
{
    {
        if (generateState)
        {
            setGenerateState(false);
        }
        else
        {
            setGenerateState(true);
        }
    }
}

private void btnNewMap_Click(object sender, RoutedEventArgs e)
{
    {
        if ((string) btnNewMap.Content == "Novo Mapa")
        {
            dgActiv = 2;
            cmbOrcament.SelectedIndex = 1;
            btnNewMap.Content = "Ocultar";
        }
        else
        {
            btnNewMap.Content = "Novo Mapa";
            dg.Visibility = Visibility.Collapsed;
            // datagrid.Begin();
        }
    }
}

private void dg_BeginningEdit(object sender, DataGridBeginningEditEventArgs e)
{
    {
        inEdit = true;
    }
}

private void replaceItemValue(mapOrcarub oldItem, string titulo, mapOrcarub newItem)
{
    {
        if (titulo=="Rubrica - Descrição") oldItem.descrica=newItem.descrica;
        if (titulo=="Orçamentação") oldItem.orcament=newItem.orcament;
        if (titulo=="Realizado") oldItem.realiza=newItem.realiza;
    }
}

```

```

        if (titulo=="Antecipações") oldItem.anticip=newItem.anticip;
        if (titulo == "Dif. Orc. Real.") oldItem.diforcreal = newItem.diforcreal;
    }

    private void updateGroupTree(mapOrcarub map)
    {
        string currency;
        double anticip = 0, orcament = 0, realiza = 0, diforcreal = 0;
        mapOrcarub mainMap;
        if (getNivel(map)>1.0)
        {
            var query = from record in myItems
                        where record.rubrica == map.dependen
                        select record;

            mainMap = (mapOrcarub)query;

            query = from record in myItems
                    where record.dependen == mainMap.rubrica
                    select record;

            List<mapOrcarub> mainSumMap = query.ToList();
            currency = mainSumMap[0].anticip.Substring(mainSumMap[0].anticip.Length -
2);

            foreach (mapOrcarub item in mainSumMap)
            {
                anticip += Double.Parse((string)item.anticip.Substring(0,
item.anticip.Length - 2));
                orcament += Double.Parse((string)item.orcament.Substring(0,
item.orcament.Length - 2));
                realiza += Double.Parse((string)item.realiza.Substring(0,
item.realiza.Length - 2));
                diforcreal += Double.Parse((string)item.diforcreal.Substring(0,
item.diforcreal.Length - 2));
            }

            mainMap.anticip = formatFloatValue(anticip) + currency;
            mainMap.orcament = formatFloatValue(orcament) + currency;
            mainMap.realiza = formatFloatValue(realiza) + currency;
            mainMap.diforcreal = formatFloatValue(diforcreal) + currency;

            updateGroupTree(mainMap);
        }
    }

    private void dg_CurrentCellChanged(object sender, EventArgs e)
    {
        if (inEdit == true)
        {
            DataGridViewColumn d=getActivDataGrid().CurrentColumn;
            string titulo=(string) d.Header;
            mapOrcarub newItem = (mapOrcarub) getActivDataGrid().SelectedItem;
            mapOrcarub oldItem = new mapOrcarub();
            foreach (mapOrcarub item in myItems)
            {
                if (item.rubrica == newItem.rubrica)
                {
                    oldItem=item;
                    break;
                }
            }
            replaceItemValue(oldItem, titulo, newItem);

            getActivDataGrid().IsEnabled = false;
        }
    }

```

```

    }
}

private void dg_IsEnabledChanged(object sender, DependencyPropertyChangedEventArgs
e)
{
    ObservableCollection<mapOrcarub> presentItems =
(ObservableCollection<mapOrcarub>)getActivDataGrid().ItemsSource;
    ObservableCollection<mapOrcarub> newPresentItems = new
ObservableCollection<mapOrcarub>();

    foreach (mapOrcarub item in presentItems)
        newPresentItems.Add(item);

    setDataGrid(newPresentItems);

    if (inEdit)
        getActivDataGrid().IsEnabled=true;

    inEdit = false;
}

private void chkEdit_Checked(object sender, RoutedEventArgs e)
{
    getActivDataGrid().IsReadOnly = false;
}

private void chkEdit_Unchecked(object sender, RoutedEventArgs e)
{
    getActivDataGrid().IsReadOnly = true;
}

private void progressBar_ValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)
{
    int valor = (int) progressBar.Value;
    string s = Convert.ToString(valor) + " %";
    txtProgress.Text = s;
}

private void btnGraph_Click(object sender, RoutedEventArgs e)
{
    if (chart.Visibility == Visibility.Collapsed)
    {
        btnGraph.Content = "Voltar";
        setControlsState(false);

        dgTotais.Visibility = Visibility.Collapsed;

        animDGfade.Begin();
        int nDados;
        double d;

        List<string> rub = new List<string>();
        List<int> orc = new List<int>();

        foreach (mapOrcarub m in getActivDataGrid().ItemsSource)
        {
            rub.Add(m.rubrica);
            d = Double.Parse((string)m.orcament.Substring(0, m.orcament.Length -
2));
            orc.Add(Convert.ToInt32(d));
        }

        nDados = rub.Count();
    }
}

```

```

KeyValuePair<string, int>[] graphData = new KeyValuePair<string,
int>[nDados];

    for (int i = 0; i < nDados; i++)
        graphData[i] = new KeyValuePair<string, int>(rub[i], orc[i]);

    ((BarSeries)chart.Series[0]).ItemsSource = graphData;
}
else
{
    setControlsState(false);
    btnGraph.Content = "Gráfico";
    animChartBarFade.Begin();
}

}

}

/* Validação das Datas
*
* A rever
*
*
* private void cmbFromMonth_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (cmbFromYear.SelectedItem != null && cmbFromMonth.SelectedItem != null)
        atualizaDias('f', (int) cmbFromMonth.SelectedItem, (int)
cmbFromYear.SelectedItem);
}

    private void cmbFromYear_SelectionChanged(object sender, SelectionChangedEventArgs
e)
{
    if (cmbFromYear.SelectedItem != null && cmbFromMonth.SelectedItem != null)
        atualizaDias('f', (int) cmbFromMonth.SelectedItem,
(int) cmbFromYear.SelectedItem);
}

    private void cmbToMonth_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (cmbToYear.SelectedItem != null && cmbToMonth.SelectedItem != null)
        atualizaDias('t', (int) cmbToMonth.SelectedItem,
(int) cmbToYear.SelectedItem);
}

    private void cmbToYear_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (cmbToYear.SelectedItem != null && cmbToMonth.SelectedItem != null)
        atualizaDias('t', (int) cmbToMonth.SelectedItem,
(int) cmbToYear.SelectedItem);
}

*/

}
}

```

F) Modelo de Classes LINQtoSQL para utilização em Mapas

```
<?xml version="1.0" encoding="utf-8"?>
<Database Name="QUIDTEMP_QCSL" Class="TEMPDataClassesDataContext"
xmlns="http://schemas.microsoft.com/linqtosql/dbml/2007">
  <Connection Mode="WebSettings" ConnectionString="Data Source=ZODIAC;Initial
Catalog=QUIDTEMP_QCSL;User ID=quidgest;Pooling=False"
SettingsObjectName="System.Configuration.ConfigurationManager.ConnectionStrings"
SettingsPropertyName="QUIDTEMP_QCSLConnectionString" Provider="System.Data.SqlClient" />
  <Table Name="dbo.M1_TMPRPT" Member="M1_TMPRPTs">
    <Type Name="M1_TMPRPT">
      <Column Name="codcleco" Type="System.String" DbType="VarChar(6) NOT NULL"
CanBeNull="false" />
      <Column Name="rubrica" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
      <Column Name="descrica" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
      <Column Name="dependen" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
      <Column Name="nivel" Type="System.Double" DbType="Float" CanBeNull="true" />
      <Column Name="fixo" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="variavel" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="tpcust" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
      <Column Name="clecor" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="classdes" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="classrec" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="orcament" Type="System.Double" DbType="Float" CanBeNull="true" />
      <Column Name="realiza" Type="System.Double" DbType="Float" CanBeNull="true" />
      <Column Name="antecip" Type="System.Double" DbType="Float" CanBeNull="true" />
    </Type>
  </Table>
  <Table Name="dbo.mapOrcarub" Member="mapOrcarubs">
    <Type Name="mapOrcarub">
      <Column Name="codigo" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="descrica" Type="System.String" DbType="VarChar(200)" CanBeNull="true" />
      <Column Name="orcament" Type="System.String" DbType="varchar(200)" CanBeNull="true" />
      <Column Name="realiza" Type="System.String" DbType="varchar(200)" CanBeNull="true" />
      <Column Name="antecip" Type="System.String" DbType="varchar(200)" CanBeNull="true" />
      <Column Name="diforcreal" Type="System.String" DbType="varchar(200)" CanBeNull="true"
/>
      <Column Name="isDespesaReceita" Type="System.Int32" CanBeNull="false" />
      <Column Member="rubrica" Type="System.String" CanBeNull="false" />
      <Column Member="dependen" Type="System.String" CanBeNull="false" />
    </Type>
  </Table>
</Database>
```

H) Modelo de Classes LINQtoSQL para leitura de Dados

```
<?xml version="1.0" encoding="utf-8"?>
<Database Name="CNTCES22009" Class="DataClasses1DataContext" Serialization="Unidirectional"
xmlns="http://schemas.microsoft.com/linqtosql/dbml/2007">
  <Connection Mode="WebSettings" ConnectionString="Data Source=ZODIAC;Initial
Catalog=CNTCES22009;Persist Security Info=True;User ID=quidgest"
SettingsObjectName="System.Configuration.ConfigurationManager.ConnectionStrings"
SettingsPropertyName="CNTCES22009ConnectionString" Provider="System.Data.SqlClient" />
  <Table Name="dbo.CNTorcam" Member="orcams">
    <Type Name="orcams">
      <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
      <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
      <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
      <Column Name="ORCAMENT" Type="System.String" DbType="VarChar(75)" CanBeNull="true" />
      <Column Name="SIGLA" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
      <Column Name="VALCOM" Type="System.String" DbType="VarChar(1)" CanBeNull="true" />
      <Column Name="VALPRO" Type="System.String" DbType="VarChar(1)" CanBeNull="true" />
      <Column Name="VALIMPED" Type="System.Int32" DbType="Int" CanBeNull="true" />
      <Column Name="ORGANISM" Type="System.String" DbType="VarChar(100)" CanBeNull="true" />
      <Column Name="REGIME" Type="System.String" DbType="VarChar(1)" CanBeNull="true" />
    </Type>
  </Table>
```

