

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

REENGENHARIA DE SOFTWARE

Caso de Estudo da Fundação para a Divulgação das Tecnologias de Informação

Dissertação de Mestrado em Tecnologias da Informação e Comunicação

Ana Sofia de Azevedo Dinis



Vila Real, 2009

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

REENGENHARIA DE SOFTWARE

Caso de Estudo da Fundação para a Divulgação das Tecnologias de Informação

Dissertação de Mestrado em Tecnologias da Informação e Comunicação

Ana Sofia de Azevedo Dinis

Dissertação submetida à universidade de Trás-os-Montes e Alto Douro para cumprimento dos requisitos necessários para a obtenção do grau mestre em Tecnologias de Informação e Comunicação, elaborado sob orientação dos professores Luís Magalhães e Vítor Filipe.

Vila Real, 2009

A g

AGRADECIMENTOS

A presente dissertação é o culminar de um esforço na realização do Mestrado em Tecnologias de Informação e Comunicação.

Nesta etapa da minha vida, foram vários os que contribuíram directa ou indirectamente, para a sua concretização e conclusão deste trabalho, os quais passo a agradecer a seguir.

Aos professores Doutores Vítor Filipe e Luís Magalhães, pelo apoio e dedicação durante a orientação desta dissertação e pela confiança em mim depositada.

Ao meu namorado, Duarte pela paciência, apoio e compreensão nesta etapa da minha vida.

Aos meus pais, Maria do Carmo e António pelo apoio constante durante toda a minha vida.

À minha irmã Carmo, aos meus sobrinhos, em especial, à minha querida sobrinha Eduarda pelos momentos inesquecíveis de alegria.

A todos os meus amigos pelo companheirismo e pelos momentos de descontração.
Faço questão de não mencionar nomes para não esquecer ninguém.

À FDTI, por toda a compreensão e disponibilidade na realização deste trabalho. Sem
você era mais difícil!

A todos o meu muito obrigado!

Re

RESUMO

A reengenharia de software (RS) é um processo que permite que uma organização se ajuste de uma forma mais rápida às condições de constante mudança do mercado, inovadora para manter-se tecnologicamente actualizada nos seus produtos e serviços, fornecendo o máximo de qualidade. Esta propõe um método que promove uma mudança para alcançar bons resultados no desempenho organizacional, assim como privilegia a aprendizagem e o trabalho em equipa. O objectivo dessas actividades é o de criar versões dos programas existentes que apresentem mais qualidade e melhor manutenção, e que estejam de acordo com as necessidades da organização.

Na Fundação para a Divulgação de Tecnologias de informação (FDTI) ao longo dos anos foi desenvolvido um conjunto de sistemas de informação, sendo empregues no seu desenvolvimento os recursos disponíveis na época. Entretanto, esses sistemas estão a funcionar até aos dias de hoje, mas com o passar dos anos estes tornaram-se obsoletos, deixando de atender às novas necessidades da organização e dos seus utilizadores. Tal facto, fez com que a organização tomasse algumas medidas, de modo, a integrar numa plataforma única todas as aplicações existentes, estando estas agrupadas e subdivididas por módulos.

Neste contexto, este trabalho tem como objectivo principal aplicar o processo de reengenharia de software nos diversos sistemas da FDTI. No entanto, neste documento apresenta-se a aplicação das fases iniciais do processo de RS, com especial ênfase em duas aplicações do sistema existente.

Neste estudo, foi adoptado o modelo de RS proposto por Pressman, tendo sido realizadas as seguintes actividades: análise de inventário, reestruturação de documentos, engenharia reversa, reestruturação de dados e a engenharia progressiva. Como resultado foi proposta uma solução que integra numa plataforma única os serviços existentes na FDTI.

Ab

ABSTRACT

Software Re-engineering (SR) is a process that allows an organization to adjust more quickly to changing conditions of the market, innovative to keep technology up to date in their products and services, providing the highest quality. This proposes a method that promotes a shift to achieve good results in organizational performance, and focuses on learning and teamwork. The purpose of these activities is to create versions of existing programs which have more quality and better maintenance, and which are consistent with the needs of the organization.

The Dissemination of Information Technology Foundation (FDTI) over the years has developed a set of information systems, being used in its development resources available at the time. However, these systems are operating to this day, but over the years these have become obsolete and no longer meet the changing needs of the organization and its users. This made organizations to take measures in order to integrate all applications into a single platform, these being grouped and subdivided in modules.

In this context, this work aims to apply the process of software re-engineering in the various FDTI systems. However, this document presents the implementation of the early

stages of the SR process, with particular emphasis on two applications of the existing system.