```

    <Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="SELNIVEL" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="orcam_clorc" Member="clorcs" ThisKey="CODORCAM" OtherKey="CODORCAM"
Type="clorc" />
  </Type>
</Table>
<Table Name="dbo.CNTcload" Member="cloads">
  <Type Name="cload">
    <Column Name="CODCLOAD" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="LNHCCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANO" Type="System.String" DbType="VarChar(4)" CanBeNull="true" />
    <Column Name="PERIODO" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
    <Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="DOTMENSL" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DESCRICA" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="INSCRICA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULACOE" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="REFORCOS" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="SOMALCC" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULAJU" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="RESFORAJ" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULTOT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="RESFTOT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DOTAJUSM" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="COMPROME" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PROCESSA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DOTDISP" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="RECEBIDO" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PORRECE" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="cload_lrece" Member="lrecs" ThisKey="CODCLOAD" OtherKey="CODCLOAD"
Type="lrece" />
    <Association Name="cload_compr" Member="comprs" ThisKey="CODCLOAD" OtherKey="CODCLOAD"
Type="compr" />
    <Association Name="cload_lreca" Member="lrecas" ThisKey="CODCLOAD" OtherKey="CODCLOAD"
Type="lreca" />
    <Association Name="cloac_cload" Member="cloac" ThisKey="CODCLOAD" OtherKey="CODCLOAD"
Type="cloac" IsForeignKey="true" />
  </Type>
</Table>
<Table Name="dbo.CNTcload" Member="cloads">
  <Type Name="cload">
    <Column Name="CODCLOAD" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="CODACTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ACTIVID" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="DOTINICI" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="RUBRICA" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
    <Column Name="REFORCOS" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULACAO" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="INSCRICA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="REFANUL" Type="System.Double" DbType="Float" CanBeNull="true" />

```

```

<Column Name="TOTAJUST" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DOTAJUST" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PROCESSA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="REPOSICO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DOTACDIS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VERBADIS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RECEITAS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RECEBIDO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="SALDOREC" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="DOTANUAL" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Association Name="cloac_cloam" Member="cloams" ThisKey="CODCLOAC" OtherKey="CODCLOAC"
Type="cloam" />
<Association Name="clorc_cloac" Member="clorc" ThisKey="CODCLORC" OtherKey="CODCLORC"
Type="clorc" IsForeignKey="true" />
</Type>
</Table>
<Table Name="dbo.CNTcleco" Member="clecos">
<Type Name="cleco">
<Column Name="CODCLECO" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
<Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODIGO" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
<Column Name="DEPENDEN" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
<Column Name="MOVIMENT" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="IFMOVIME" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RUBRICA" Type="System.String" DbType="VarChar(25)" CanBeNull="true" />
<Column Name="NIVEL" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCRICA" Type="System.String" DbType="VarChar(90)" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="SUJIRS" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="CLASSDES" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="CLASSREC" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="FIXO" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="VARIAVEL" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="TPCUST" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
<Column Name="CLECOR" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Association Name="cleco_clorc" Member="clorcs" ThisKey="CODCLECO" OtherKey="CODCLECO"
Type="clorc" />
<Association Name="cleco_procs" Member="procs" ThisKey="CODCLECO" OtherKey="CODCLECO"
Type="procs" />
</Type>
</Table>
<Table Name="dbo.CNTclorc" Member="clorcs">
<Type Name="clorc">
<Column Name="CODCLORC" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
<Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
<Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="RUBRICA" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
<Column Name="NAODUPLI" Type="System.String" DbType="VarChar(12)" CanBeNull="true" />
<Column Name="DOTINICI" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="REFORCOS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ANULACAO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="INSCRICA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="REFANUL" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DOTAJUST" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PROCESSA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PAGO" Type="System.Double" DbType="Float" CanBeNull="true" />

```

```

<Column Name="REPOSICO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DOTACDIS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VERBADIS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RECEITAS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RECEBIDO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="SALDOREC" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="COMPROME" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="CLASREC" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="CLOAMSR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Association Name="clorc_cloac" Member="cloacs" ThisKey="CODCLORC" OtherKey="CODCLORC"
Type="cloac" />
<Association Name="orcamlorc" Member="orcaml" ThisKey="CODORCAM" OtherKey="CODORCAM"
Type="orcaml" IsForeignKey="true" />
<Association Name="cleco_cloac" Member="cleco" ThisKey="CODCLECO" OtherKey="CODCLECO"
Type="cleco" IsForeignKey="true" />
</Table>
<Table Name="dbo.CNTlcp" Member="lcp">
  <Type Name="lcp">
    <Column Name="CODLCP" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODTCCUS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCCUST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPROCS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCOMPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="IFPTI" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="SIGLA" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="CODIGO" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
    <Column Name="ESAUTOCB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOOPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOOCM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="PERCENTA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PERCREAL" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="PAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="ADEBCRED" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="NUMPROCS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="procs_lcp" Member="procs" ThisKey="CODPROCS" OtherKey="CODPROCS"
Type="procs" IsForeignKey="true" />
  </Type>
</Table>
<Table Name="dbo.CNTprocs" Member="procs">
  <Type Name="procs">
    <Column Name="CODPROCS" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCOMP" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCOMPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />

```

```
<Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODAFFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFOLHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODAUTOR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFVENC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="NUMFOLHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="DATAPAGA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="CODAGENC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="TIPOCONT" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="CODHMBKG" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="RUBRICA" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
<Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
<Column Name="CODMAGIC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CONTAPOC" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
<Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
<Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODACTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ACTIVID" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
<Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFACTU" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="NUMFACTU" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAFACT" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="NUMAUTOR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="DTAUTOR" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="NOMEFORN" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
<Column Name="CODFIVA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODARTDE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFMANE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="SEFLHIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="CODDIARO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="LNHCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="NAODUPLI" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
<Column Name="NUMPROCS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="DATAPROC" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="ASSUNTO" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
<Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="AIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="IVADEDUT" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="MERCORIG" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="REGIAO" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="IVAADQUI" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="PDESCONT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCOFIN" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCOCOM" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCONTO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PROC_IVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="IVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="IVA2" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALIVAD" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALIVAS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ENTREIVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="VALORLIQ" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PROCESSA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALORFIN" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RETENCAO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="REGULPRO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRAZOPAG" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRAZODIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="REPOSABT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="NCREDABT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PAGML" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="SEPAPO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PORPAGAR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="EMATRASO" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="RETEPAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="TOTPAGO" Type="System.Double" DbType="Float" CanBeNull="true" />
```

```

<Column Name="REPIVAD" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATAACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="SEAUTOR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="NUMCOMPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="VALIDADR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="CODCDIVA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLOM2" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFFPROC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ZIDOTAXA" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="CONDESC" Type="System.String" DbType="VarChar(60)" CanBeNull="true" />
<Column Name="PORCLASF" Type="System.Double" DbType="Float" CanBeNull="true" />
<Association Name="procs_lcpro" Member="lcpros" ThisKey="CODPROCS" OtherKey="CODPROCS"
Type="lcpro" />
<Association Name="procs_repos" Member="repos" ThisKey="CODPROCS" OtherKey="CODPROCS"
Type="repos" />
<Association Name="cleco_procs" Member="cleco" ThisKey="CODCLECO" OtherKey="CODCLECO"
Type="cleco" IsForeignKey="true" />
</Type>
</Table>
<Table Name="dbo.CNTlcrc" Member="lcrcs">
  <Type Name="lcrc">
    <Column Name="CODLCREC" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODTCCUS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCCUST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODLRECE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODRECEI" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="IFPTI" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="SIGLA" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="CODIGO" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
    <Column Name="ESAUTOCB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOCM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="PERCENTA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PERCREAL" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="DATAACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="ADEBCRED" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="TIPOAFEC" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
    <Column Name="RECEIBID" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULADA" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="lrece_lcrc" Member="lrece" ThisKey="CODLRECE" OtherKey="CODLRECE"
Type="lrece" IsForeignKey="true" />
  </Type>
</Table>
<Table Name="dbo.CNTlrece" Member="lrecs">
  <Type Name="lrece">
    <Column Name="CODLRECE" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODRECEI" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
  </Type>
</Table>

```

```

<Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODIVARC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODMAGIC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
<Column Name="CODACTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
<Column Name="ACTIVID" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
<Column Name="CONTAPOC" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
<Column Name="CODFOLHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="NOMECLI" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
<Column Name="CODARTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="DESCARTI" Type="System.String" DbType="VarChar(100)" CanBeNull="true" />
<Column Name="CODMOEDA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="LNHCCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="GUIA" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
<Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="TIPORECE" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="POSICAO" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
<Column Name="RUBRICA" Type="System.String" DbType="VarChar(25)" CanBeNull="true" />
<Column Name="DESCRICA" Type="System.String" DbType="VarChar(100)" CanBeNull="true" />
<Column Name="QTD" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCRRTF" Type="System.String" DbType="VarChar(8000)" CanBeNull="true" />
/>

<Column Name="UNIDADE" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
<Column Name="AIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRUNITCT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRUNIT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRECOIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ISENTIVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="DIRDEDUT" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="VALORILI" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PDESCONT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DESCOEME" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRECIVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="INCIDENC" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="INCIDENE" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RETECAO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="SERECEBI" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RECEBIDO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PORRECEB" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RESTITUI" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="ANULADA" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="COTACAO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALDIVIS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="RESTIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="PERCENTA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALRESER" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="MONTADE" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VNDLNFAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="VALCOMIV" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="CODLRECA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="PRUNITC" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRUNITDF" Type="System.Double" DbType="Float" CanBeNull="true" />
<Association Name="lrece_lcrec" Member="lcrecs" ThisKey="CODLRECE" OtherKey="CODLRECE"
Type="lcrec" />
<Association Name="lrece_resti" Member="restis" ThisKey="CODLRECE" OtherKey="CODLRECE"
Type="resti" />
<Association Name="cloam_lrece" Member="cloam" ThisKey="CODCLOAM" OtherKey="CODCLOAM"
Type="cloam" IsForeignKey="true" />
</Type>

```

```

</Table>
<Table Name="dbo.CNTrestis" Member="restis">
  <Type Name="restis">
    <Column Name="CODRESTI" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODGREST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODLRECE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="CODFOLHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFIVA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODMAGIC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="LNHCCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PARAREG" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="TIPONTCR" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="SEFLHIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATAREST" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALORPAG" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALORIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALORABT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="OBS" Type="System.String" DbType="VarChar(50)" CanBeNull="true" />
    <Column Name="DATAACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="resti_lcrst" Member="lcrsts" ThisKey="CODRESTI" OtherKey="CODRESTI"
Type="lcrst" />
    <Association Name="lrece_resti" Member="lrece" ThisKey="CODLRECE" OtherKey="CODLRECE"
Type="lrece" IsForeignKey="true" />
  </Type>
</Table>
<Table Name="dbo.CNTrepos" Member="repos">
  <Type Name="repos">
    <Column Name="CODREPOS" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODGREPO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPROCS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCOMPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="CODFOLHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFIVA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODMAGIC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="SEFLHIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="CODHMBKG" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="DATAECE" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="LNHCCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="TIPODOC" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="TIPONTCR" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="AIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="IVADEDUT" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="MERCORIG" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="REGIAO" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
  </Type>
</Table>

```

```

<Column Name="PARAREG" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="ABATPROC" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="VALABPRO" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="NCVALABP" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALPRTOT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALORECE" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALORABT" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALORIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALIVAD" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALIVAS" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="OBS" Type="System.String" DbType="VarChar(60)" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="PERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Association Name="repos_lcrpo" Member="lcrpos" ThisKey="CODREPOS" OtherKey="CODREPOS"
Type="lcrpo" />
  <Association Name="procs_repos" Member="procs" ThisKey="CODPROCS" OtherKey="CODPROCS"
Type="procs" IsForeignKey="true" />
</Type>
</Table>
<Table Name="dbo.CNTlcrpo" Member="lcrpos">
  <Type Name="lcrpo">
    <Column Name="CODLCRPO" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODTCCUS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCCUST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODREPOS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="IFPTI" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="SIGLA" Type="System.String" DbType="VarChar(25)" CanBeNull="true" />
    <Column Name="CODIGO" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
    <Column Name="ESAUTOCB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOOPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTOOCM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ESAUTORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="PERCENTA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PERCREAL" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="PERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="ADEBCRED" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Association Name="repos_lcrpo" Member="repos" ThisKey="CODREPOS" OtherKey="CODREPOS"
Type="repos" IsForeignKey="true" />
  </Type>
</Table>
<Table Name="dbo.CNTlcrst" Member="lcrsts">
  <Type Name="lcrst">
    <Column Name="CODLCRST" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODTCCUS" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCCUST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODRESTI" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODGREST" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />

```

```

<Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
<Column Name="IFPTI" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="SIGLA" Type="System.String" DbType="VarChar(25)" CanBeNull="true" />
<Column Name="CODIGO" Type="System.String" DbType="VarChar(99)" CanBeNull="true" />
<Column Name="ESAUTOCB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ESAUTOPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ESAUTOCM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="ESAUTORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Column Name="PERCENTA" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PERCREAL" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="VALOR" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
<Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
<Column Name="ADEBCRED" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Association Name="resti_lcrst" Member="resti" ThisKey="CODRESTI" OtherKey="CODRESTI"
Type="resti" IsForeignKey="true" />
</Type>
</Table>
<Table Name="dbo.CNTcompr" Member="comprs">
  <Type Name="compr">
    <Column Name="CODCOMPR" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFVENC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCOMP" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODARTDE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODIGOPR" Type="System.String" DbType="VarChar(13)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODACTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="NOMEFORN" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
    <Column Name="LNHCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="NUMCOMPR" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="NAODUPLI" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
    <Column Name="NUMLINHA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="DATACOMP" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="DESCRICA" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
    <Column Name="PORVALID" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="NUMART" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PRECOIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALINIPR" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALORINI" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALACOMP" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALORFIN" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PROCESSA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="IVADEDUT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="NAOPROCS" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PORPROCS" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="REPOSABT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="ATIVID" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
    <Column Name="RUBRICA" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
    <Column Name="CIBREQUI" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="NCREDABT" Type="System.Double" DbType="Float" CanBeNull="true" />
  </Type>
</Table>

```

```

<Column Name="SETEMART" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="PARACONT" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
<Column Name="ADJUDICA" Type="System.String" DbType="VarChar(13)" CanBeNull="true" />
<Column Name="PORCLASF" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="PRECOUNI" Type="System.Double" DbType="Float" CanBeNull="true" />
<Association Name="cloam_compr" Member="cloam" ThisKey="CODCLOAM" OtherKey="CODCLOAM"
Type="cloam" IsForeignKey="true" />
</Type>
</Table>
<Table Name="dbo.CNTlreca" Member="lrecas">
  <Type Name="lreca">
    <Column Name="CODLRECA" Type="System.String" DbType="VarChar(6) NOT NULL"
IsPrimaryKey="true" CanBeNull="false" />
    <Column Name="CODORGAN" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODANO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODAFCC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODESTAB" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLORC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODORCAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODMAGIC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLECO" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAC" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODCLOAM" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="ORCAMENT" Type="System.String" DbType="VarChar(15)" CanBeNull="true" />
    <Column Name="CODACTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="CODPOCP" Type="System.String" DbType="VarChar(7)" CanBeNull="true" />
    <Column Name="ACTIVID" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
    <Column Name="CONTAPOC" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
    <Column Name="CODFORNE" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="NOMECLI" Type="System.String" DbType="VarChar(80)" CanBeNull="true" />
    <Column Name="CODARTIV" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="DESCARTI" Type="System.String" DbType="VarChar(100)" CanBeNull="true" />
    <Column Name="CODMOEDA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="LNHCUST" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="NAODUPLI" Type="System.String" DbType="VarChar(18)" CanBeNull="true" />
    <Column Name="GUIA" Type="System.String" DbType="VarChar(10)" CanBeNull="true" />
    <Column Name="DATA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="TIPORECE" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="RUBRICA" Type="System.String" DbType="VarChar(25)" CanBeNull="true" />
    <Column Name="DESCRICA" Type="System.String" DbType="VarChar(100)" CanBeNull="true" />
    <Column Name="QTD" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DESCRRTF" Type="System.String" DbType="VarChar(8000)" CanBeNull="true"
/>

    <Column Name="UNIDADE" Type="System.String" DbType="VarChar(2)" CanBeNull="true" />
    <Column Name="AIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PRUNITCT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PRUNIT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PRECOIVA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ISENTIVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="DIRDEDUT" Type="System.Char" DbType="VarChar(1)" CanBeNull="true" />
    <Column Name="VALORILI" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PDESCONT" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DESCOEME" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="PRECIVA" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="INCIDENC" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="INCIDENE" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ANULADA" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="COTACAO" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="VALDIVIS" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="DATACRIA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERCRIA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="DATAMUDA" Type="System.DateTime" DbType="DateTime" CanBeNull="true" />
    <Column Name="OPERMUDA" Type="System.String" DbType="VarChar(20)" CanBeNull="true" />
    <Column Name="VALORFIN" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="ZZSTATE" Type="System.Int32" DbType="Int" CanBeNull="true" />
    <Column Name="CODRECA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
    <Column Name="NUMLINHA" Type="System.String" DbType="VarChar(5)" CanBeNull="true" />
    <Column Name="ARECA" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="RECEITAD" Type="System.Double" DbType="Float" CanBeNull="true" />
    <Column Name="NAORECEB" Type="System.Double" DbType="Float" CanBeNull="true" />

```

```
<Column Name="PORRECEB" Type="System.Double" DbType="Float" CanBeNull="true" />
<Column Name="CIBLNFA" Type="System.String" DbType="VarChar(6)" CanBeNull="true" />
<Association Name="cloam_lreca" Member="cloam" ThisKey="CODCLOAM" OtherKey="CODCLOAM"
Type="cloam" IsForeignKey="true" />
</Type>
</Table>
</Database>
```

I) Interface do Web Service

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace linqwitheverything.Web
{
    // NOTE: If you change the interface name "IService1" here, you must also update the
    // reference to "IService1" in Web.config.
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        List<mapOrcarub> GetMap1(string codorcam, DateTime dt1, DateTime dt2);

        [OperationContract]
        List<orcament> GetOrcament();

        [OperationContract]
        void SaveMap1(List<mapOrcarub> mapa, string filename);

        [OperationContract]
        List<mapOrcarub> LoadMap1(string filename);
    }
}
```

J) Implementação do Web Service

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

using System.IO;

using System.Web;
```

```
namespace linqwitheverything.Web
```

```
{

    // NOTE: If you change the class name "Service1" here, you must also update the
    reference to "Service1" in Web.config.

    public class Service1 : IService1
    {
        #region IService1 Members

        private DataClasses1DataContext db = new DataClasses1DataContext();

        private TEMPDataClassesDataContext db_TMPRPT = new
TEMPDataClassesDataContext();

        private List<M1_TMPRPT> lt, ltTemp;

        private string pathname = @"C:\Documents and Settings\lmatias\Desktop\mapas\";

        //configurações

        // protected string codorcam = "      1"; //Orçamento

        // protected DateTime dt1 = new DateTime(2000, 1, 1); //data1

        // protected DateTime dt2 = new DateTime(2009, 12, 31); //data2

        protected bool ifivaded = false; //iva dedutível

        protected string currency = "€"; //caracter da unidade monetária

        protected int numEspacos = 4; //Numero de espaços de indentação das rubricas


        public Service1()
        {
            //codorcam = "      1"; //Orçamento

            //dt1 = new DateTime(2000, 1, 1); //data1

            //dt2 = new DateTime(2009, 12, 31); //data2

            ifivaded = false; //iva dedutível

            currency = "€"; //caracter da unidade monetária

            numEspacos = 4; //Numero de espaços de indentação das rubricas

        }

        private void adicionaFilhos(M1_TMPRPT cleco)
        {
            ltTemp.Add(cleco);
```

```
string rubrica = cleco.rubrica;

var query = from record in lt
            where record.dependen == rubrica
            select record;

query = query.OrderBy(q => q.rubrica);

foreach (var item in query)
    adicionaFilhos(item);
}

private void somaSaldos(double nivel)
{
    if (nivel > 0.0)
    {
        var query = from record in lt
                    where record.nivel == nivel && record.orcament== -1.0
                    select record;

        string rubrica, codcleco;

        foreach (var record in query)
        {
            rubrica = record.rubrica;
            codcleco = record.codcleco;

            var saldos = from item in lt
                        where item.dependen.CompareTo(rubrica) == 0
                        group item by item.dependen into s
                        select new
                        {
                            //TotalUnitsInStock = g.Sum(p => p.UnitsInStock)
                            CODCLECO = s.Key,
                            TOTALDOT = s.Sum(item => item.orcament),
                            TOTALREA = s.Sum(item => item.realiza),
                            TOTALANT = s.Sum(item => item.antecip)
                        }
                    }
            }
    }
```