In this study, we adopted the SR model proposed by Pressman in which the following activities were conducted: inventory analysis, document restructuring, reverse engineering, restructuring of data and progressive engineering. As a result, a solution that integrates all existing services of the FDTI on a single platform was proposed.

ÍG

ÍNDICE GERAL

Agradecimentos	iv
Resumo	v
Abstract	vi
Índice Geral	vii
Índice de Figuras	viii
Siglas	ix
1 Introdução.....	1
1.1 Enquadramento do estudo.....	1
1.2 Motivações e objectivos.....	0
1.3 Organização da dissertação.....	2
2 Reengenharia nas organizações	4
2.1 Reengenharia do processo de negócio	5
2.2 Reengenharia de software	8
2.3 Modelos de reengenharia de software.....	10

2.4	Motivações e limitações para adopção da reengenharia	17
3	Análise de requisitos do software.....	21
3.1	Análise de requisitos	21
3.2	Técnicas de modelação	26
3.2.1	Análise estruturada.....	26
3.2.2	UML.....	28
4	Reengenharia do sistema da FDTI	32
4.1	Organização	33
4.2	Análise do sistema actual.....	36
4.2.1	Análise de inventário	36
4.2.2	Reestruturação de documentos	40
4.2.3	Engenharia reversa.....	41
4.3	Aplicação GesPolos	42
4.4	Aplicação GesCDTI.....	47
5	Sistema proposto para a FDTI.....	52
5.1	Engenharia progressiva.....	52
5.2	Análise de requisitos	56
5.3	Diagramas do sistema	61
5.3.1	Diagramas de classes	61
5.3.2	Diagramas de sequência.....	64
5.3.3	Diagramas de actividades	66
5.4	Reestruturação de dados	68
6	Conclusão e trabalho futuro.....	70
	Bibliografia.....	72

Índice

ÍNDICE FIGURAS

Figura 1 - Perspectiva sociotécnica	4
Figura 2 - Modelo de reengenharia do processo de negócio	7
Figura 3 - Modelo de reengenharia de software	10
Figura 4 - Modelo de reengenharia	14
Figura 5 - Modelo de reengenharia de software	20
Figura 6 - Modelo de análise estruturada	27
Figura 7- Origens do UML	28
Figura 8 - Diagramas do UML.....	33
Figura 9 - Organograma da FDTI.....	35
Figura 10 - Rede de apropriação tecnológica da FDTI	31
Figura 11- Sistema de informação da FDTI	42
Figura 12 - Aplicações GesPolos e GesCDTI	43
Figura 13 - Diagrama casos-de-uso aplicação GesPolos.....	48

Figura 14 - Modelo de dados da aplicação GesPolos.....	50
Figura 15 - Diagrama casos-de-uso aplicação GesCDTI	52
Figura 16 - Modelo de dados da aplicação GesCDTI	51
Figura 17 - Plataforma única	56
Figura 18 - Diagrama de casos-de-uso módulo infraestrutura	64
Figura 19 - Diagrama de classes.....	67
Figura 20 - Diagrama de sequência - inserir pólo	64
Figura 21 - Diagrama de sequência - candidatar abertura de CDTI.....	65
Figura 22 - Diagrama de actividades - inserir polo	66
Figura 23 - Diagrama de actividades- candidatar abertura de CDTI.....	67
Figura 24 - Modelo de dados do módulo infraestrutura	69

SI

SIGLAS

Neste documento são utilizadas abreviaturas de designações comuns apenas apresentadas aquando da sua primeira utilização. As siglas utilizadas são:

CA - Concelho de Administração

CRM - Customer Relationship Management

UML - Unified Modeling Language

RS - Reengenharia de software

RPN - Reengenharia do processo de negócio

ES - Engenharia de software

CDTI- Centro de Divulgação de Tecnologias de Informação

FDTI - Fundação para a Divulgação das Tecnologias de Informação

TI - Tecnologias de Informação

SI - Sistemas de informação

CAPÍTULO 1

1 INTRODUÇÃO

1.1 ENQUADRAMENTO DO ESTUDO

O fenómeno da globalização, a competitividade e o aumento da exigência relativamente a produtos ou serviços tendem a que num futuro próximo, todos os aspectos das organizações possam influenciar o seu próprio posicionamento competitivo, em especial, a eficácia dos seus sistemas de informação (LANDON & LANDON, 2003).