```
};

var saldo = saldos.First();

record.antecip = saldo.TOTALANT;

record.realiza = saldo.TOTALREA;

record.orcament = saldo.TOTALDOT;

}

somaSaldos(nivel - 1.0);

}

}

private bool containsCleco(string codcleco)
{
    foreach(var item in lt)
        if (codcleco.CompareTo(item.codcleco)==0) return true;

    return false;
}

//adicionaPai(c, nivel,lt);

private void adicionaPai(string codcleco, int nivel)
{
    if (nivel > 1)
    {
        var query = from cleco in db.clecos
                     where cleco.CODCLECO==codcleco
                     select cleco;

        var cl = query.First();

        string rubrica = cl.DEPENDEN;

        query = from cleco in db.clecos
                 where cleco.RUBRICA.CompareTo(rubrica) == 0
                 select cleco;
```

```
cl = query.First();

M1_TMPRPT m = new M1_TMPRPT();
m.codcleco = cl.CODCLECO;
if (!containsCleco(m.codcleco))
{
    m.antecip = -1.0;
    m.classdes = cl.CLASSDES;
    m.classrec = cl.CLASSREC;
    m.clecor = cl.CLECOR;
    m.dependen = cl.DEPENDEN;
    m.descrica = cl.DESCRICA;
    m.fixo = cl.FIXO;
    m.nivel = cl.NIVEL;
    m.orcament = -1.0;
    m.realiza = -1.0;
    m.rubrica = cl.RUBRICA;
    m.tpcust = cl.TPCUST;
    m.variavel = cl.VARIAVEL;

    lt.Add(m);
    adicionaPai(m.codcleco, nivel - 1);
}
}

private string formatFloatValue(double d)
{
    string s = Math.Round(d, 2).ToString();
    if (s.Contains(',') == false) s += ",00";
    return s;
}

public List<orcament> GetOrcament()
```

```
{
    var query = from record in db.orcams
                select record;
    return query.ToList();
}

public List<mapOrcarub> GetMap1(string codorcam, DateTime dt1, DateTime dt2)
{
    M1_TMPRPT it;

    //Dados de entrada

    // var query = from it in db.CNTorgans
    //             where 1==1
    //             select it;

    //group p by p.Category into g

    //Dotações mensais existentes
    var dotAjusm =
        from cloam in db.cloams
        join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC
        join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC
        join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM
        join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO
        where 1==1 && cloam.DATA<=dt2 && cloam.DATA>=dt1 &&
orcam.CODORCAM==codorcam
        select new
        {
            // { OrderID = o.OrderID, Cost = o.Cost };
            CODCLECO = cleco.CODCLECO,
            ORCAMENT = cloam.DOTAJUSM
        };
}
```

```
var dotAjusmSum =

    from d in dotAjusm

    group d by d.CODCLECO into s

    select new

    {

        //TotalUnitsInStock = g.Sum(p => p.UnitsInStock)

        CODCLECO = s.Key,

        TOTALDOT = s.Sum(d => d.ORDCUMENT)

    };


//Cálculo do realizado

var realizLcpro=

    from lcpro in db.lcpros

    join procs in db.procs on lcpro.CODPROCS equals procs.CODPROCS

    join cleco in db.clecos on procs.CODCLECO equals cleco.CODCLECO

    join clorc in db.clorcs on cleco.CODCLECO equals clorc.CODCLECO

    //join cloac in db.cloacs on clorc.CODCLORC equals cloac.CODCLORC

    //join cloam in db.cloams on cloac.CODCLOAC equals cloam.CODCLOAC

    join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

    where 1 == 1 && lcpro.DATA <= dt2 && lcpro.DATA >= dt1 &&

orcam.CODORCAM == codorcam && cleco.CLASSEDES == 1

    select new

    {

        CODCLECO = cleco.CODCLECO,

        REALIZA = ifivaded == true ? lcpro.VALOR - (lcpro.PERCENTA / 100 *

procs.VALIDADR) : lcpro.VALOR

    };


var realizLcrec =

    from lcrec in db.lcrecs

    join lrece in db.lreces on lcrec.CODLRECE equals lrece.CODLRECE

    join cloam in db.cloams on lrece.CODCLOAM equals cloam.CODCLOAM
```

```

join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC
join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC
join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO
join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM
join resti in db.restis on lrece.CODLRECE equals resti.CODLRECE

where 1 == 1 && lcrec.DATA <= dt2 && lcrec.DATA >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSREC == 1 && lcrec.ANULADA == 0

select new
{
    CODCLECO = cleco.CODCLECO,

    REALIZA = ifivaded == true ? lcrec.VALOR + lrece.VALCOMIV +
resti.VALOR : lcrec.VALOR + lrece.INCIDENC + resti.VALOR + resti.VALORIVA
};

var realizLcrpo =
    from lcrpo in db.lcrpos
    join repos in db.repos on lcrpo.CODREPOS equals repos.CODREPOS
    join procs in db.procs on repos.CODPROCS equals procs.CODPROCS
    join cleco in db.clecos on procs.CODCLECO equals cleco.CODCLECO
    join clorc in db.clorcs on cleco.CODCLECO equals clorc.CODCLECO
    //join cloac in db.cloacs on clorc.CODCLORC equals cloac.CODCLORC
    // join cloam in db.cloams on cloac.CODCLOAC equals cloam.CODCLOAC
    join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

    where 1 == 1 && lcrpo.DATA <= dt2 && lcrpo.DATA >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSEDES == 1

    select new
    {
        CODCLECO = cleco.CODCLECO,

        REALIZA = lcrpo.VALOR * -1
    };

var realizLcrst =
    from lcrst in db.lcrsts
    join resti in db.restis on lcrst.CODRESTI equals resti.CODRESTI
    join lrece in db.lreces on resti.CODLRECE equals lrece.CODLRECE
    join cloam in db.cloams on lrece.CODCLOAM equals cloam.CODCLOAM

```

```
join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC
join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC
join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO
join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

where 1 == 1 && lcrst.DATA <= dt2 && lcrst.DATA >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSREC == 1

select new
{
    CODCLECO = cleco.CODCLECO,
    REALIZA = lcrst.VALOR * -1
};
```

```
//agrupamento de todos os realizados
```

```
var realiz = realizLcrec.Concat(realizLcpro);
realiz = realiz.Concat(realizLcrpo);
realiz = realiz.Concat(realizLcrst);
```

```
var realizSum =

from r in realiz

group r by r.CODCLECO into s

select new
{
    CODCLECO = s.Key,
    TOTALREALIZA = s.Sum(r => r.REALIZA)
};
```

```
//Cálculo da antecipação
```

```
var antecipCompr =

from compr in db.comprs

join cloam in db.cloams on compr.CODCLOAM equals cloam.CODCLOAM
join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC
join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC
```

```

join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO

join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

where 1 == 1 && compr.DATACOMP <= dt2 && compr.DATACOMP >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSDES == 1

select new
{
    CODCLECO = cleco.CODCLECO,
    ANTECIP = compr.VALORFIN
};

var antecipRepos =

from repos in db.repos

join procs in db.procs on repos.CODPROCS equals procs.CODPROCS

join cleco in db.clecos on procs.CODCLECO equals cleco.CODCLECO

join clorc in db.clorcs on cleco.CODCLECO equals clorc.CODCLECO

// join cloac in db.cloacs on clorc.CODCLORC equals cloac.CODCLORC
// join cloam in db.cloams on cloac.CODCLOAC equals cloam.CODCLOAC

join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

where 1 == 1 && repos.DATA <= dt2 && repos.DATA >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSDES == 1

select new
{
    CODCLECO = cleco.CODCLECO,
    ANTECIP = (ifivaded == true ? repos.NCVALABP : repos.NCVALABP -
repos.IVADEDUT) * -1
};

var antecipLreca =

from lreca in db.lrecas

join cloam in db.cloams on lreca.CODCLOAM equals cloam.CODCLOAM

join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC

join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC

join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO

join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM

where 1 == 1 && lreca.DATA <= dt2 && lreca.DATA >= dt1 &&
orcam.CODORCAM == codorcam && cleco.CLASSREC == 1

```

```

select new
{
    CODCLECO = cleco.CODCLECO,
    ANTECIP = lreca.VALORFIN
};

var anticipResti =
    from resti in db.restis
    join lreca in db.lreces on resti.CODLRECE equals lreca.CODLRECE
    join cloam in db.cloams on lreca.CODCLOAM equals cloam.CODCLOAM
    join cloac in db.cloacs on cloam.CODCLOAC equals cloac.CODCLOAC
    join clorc in db.clorcs on cloac.CODCLORC equals clorc.CODCLORC
    join cleco in db.clecos on clorc.CODCLECO equals cleco.CODCLECO
    join orcam in db.orcams on clorc.CODORCAM equals orcam.CODORCAM
    where 1 == 1 && resti.DATAREST <= dt2 && resti.DATAREST >= dt1 &&
    orcam.CODORCAM == codorcam && cleco.CLASSREC == 1
    select new
    {
        CODCLECO = cleco.CODCLECO,
        ANTECIP = (ifivaded == true ? resti.VALOR - resti.VALORIVA :
resti.VALOR) * -1
    };

//agrupamento de todos os realizados
var anticip = anticipRepos.Concat(anticipCompr);
anticip = anticip.Concat(anticipLreca);
anticip = anticip.Concat(anticipResti);

var anticipSum =
    from a in anticip
    group a by a.CODCLECO into s
    select new
    {
        CODCLECO = s.Key,
        TOTALANTECIP = s.Sum(a => a.ANTECIP)
    }

```

```
};

//Agrupamento final
var sumFin =
    from da in dotAjumSum
    join d in db.clecos on da.CODCLECO equals d.CODCLECO
    select new
    {
        CODCLECO = d.CODCLECO,
        RUBRICA = d.RUBRICA,
        DESCRICA = d.DESCRICA,
        DEPENDEN = d.DEPENDEN,
        NIVEL = d.NIVEL,
        FIXO = d.FIXO,
        VARIAVEL = d.VARIAVEL,
        TPCUST = d.TPCUST,
        CLECOR = d.CLECOR,
        CLASSDES = d.CLASSDES,
        CLASSREC = d.CLASSREC,
        ORCAMENT = da.TOTALDOT,
        REALIZA = 0,
        ANTECIP = 0
    };

//lista de m1_TMPRPT
lt = new List<M1_TMPRPT>();

foreach (var line in sumFin)
{
    it = new M1_TMPRPT();

    //realizado
    var real = from item in realizSum
                where item.CODCLECO == line.CODCLECO
```

```
        select item;

it.realiza = real.Count() == 0 ? 0 : real.First().TOTALREALIZA;

//antecipado

var atc = from item in antecipSum

        where item.CODCLECO == line.CODCLECO

        select item;

it.antecip = atc.Count() == 0 ? 0 : atc.First().TOTALANTECIP;


//orcamentado; previamente calculado

it.orcament = line.ORCAMENT;


it.classdes = line.CLASSDES;
it.classrec = line.CLASSREC;
it.clecor = line.CLECOR;
it.codcleco = line.CODCLECO;
it.dependen = line.DEPENDEN;
it.descrica = line.DESCRICA;
it.fixo = line.FIXO;
it.nivel = line.NIVEL;
it.rubrica = line.RUBRICA;
it.tpcust = line.TPCUST;
it.variavel = line.VARIAVEL;

lt.Add(it);
}


//Agrupamento das rubricas pai

List<string> tempCod =new List<string>();

List<int> tempInt = new List<int>();

int i = 0;

double nivelMaximo;


foreach (M1_TMPRPT item in lt)
{

    tempCod.Add(item.codcleco);
```

```
        tempInt.Add((int) item.nivel);
    }

    for (i = 0; i < tempCod.Count(); i++)
        adicionaPai(tempCod[i], tempInt[i]);

    nivelMaximo = (double) lt.Max(q => q.nivel);
    somaSaldos(nivelMaximo - 1);

    //Ordenação das rubricas
    var ordena = from item in lt
                  where item.nivel == 1
                  select item;
    ordena = ordena.OrderBy(o => o.rubrica);

    ltTemp = new List<M1_TMPRPT>();
    foreach (var cleco in ordena)
        adicionaFilhos(cleco);
    lt = ltTemp;
    //Mapa final

    return mapOrcarublistToString(lt);

}

private List<mapOrcarub> mapOrcarublistToString(List<M1_TMPRPT> itemsList)
{
    List<mapOrcarub> mapa;
    mapOrcarub mapLine;
    int i = 0, k = 0, j = 0;
    mapa = new List<mapOrcarub>();

    foreach (M1_TMPRPT item in itemsList)
```

```

{
    mapLine = new mapOrcarub();

    mapLine.codigo = i;

    mapLine.descrica = item.rubrica + " - " + item.descrica;

    mapLine.orcament = formatFloatValue((double)item.orcament) + " " +
currency;

    mapLine.realiza = formatFloatValue((double)item.realiza) + " " +
currency;

    mapLine.antecip = formatFloatValue((double)item.antecip) + " " +
currency;

    mapLine.diforcreal = formatFloatValue((double)(item.orcament -
item.realiza)) + " " + currency;

    mapLine.isDespesaReceita = (int)item.classrec;

    for (j = 0; j < (item.nivel - 1); j++)
        for (k = 0; k < numEspacos; k++)
            mapLine.descrica = ' ' + mapLine.descrica;

    i++;

    mapLine.rubrica = item.rubrica;

    mapLine.dependen = item.dependen;

    mapa.Add(mapLine);
}

//return lt;

return mapa;

}

```

```

private string toLine(mapOrcarub item)
{
    string res;

    char token='|';

    int codigo, isDespesaReceita;

    if (item.codigo == null) codigo = -1;

```

```
        else codigo = (int) item.codigo;

        if (item.codigo == null) isDespesaReceita = -1;

        else isDespesaReceita = (int)item.isDespesaReceita;

        res = codigo.ToString() + token + isDespesaReceita.ToString() + token +
        item.rubrica + token + item.dependen + token + item.descrica + token + item.orcament +
        token + item.realiza + token + item.antecip + token + item.diforcreal + token;

        return res;
    }

    public void SaveMap1(List<mapOrcarub> mapa, string filename)
    {

        string filen = "MapaOrcamento" + DateTime.Now.ToShortDateString()+".mcg";
        if (filename == null) filename = pathname + filen;
        else filename = pathname + filename;
        TextWriter tw = new StreamWriter(filename);

        tw.WriteLine("=====INÍCIO DO MAPA=====");
        tw.WriteLine("=====TIPO: MAPA DE EXECUÇÃO DO ORÇAMENTO=====");
        tw.WriteLine("=====GERADO EM: "+DateTime.Now.ToLongDateString()+"=====");

        tw.WriteLine("");
        foreach (mapOrcarub item in mapa)
            tw.WriteLine(toLine(item));

        // close the stream
        tw.Close();
    }

    private mapOrcarub toMapOrcaRub(string line)
    {
        mapOrcarub m = new mapOrcarub();
        string[] fields = line.Split('|');
```

```
m.codigo = Convert.ToInt32(fields[0]);
m.isDespesaReceita = Convert.ToInt32(fields[1]);

m.rubrica = fields[2];
m.dependen = fields[3];
m.descrica = fields[4];
m.orcament = fields[5];
m.realiza = fields[6];
m.antecip = fields[7];
m.diforcreal = fields[8];

return m;
}

public List<mapOrcarub> LoadMap1(string filename)
{
    List<mapOrcarub> mapa= new List<mapOrcarub>();

    mapOrcarub m;

    TextReader tr= new StreamReader(pathname+filename);

    //posicionar
    for (int i = 0; i < 5; i++)
        tr.ReadLine();

    string line;
    while ((line = tr.ReadLine()) != null)
    {
        m = toMapOrcaRub(line);
        mapa.Add(m);
    }

    tr.Close();

    return mapa;
}

#endregion
}
```

K) Integração – Especificações Técnicas (MSMQ)

“Microsoft Message Queuing (MSMQ) technology enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily offline.” [msm08]

Os objectos Flash recebem e enviam de volta mensagens XML como forma de comunicação com a aplicação. Estas mensagens circulam com base na tecnologia de *Microsoft Message Queuing* do *Windows*.

O *Microsoft Message Queuing* permite que aplicações que correm em alturas diferentes ou sobre diferentes plataformas e redes comuniquem entre si, ainda que estejam *offline* (comunicação assíncrona). Neste caso, esta tecnologia é utilizada como meio para comunicar entre controlos da mesma aplicação mas que utilizam tecnologias diferentes (MFC e Flash).

Tal como podemos verificar na Figura 31, a arquitectura do *Microsoft Message Queuing* (MSMQ) é simples e utilizável em vários contextos. Uma vez que é utilizada mesmo quando as aplicações estão *offline*, é uma alternativa mais estável e com maior desempenho que os *web services* (vistos em capítulos posteriores), servindo actualmente como base de comunicação entre todas as aplicações da Quidgest. O MSMQ funciona com uma *queue* de mensagens que coloca as mensagens a serem recebidas por uma aplicação numa fila de prioridades, à espera de serem processadas.

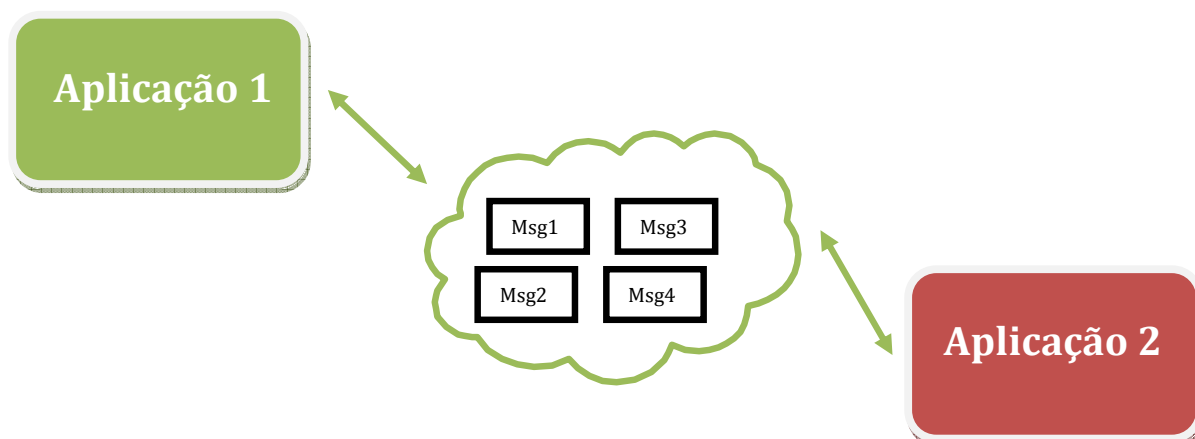


FIGURA 31 – ARQUITECTURA DO MICROSOFT MESSAGE QUEUING.

A última versão do MSMQ também possui funcionalidades de transacção. Esta funcionalidade permite-nos definir uma transacção para as nossas mensagens em que, caso alguma delas tenha uma mensagem de erro no retorno, o próprio MSMQ anula o envio das anteriores. Isto é muito útil na parametrização do Flash uma vez que nos permite enviar vários comandos em XML e garantir que estes só são executados se os também o tenham sido com sucesso.

L) Considerações Técnicas de Contabilidade

Como podemos verificar na Figura 32, não devemos considerar as linhas de movimento associadas às contas 68 e 69, bem como as 77, 78 e 79. Pelo menos, para o cálculo dos Resultados Operacionais, uma vez que essas contas devem ser consideradas sim para o cálculo das contas 82, 83 e 84.



19 - Ajustamentos de aplicações do tesouro	29 - Provisões	39 - Ajustamentos de existências	49 - Ajustamentos de investimentos financeiros	59 - Resultados transitados
6 - Custos e perdas	7 - Proventos e ganhos	8 - Resultados	9 - Contabilidade de custos	0
61 - Custos das mercadorias vendidas e das matérias consumidas	71 - Vendas	01 - Resultados operacionais		
62 - Fornecimentos externos	72 - Prestações de serviços	02 - Resultados financeiros		
63 - Impostos	73 - Proventos suplementares	03 - (Resultados correntes)		
64 - Custos com o pessoal	74 - Subsidios à exploração	04 - Resultados extraordinários		
65 - Outros custos e perdas operacionais	75 - Trabalhos para a própria empresa	05 - (Resultados antes do imposto)		
66 - Amortizações e ajustamentos do exercício	76 - Outros proventos e ganhos operacionais	06 - Imposto sobre o rendimento do exercício		
67 - Provisões do exercício	77 - Reversões de amortizações e ajustamentos			
68 - Custos e perdas financeiros	78 - Proventos e ganhos financeiros	08 - Resultado líquido do exercício		
69 - Custos e perdas extraordinários	79 - Proventos e ganhos extraordinários	09 - Dividendos antecipados		

FIGURA 32 – TABELA DO PLANO OFICIAL DE CONTAS DO CNC (COMISSÃO DE NORMALIZAÇÃO CONTABILÍSTICA). PODEMOS AQUI VERIFICAR AS CONTAS FUNDAMENTAIS PARA OS INDICADORES FINANCEIROS ACIMA REFERIDOS.

Da análise ao Plano Oficial de Contas, temos que as contas dos Resultados devem ser calculadas da seguinte forma:

$$81 \text{ (Resultados Operacionais)} = 71 + 72 + 73 + 74 + 75 + 76 - 61 - 62 - 63 - 64 - 65 - 66 - 67$$

$$82 \text{ (Resultados Financeiros)} = 78 - 68$$

$$83 \text{ (Resultados Correntes)} = 81 + 82$$

84 (Resultados Extraordinários) = 79 - 69

88 (Resultados Líquido do Exercício ou **Resultados Líquidos**) = 83 + 84

Os Resultados Líquidos são apenas a soma entre os Resultados Operacionais, os Resultados Financeiros e os Resultados Extraordinários e dão-nos uma perspectiva mais concreta dos lucros da organização.

Segundo o Plano Oficial de Contabilidade Português (POC), o Volume de Negócios “corresponde à quantia líquida das vendas e prestações de serviços(...)”, tendo em conta as reduções em vendas e não incluindo o IVA (Imposto sobre o Valor Acrescentado) nem quaisquer outras taxas similares. Por aqui, conseguimos compreender essencialmente o comportamento das vendas ao longo do tempo. Segundo o mesmo POC, podemos obter este valor analisando todas as linhas de movimento (LMOVIs do CNT) associadas à conta “71 – Vendas e Prestações de Serviços”.

Os Resultados Operacionais, ou EBIT (*Earnings Before Interest and Taxes*) é uma medida que avalia a capacidade duma organização de gerar lucros que exclui juros e impostos. Esta medida encontra-se através da conta 81 do POC. Estes equivalem ao saldo da diferença entre os Custos e Perdas (Conta 6) e os Proveitos e Ganhos (Conta 7). Contudo, nem todas as contas que pertencem devem ser contabilizadas.

M) Análise De Mercado

Este Anexo pretende descrever e analisar de forma sumária o que existe feito no mercado que entre no âmbito do conjunto de problemas descrito no Capítulo 1. Deste capítulo analisamos três tipos de aplicações tipo:

- Aplicações que implementem Sistemas de Gestão Financeira;
- Aplicações que implementem Controlo de Gestão e/ou metodologias de apoio;
- Aplicações que implementem *RIA*'s no contexto de *data binding*;

Perante as três tipologias acima descritas, pretende-se efectuar uma análise de mercado, que nos permita compreender o que já existe feito e/ou os estudos e investigações realizadas nesse sentido, que nos permita compreender melhor quais as soluções que temos ainda de encontrar para resolver o nosso problema ou se esta será apenas uma compilação de várias já existentes.

A análise não será exaustiva uma vez que muitas das tecnologias que vamos utilizar para resolver o problema foram previamente definidas por escolha estratégica da empresa. O mesmo acontece com os requisitos definidos para a solução base do Sistema de Gestão Financeira.

SISTEMAS DE GESTÃO FINANCEIRA NO MERCADO

No presente Capítulo, vamos efectuar uma pequena análise de mercado para compreender que funcionalidades oferece a actual concorrência no mercado face ao produto que pretendemos desenvolver. Através da pesquisa efectuada, identificámos sete concorrentes directos do CNT no mercado: SAGE, Primavera, ArtSoft, PHC, Sydneys, IBS, Gestware.

Destes, seleccionámos dois casos particulares por entendermos que são as empresas cujos produtos afectam a mesma quota de mercado que o CNT: a Sage e a Primavera.

A maior parte destas aplicações possui uma série de funcionalidades base que não permitem qualquer diferenciação tais como introdução de compromissos, processamento de facturas, gestão de diários, etc.. Como tal, vamos efectuar uma comparação com recurso a uma série de funcionalidades avançadas que permitam critérios de diferenciação mais mensuráveis.

Em seguida, vamos enunciar as funcionalidades avançadas que iremos analisar nas aplicações, para em seguida analisarmos as duas aplicações concorrentes.

FUNCIONALIDADES AVANÇADAS

Vamos em seguida enunciar uma lista de funcionalidades avançadas para comparar as duas aplicações, optando por uma descrição simplificada para evitar que este documento se torne demasiado técnico, no que diz respeito à Contabilidade. As funcionalidades são as seguintes:

Multi Organismo ou Multi Empresa: Suporte de várias empresas (sector privado) ou organismos (sector público) na mesma aplicação.

Multi Anual: Suporte de vários anos na organização da contabilidade.

Imputação de Vários Tipos de Centros de Custo: Imputação de afectações a Centros de Custo de distintas Tipificações.

Centros de Custo em Árvore: Suporta uma organização em árvore dos Centros de Custo

Automatismo Lançamentos Contabilísticos: Modo de efectuação dos lançamentos contabilísticos (de forma automática ou manual). A existência do modo automático reveste-se de particular importância uma vez que a sua existência dispensa um conhecimento funcional de contabilidade por parte do utilizador.

Classificador da Despesa/Receita: Associa um determinado produto/artigo à sua rubrica orçamental, valor de iva, mercado, conta do POC, etc..

Gestão da Contabilidade Geral e Analítica: Gestão na mesma aplicação da contabilidade geral, associada ao POC e a contabilidade Analítica, dos Centros de Custo.

Gestão das Contas Correntes: Análise as actuais contas de fornecedores e clientes e sua associação com o mapa de antiguidade de saldos¹⁷.

Apuramento de IVA baseado em Tabelas Configuráveis: Apuramento automático dos vários tipos de IVA. Este automatismo pode-se basear numa tabela do IVA, que pretende codificar os vários tipos de IVA de acordo com as suas características: valor, mercado de origem, tipo, etc.. Este código do IVA serve para que possamos alterar o IVA de determinada configuração apenas numa tabela, sendo que o código se mantém inalterado nos documentos contabilísticos.

Cálculo Pro Rata: Implementação do cálculo automático do pro rata, isto é, pré-definir qual a percentagem de dedução do IVA.

Diários: Gestão de contabilidade por Diários. Os Diários agregam movimentos numa determinada área funcional (ex.: vendas ou compras) e possuem uma numeração específica. Estes movimentos são registados em relação ao dia em que são efectuados.

Gestão e Impressão de Mapas e Obrigações Legais: Geração automática de vários mapas tais como a Demonstração de Resultados, o Anexo 1, o Modelo 30, os Balanços e Balancetes, o Modelo 22, etc..

Ficheiro SAFT: O ficheiro SAFT - *Standard Audit File for Tax Purposes* – é um ficheiro que contém os dados contabilísticos de uma dada organização relativos a um dado período

¹⁷ - mapa que nos permite verificar quais os clientes devedores e quais os custos associados a essa dívida no período em que esta se mantém.

de tempo. Este ficheiro encontra-se normalizado e existe para efectuar uma parametrização da contabilidade por parte do estado. O sistema deverá gerar este ficheiro automaticamente a partir da sua contabilidade.

Consolidação Integrada: consolidação das diferentes visões das várias organizações presentes no sistema (multiorganismo) numa só.

Reconciliação Bancária: Esta funcionalidade traduz-se num comparador de movimentos efectuados no sistema com os pagamentos e recebimentos presentes nos extractos bancários (integração com sistema bancário) do mesmo período.

Posicionamento Bancário: Acesso à situação actual das contas bancárias afectas ao presente organismo: créditos, depósitos, listagem de contas à ordem e saldos actuais das várias contas nos diversos bancos.

Controlo Orçamental Integrado: verificação do grau de cumprimento do Orçamento. O Controlo Orçamental é uma das disciplinas do Controlo de Gestão devendo, como tal, implementar mapas de controlo orçamental na mesma aplicação da gestão de contabilidade.

Relatórios sobre Indicadores Chave do Negócio: geração de mapas com indicadores chave de Negócio tais como Volume de Negócios ou Resultados Operacionais. Este tipo de indicadores podem também estar associados a uma ferramenta de Gestão Estratégica como o Balanced ScoreCard.

Apresentação Gráfica dos Dados: geração automática de gráficos com base nos dados do sistema de gestão de contabilidade.

Geração Automática do Orçamento: geração automática das dotações mensais para uma determinada rubrica orçamental com base no orçamento anual desta.

Definição de Objectivos Estratégicos: definição de objectivos estratégicos para organização numa forma integrada com a aplicação (tal como os Relatórios de Indicadores Chave do Negócio), Tipicamente, a implementação destes objectivos requer a integração e/ou a implementação numa metodologia de gestão estratégica como o Balanced Scorecard.

Comparação Económica Inter-Anos: geração de mapas referentes a anos diferentes e formas de os comparar.

Análise dos Desvios de Orçamentação: a implementação de mapas de execução de orçamento.

Em seguida, vamos falar um pouco das aplicações do mercado, referindo se elas possuem, ou não, as funcionalidades listadas.

SAGE E PRIMAVERA BSS

A Sage Portugal é uma empresa com cerca de 20 anos que disponibiliza software de gestão e serviços para pequenas e médias empresas. Tem crescido ao longo dos anos através de uma política agressiva de aquisições e está integrada num grupo multinacional do sector. [sag08]

A Primavera BSS (Business Software Solutions) é uma multinacional que nasceu em 1993, de origem portuguesa, que desenvolve e comercializa, à semelhança da Sage, soluções de gestão para empresas de média e grande dimensão do sector privado e do sector público. Os seus sistemas possuem cerca de 150 mil utilizadores diários pertencentes a 40 mil organizações sediadas, essencialmente, em mercados de latinos e de língua portuguesa. [pri08]

Os produtos que estamos aqui a analisar são o Sage Next e a Área Financeira do Primavera Executive. O Sage Next é um sistema de gestão de contabilidade destinado a empresas de média dimensão. O Primavera Executive é uma solução completa de gestão contudo, neste contexto, vamos analisar apenas a sua vertente financeira. Em seguida, apresentamos na Tabela 4 um quadro com as diversas funcionalidades avançadas listadas.

Funcionalidade	Sage Next	Primavera
Multi Organismo ou Multi Empresa	Sim	Sim
Multi Anual	Não	Sim
Imputação de Vários Tipos de Centros de Custo	Sim	Não
Centros de Custo em Árvore	Sim	Sim
Automatismo Lançamentos Contabilísticos	Sim	Sim
Classificador da Despesa/Receita	Não	Não
Gestão da Contabilidade Geral e Analítica	Sim	Sim
Gestão das Contas Correntes	Sim	Não
Apuramento de IVA baseado em Tabelas Configuráveis	Sim	Sim
Cálculo Pro Rata	Não	Não
Diários	Sim	Sim
Gestão e Impressão de Mapas e Obrigações Legais	Sim	Sim
Ficheiro SAFT	Sim	Não
Consolidação Integrada	Sim	Sim
Reconciliação Bancária	Sim	Sim
Posicionamento Bancário	Não	Sim
Controlo Orçamental Integrado	Não	Não
Relatórios sobre Indicadores Chave do Negócio	Sim	Não
Apresentação Gráfica dos Dados	Sim	Não
Geração Automática do Orçamento	Não	Não
Definição de Objectivos Estratégicos	Não	Não ¹⁸
Comparação Económica Inter-Anos	Não	Não
Análise dos Desvios de Orçamentação	Sim	Não

TABELA 4 – QUADRO DAS FUNCIONALIDADES AVANÇADAS DO SAGE NEXT E PRIMAVERA EXECUTIVE

¹⁸ - esta vertente só está disponível nos restantes módulos do Executive..

Pela análise efectuada, consideramos que o Sage Next é um produto bastante completo e com uma quota de mercado bastante razoável, sendo que possui algumas falhas no que respeita à integração com o Controlo de Gestão.

Consideramos ainda que o Primavera Executive é um produto mais incompleto que o Sage Next na medida em que não possui tantas funcionalidades avançadas. Este facto torna-se mais evidente se analisarmos as funcionalidades que implicam automatismos e integração do Controlo de Gestão, o que faz com que os utilizadores desta aplicação tenham forçosamente que possuir conhecimento funcional de Contabilidade.

BALANCED SCORECARD E CONTROLO DE GESTÃO

Este capítulo descreve, de forma sumária, uma pequena análise de mercado sobre as aplicações que suportam o Controlo de Gestão e a Gestão Estratégica através da metodologia Balanced ScoreCard.

Controlo de Gestão

No mercado, existem várias aplicações de Controlo de Gestão. Todas as empresas identificadas no capítulo anterior possuem, de forma geral, uma solução de Controlo de Gestão que assenta na geração de mapas estáticos como o mapa de execução de orçamento, por exemplo. Contudo, não existe nenhuma aplicação com geração de conteúdos dinâmicos ou interacções. As aplicações actuam normalmente de forma isolada e não estão integradas nas aplicações de Contabilidade.

Verifica-se ainda uma tendência por parte das empresas em optarem por desenvolverem o seu Controlo de Gestão através de mapas feitos manualmente à medida da sua estratégia recorrendo ao Microsoft Excel.

Pelo facto de estas serem, duma forma geral, aplicações muito simples e semelhantes à aplicação que aqui apresentamos, não vamos analisar nenhuma das aplicações em particular.

Balanced ScoreCard

Existem neste momento no mercado várias ferramentas que implementam a metodologia Balanced ScoreCard (Capítulo 4). A maior parte recorre a uma ferramenta gráfica onde temos de introduzir todos os dados manualmente, bem como definir os objectivos, os indicadores, etc.. Contudo, estas ferramentas não possuem normalmente quaisquer integrações com ferramentas de gestão de contabilidade, por exemplo. Um bom exemplo duma aplicação que implementa Balanced ScoreCard é a aplicação da eVisualReport, que vamos analisar em seguida.

Esta aplicação tem por objectivo implementar a metodologia Balanced ScoreCard na gestão estratégica duma organização. Do ponto de vista tecnológico, esta aplicação é totalmente desenvolvida em Adobe Flash. Podemos verificar o seu interface aqui [evi08].

As suas principais características são as seguintes:

- Importação de dados apenas a partir de Microsoft Excel;
- Visualização em páginas de internet HTML, Powerpoint e Flash;
- Multi utilizador sem vários níveis de permissões;
- Definição de mapas estratégicos, objectivos, indicadores e perspectivas sem suporte multi anual;
- Limite do número de mapas estratégicos: 6;

Apesar do bom ambiente gráfico, esta aplicação apresenta algumas limitações como podemos verificar através da análise das suas características.

Como já verificámos no Capítulo 2, o facto de assentar todo o seu desenvolvimento em *RIA's* coloca limitações ao tratamento dos conteúdos que limita a progressão desta aplicação. Por estes factores, entende-se que a solução da eVisualReports é uma solução interessante apenas para uma pequena empresa.

RIA'S E DATA BINDING

Neste capítulo, vamos falar sobre o desenvolvimento de *Rich Internet Applications* no contexto de *Data Binding*. *Data Binding* é, tal como o nome indica, a ligação efectuada entre aplicações e fontes de dados, tais como SGBD.

O desenvolvimento de RIA's neste contexto é complexo uma vez que as duas tecnologias que implementam RIA's são executadas em plug-ins o que limita, como já verificámos, o seu acesso a fontes de dados.

Como já foi referido no Capítulo 2, vamos, por opção estratégica, falar de apenas duas tecnologias que implementam RIA's: o Adobe Flash e o Microsoft Silverlight. Todavia, por se entender relevante no contexto deste relatório, vamos falar inicialmente duma análise de mercado efectuada por uma tese particularmente interessante de Florian Moritz intitulada “Rich Internet Applications (RIA): A Convergence of User Interface; Paradigms of Web and Desktop; Exemplified by JavaFX” [mor08]. Em seguida, vamos verificar que aplicações existem no mercado que desenvolvam RIA's neste contexto, recorrendo a Flash e/ou Silverlight.

FLASH E DATA BINDING

Nesta secção vamos analisar que tipo de aplicações estão presentes no mercado que recorrem à tecnologia Flash para implementar aplicações com ligações a fontes de dados (*data binding*).

O Flash dispõe de vários tipos de controlos pré-definidos que nos permitem conectar a uma fonte de dados tais como: *DataHolder* e *DataSet*, *RBDMSResolver*, *DataGrid*, *XML Connector*, *WebServices Connector* e *Xupdate Resolver*. Existem várias aplicações no mercado que utilizam estes componentes. Vamos então considerar as aplicações desenvolvidas por uma empresa no mercado: a Advance Flash. A Advance Flash é uma empresa que desenvolve aplicações e componentes web com recurso a Flash. [adv09]

Contudo, esta também desenvolve vários componentes comercializáveis. Estes componentes limitam-se a implementar de uma forma generalista funcionalidades muito específicas, como uma animação simples duma paisagem agradável com a passagem de

algumas nuvens até à listagem de produtos com base numa Base de Dados, por exemplo. Deste último temos uma demonstração possível no seu próprio site, todo ele desenvolvido nesta tecnologia, que assenta num módulo de compras online de componentes Flash cujos produtos estão inseridos numa Base de Dados remota. [adv09]

Para verificarmos componentes mais complexos e mais adequados ao nosso contexto de utilização destas tecnologias, temos que abordar aplicações desenvolvidas em Flash com recurso ao controlo de DataGrid. Para tal, temos dois exemplos da utilização deste controlo presentes num conhecido *Blog* desta tecnologia [pin07].

Na primeira aplicação, podemos verificar a utilização duma *datagrid* para inquéritos ou avaliações dinâmicas, em que as perguntas estão numa determinada tabela da Base de Dados. Na segunda aplicação, verificamos uma espécie de *showcase* de casas, cuja toda a informação (incluindo fotografias) é também proveniente duma base de dados remota.

Estes exemplos de utilizações de Flash no mercado vêm comprovar não só a variedade de soluções existentes bem como as suas diferentes utilizações, o que nos deixa concluir que o Flash é uma tecnologia com provas dadas no mercado no que diz respeito às aplicações com *data binding*.

SILVERLIGHT E DATA BINDING

À semelhança do Capítulo anterior, aqui vamos efectuar uma análise sobre o que já existe feito no mercado utilizando esta tecnologia com *Data Binding*. Apesar de ser uma tecnologia recente, existem já diversos desenvolvimentos nesta tecnologia uma vez que esta assenta em tecnologias com maior utilização, tais como o C# ou a própria *framework* do .NET. Contudo, o grande problema destas aplicações é que esta tecnologia ainda não apresenta qualquer versão estável (release) e compatibilidade entre versões deixa muito a desejar, pelo que uma aplicação que funcione numa versão 2.0 Beta 2 poderá não funcionar na versão 1.0 do Silverlight. Para compreendermos o que já existe desenvolvido, vão ser apresentados dois exemplos:

- “*Building a Framework for Silverlight Line-Of-Business Applications*” é um título dum artigo com várias partes que nos descreve e sugere formas, controlos e outras utilizações para desenvolver aplicações *business oriented*; [bui08]
- DataGrid Xceed. Um controlo que, utilizando como base o controlo DataGrid da versão 2.0 Beta 2 do Silverlight, lhe adiciona várias funcionalidades interessantes neste contexto. [xce08]

O primeiro é um artigo desenvolvido por um elemento da equipa de desenvolvimento da Microsoft da tecnologia Silverlight. Fala-nos da utilização duma *datagrid*, do *styling*, da ligação de vários tipos de controlos à base de dados, políticas de segurança, encriptação de dados, etc.. A relevância deste artigo é que está dividido em várias partes e é constantemente actualizado, sendo lançadas com uma periodicidade considerável novas partes do artigo. Como tal, acaba por ser uma espécie de magazine que revela quais as tendências e o que se faz de novo em aplicações *business oriented*, que acaba por ser uma generalização possível do *data binding*.

O segundo, é um controlo desenvolvido pela Xceed. A Xceed é uma empresa que desenvolve componentes e ferramentas para a plataforma .NET. O seu produto mais famoso é um controlo *DataGrid* que é utilizado em várias aplicações da Microsoft tais como o Flight Simulator ou o Windows Home Server. Contudo, este controlo adiciona várias funcionalidades que não estão disponíveis no controlo nativo do WPF/Silverlight.

Temos então um controlo com boa usabilidade e um aspecto gráfico acima da média. Podemos reparar que cada registo possui uma fotografia associada. Conseguimos ainda definir uma ordem para os campos de forma gráfica, entre muitas outras funcionalidades. A grande diferença é que a implementação destas funcionalidades consiste apenas numa declaração dum componente ou na alteração dum determinado estado ou propriedade. Isto é, o controlo possui na sua API métodos que suportam estas funcionalidades.

Algumas das funcionalidades suportadas por este controlo são as seguintes:

- Listagem automática das fotografias;

- Ordenação gráfica e hierarquizada das listagens;
- Diferentes vistas para os dados (*3D*, *Cards*, *list*, etc.);
- Diferentes *skins*;
- Suporte a edição e validação de campos formatados tais como datas, preços, etc..;
- Geração automática de estatísticas de utilização dos controlos;
- Suporte a grandes volumes de dados com gestão de campos e de memória;
- Possibilidade de alterar a fonte de dados em *runtime* – Data Binding Dinâmico;

Este controlo é, de facto, de uma enorme valia tecnológica pois permite a qualquer empresa desenvolver as suas aplicações com data binding partindo dum ponto muito avançado da tecnologia. Contudo, a sua utilização em Silverlight é comprometida pela limitação existente na utilização das funcionalidade de WPF em Silverlight, – este corre num *plug-in* e devido a isso, não possui todas as capacidades que uma aplicação baseada em WPF teria.

MERCADO – VISÃO GERAL

Após as diversas análises efectuadas nos capítulos anteriores, podemos concluir antes de mais que existem soluções no mercado que abordam os diversos contextos que envolvem este projecto.

Existem vários sistemas de Gestão Financeira e Contabilidade, sendo que a maioria requer um conhecimento funcional médio ou elevado para que possam ser utilizados. Existem ainda produtos mais completos que possuem algumas integrações e automatismos particularmente interessantes, tais como os lançamentos contabilísticos ou a consolidação multiempresa integrada. Apesar de existirem aplicações que implementam a metodologia BSC, estas não possuem, normalmente, integrações generalistas com outras aplicações.

O mercado possui diversas soluções de *RIA's* utilizadas num contexto empresarial, com recurso a *data binding*. Existem aplicações e componentes desenvolvidos em Flash e Silverlight, sendo que o primeiro contém uma quota de mercado substancialmente superior.

Esta diferença deve-se, essencialmente, pelo facto de as empresas depositarem uma maior confiança nesta tecnologia por existir no mercado há já vários anos.

BIBLIOGRAFIA E REFERÊNCIAS

- [adv09] Advance Flash. (2009). Stock Flash. Obtido em 25 de Fevereiro de 2009, de Advance Flash: <http://www.advanceflash.com/>
- [evi08] eVisualReport. (2008). Balanced ScoreCard. Obtido em 17 de Fevereiro de 2009, de eVisualReport: <http://www.evisualreport.com/?gclid=CN6c3Ne49ZgCFQulQwodQ0X21w>
- [bui08] Microsoft. (8 de Setembro de 2008). Building a Framework for Silverlight Line-Of-Business Applications. Obtido em 26 de Fevereiro de 2009, de SilverlightShow: <http://www.silverlightshow.net/items/Building-a-Framework-for-Silverlight-Line-Of-Business-Applications.aspx>
- [pin07] Pinho, C. (2 de Fevereiro de 2007). DataGrid CellRenderer more examples. Obtido em 25 de Fevereiro de 2009, de Flash Enabled Blog: <http://flashenabledblog.com/2007/02/02/datagrid-cellrenderer-more-examples/>
- [pri08] Primavera. (2008). Primavera BSS Portugal. Obtido em 17 de 2 de 2009, de Primavera BSS Portugal: <http://www.primaverabss.com/pt/PortalRender.aspx?PageID=aca4d893-9473-4b55-a93b-5e665f3bb1e9>
- [sag08] Sage. (2008). Sage Portugal. Obtido em 23 de Fevereiro de 2009, de Sage Portugal: <http://www.sage.pt/UI/Editor2/editor/Default.aspx?action=ArticleViewer&target=445&sec=440&code=3>
- [xce08] Xceed. (2008). Xceed DataGrid for Wpf. Obtido em 9 de Fevereiro de 2009, de Xceed: http://xceed.com/Grid_WPF_Demo.html
- [ado08] Adobe. (7 de Agosto de 2008). *Adobe - Flash Lite: Architecture*. Obtido em 16 de Fevereiro de 2009, de Adobe - Flash Lite: Architecture: <http://www.adobe.com/products/flashlite/architecture/>
- [cma07] C.M. Aveiro. (31 de Julho de 2007). *C.M. Aveiro*. Obtido em 17 de Fevereiro de 2009, de Balancete de Contabilidade: <http://www.cm-aveiro.pt/www/cache/imagens/XPQ5FaAXX9847aGdb9zMjjeZKU.pdf>
- [cai03] Caiado, A. C. (2003). *Contabilidade de Gestão*. Áreas Editora.
- [eze07] Ezell, J. (3 de Maio de 2007). *Silverlight vs. Flash: The Developer Story*. Obtido em 19 de Fevereiro de 2009, de Silverlight vs. Flash: The Developer Story: <http://weblogs.asp.net/jezell/archive/2007/05/03/silverlight-vs-flash-the-developer-story.aspx>

- [har08] Hard Rock. (2008). *Hard Rock Memorabilia*. Obtido em 5 de Janeiro de 2009, de Hard Rock: <http://memorabilia.hardrock.com/>
- [kap04] Kaplan, R. S., & Norton, D. P. (2004). *Strategy Maps - Converting Intangible Assets Into Tangible Outcomes*. Harvard Business School Press.
- [kap96] Kaplan, R. S., & Norton, D. P. (1996). *The Balanced ScoreCard*. Harvard Business School Press.
- [kap08] Kaplan, R. S., & Norton, D. P. (2008). *The Execution Premium - Linking Strategy to Operations for Competitive Advantage*. Harvard Business School Press.
- [kap01] Kaplan, R. S., & Norton, D. P. (2001). *The Strategy Focused Organization*. Harvard Business School Press.
- [man08] Manual do Utilizador do Balanced ScoreCard Quidgest. (2008). Quidgest - Consultores de Gestão, SA.
- [clr08] Microsoft. (2008). *Common Language Runtime Overview*. Obtido em 22 de 1 de 2009, de MSDN: [http://msdn.microsoft.com/en-us/library/ddk909ch\(vs.71\).aspx](http://msdn.microsoft.com/en-us/library/ddk909ch(vs.71).aspx)
- [mic08] Microsoft. (2008). *Data Points: Service Driven Apps With Silverlight 2 and WCF*. Obtido em 17 de Fevereiro de 2009, de MSDN: <http://msdn.microsoft.com/en-us/magazine/cc794260.aspx>
- [dep08] Microsoft. (Novembro de 2008). *Deep Zoom*. Obtido em 17 de Fevereiro de 2009, de MSDN: [http://msdn.microsoft.com/en-us/library/cc645050\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc645050(VS.95).aspx)
- [iso08] Microsoft. (2008). *Isolated Storage*. Obtido em 17 de Fevereiro de 2009, de MSDN: [http://msdn.microsoft.com/en-us/library/3ak841sy\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/3ak841sy(VS.80).aspx)
- [msm08] Microsoft. (s.d.). *Microsoft Message Queuing*. Obtido em 16 de 2 de 2009, de Microsoft Message Queuing: <http://www.microsoft.com/windowsserver2003/technologies/msmq/default.aspx>
- [swi08] Microsoft. (17 de Março de 2008). *Swiss MSDN Team Blog : Silverlight 2 Beta 1 + WCF + LINQ to SQL*. Obtido em 5 de Janeiro de 2009, de MSDN: http://blogs.msdn.com/swiss_dpe_team/archive/2008/03/17/silverlight-2-beta1-wcf-linq-to-sql-a-powerfull-combination.aspx
- [the08] Microsoft. (11 de Outubro de 2008). *The Official Microsoft Silverlight Site*. Obtido em 24 de Outubro de 2009, de The Official Microsoft Silverlight Site: <http://silverlight.net/>
- [usi08] Microsoft. (17 de 2 de 2008). *Using MFC to Host a WebBrowser Control*. Obtido em 5 de 2 de 2009, de MSDN: <http://msdn.microsoft.com/en-us/library/aa752046.aspx>

- [mor08] Moritz, F. (Janeiro de 2008). *Rich Internet Applications (RIA)*. Obtido em 5 de Janeiro de 2009, de Rich Internet Applications (RIA): <http://www.flomedia.de/diploma/documents/DiplomaThesisFlorianMoritz.pdf>
- [lmo08] Moroney, L. (2008). *Introducing Microsoft Silverlight 2*. Microsoft Corporation.
- [poc08] POC - Plano Oficial de Contabilidade. (2008). Obtido em 19 de 1 de 2009, de Comissão Nacional de Contabilidade: <http://www.cnc.min-financas.pt/POC/POContabilidade.pdf>
- [rei06] Reis, J. F. (2006). *Balancete*. Obtido em 17 de Fevereiro de 2009, de Balancete: http://www.google.pt/url?sa=t&source=web&ct=res&cd=1&url=http%3A%2F%2Fwww.estig.ipbeja.pt%2F~jfrpr%2FSebentas%2FCGII%2FSebenta%2520teoria%2520%25202006%25202007.doc&ei=0ZuaSZPIHMyL-gashfGQCQ&usg=AFQjCNGSnsYtonqz_avCyX4YjP0fHCNzlA&sig2=npZ6OLjlhhvM_bpFlZ9N
- [rod09] Rodrigues, J. A. (s.d.). *Controlo de Gestão e Performance*. Obtido em 16 de Fevereiro de 2009, de Controlo de Gestão e Performance: <http://www.indeg.org/cursos/mestradosexecutivos/contabilidade/controlo/>
- [tut08] Silverlight Tutorial: Home. (03 de Dezembro de 2008). Obtido em 16 de Janeiro de 2009, de Silverlight Tutorial: <http://www.learn-silverlight-tutorial.com/>
- [swa07] Swaminarayan, N. (25 de Julho de 2007). *Silverlight vs. Flash - An Analysis Report*. Obtido em 17 de Fevereiro de 2009, de The Official Microsoft Silverlight Site: <http://silverlight.net/forums/t/3015.aspx>
- [vol99] Volume de Negócios. (20 de Janeiro de 1999). Obtido em 19 de Janeiro de 2009, de Comissão de Normalização Contabilística: http://www.cnc.min-financas.pt/Entendimentos/volume%20negocios%20dc22_of_26_99.pdf
- [w3s08] w3schools. (1 de Fevereiro de 2008). *Flash in HTML*. Obtido em 16 de Fevereiro de 2009, de Flash in HTML: http://www.w3schools.com/flash/flash_inhtml.asp
- [wil08] Williams, J. R., Haka, S. F., Bettner, M. S., & Carcello, J. V. (2008). *Financial & Managerial Accounting*. MCGraw-Hill Irwin.