Em muitos dos casos, como os sistemas de software apresentam um estado contínuo de mudança, a manutenção destes sistemas torna-se difícil, sendo necessário reconstruir o sistema partindo do sistema existente, recorrendo a algumas técnicas de Reengenharia de Software (RS), para que a manutenção seja mais flexível.

O termo reengenharia é universalmente aceite, estando esta designação relacionada com a reconstrução de algo do mundo real, e independentemente da sua aplicação, o seu principal propósito é a busca de melhorias que permitam produzir algo de melhor qualidade ou, pelo menos, de qualidade comparável ao produto inicial.

A reengenharia tem como principal objectivo melhorar um sistema de alguma maneira, através de alterações significativas que proporcionem melhoria, porém, sem alterar as suas funções (WARDEN, 2000).

Para se poder compreender de um modo mais claro o conceito de RS, é necessário consolidar-se outros conceitos relacionados com esta temática, tais como, engenharia de software, software e manutenção de software.

A engenharia de software é uma área do conhecimento da computação voltada para a especificação, desenvolvimento e manutenção dos sistemas de software encontrando-se envolvida nos aspectos tecnológicos do processo de desenvolvimento da RS. O conhecimento adquirido através dos sistemas antigos é usado como base da evolução contínua do software (GIMENES, 1994).

O software tornou-se a base de sustentação de muitas organizações dos mais diversos ramos de actuação espalhados pelo planeta, constituindo deste modo, um elemento estratégico de diferenciação de produtos e serviços, estando embutido em sistemas de variadas áreas, tais como, as ciências e as tecnologias. Assim como, é responsável pelos novos avanços que influenciam uma série de mercados, envolvendo desde a educação às organizações envolvidas com a engenharia genética (PRESSMAN, 2001).

A partir do momento que o sistema é utilizado, este ingressa num estado de contínua mudança, mesmo quando se utilizam as melhores técnicas no projecto, os sistemas vão-se tornando obsoletos em relação às novas tecnologias que são disponibilizadas no quotidiano.

As mudanças mais comuns, além das correcções de erros, são as migrações para novas plataformas, modificações para compatibilizar as mudanças de tecnologias ao nível do hardware, do software, assim como alterações de funcionalidades para atender às novas necessidades dos utilizadores.

As organizações demonstram interesse em manter esses sistemas de software em funcionamento, pois são considerados como propriedades que agregam a lógica de negócio codificada, investimento, anos de desenvolvimento e teste, experiências, estratégias organizacionais, entre outros aspectos.

As actividades de manutenção neste tipo de sistemas de informação são geralmente realizadas quando existe necessidade de:

- Adicionar uma funcionalidade que satisfaça as novas regras de negócio e novas políticas governamentais;

- Adaptar o sistema com novas tecnologias emergentes, tanto ao nível de hardware quanto de software;
- Efectuar a correcção de erros inseridos durante o desenvolvimento ou manutenção anterior no sistema;
- Adicionar melhorias no sistema a fim de facilitar manutenções futuras e melhorar a sua confiabilidade.

A actividade de manutenção consome 70% de todo o esforço de software despendido durante o ciclo de vida do software (PRESSMAN, 2005).

A variedade de problemas que envolve a manutenção de software cresce constantemente, sendo que as soluções não acompanham essa evolução, esses problemas são resultantes do código fonte e da documentação mal elaborada não facilitando a compreensão do sistema (CHIKOFFSKY & CROSS, 1990).

1.2 MOTIVAÇÕES E OBJECTIVOS

Actualmente, existem uma série de estudos que apontam um conjunto de alternativas para adequar sistemas de informação antigos para a actual realidade, uma delas é a RS.

Através do processo de reengenharia de software, o ambiente nativo do sistema de informação é preservado, as funcionalidades acondicionadas são reutilizadas total ou parcialmente e conseqüentemente, o tempo e os custos para a realização dessa reengenharia são reduzidos.

Os sistemas de software da FDTI estão a funcionar até aos dias de hoje, mas com o passar dos anos estes tornaram-se obsoletos, deixando de atender às novas necessidades da organização e dos seus utilizadores, existindo inúmeras aplicações, não funcionando de modo integrado. Existe a necessidade de integrar numa plataforma única todas as aplicações, funcionando esta por módulos.

O âmbito da elaboração deste trabalho deve-se à existência na FDTI de sistemas de informação antigos e da necessidade de os integrar numa plataforma única, onde se pretende integrar todos os dados e processos da organização. A integração pode ser vista sob a perspectiva funcional e sob a perspectiva sistemática, recorrendo ao processo de RS.

Além disso, neste trabalho é clarificado, o conceito de reengenharia nas organizações, em particular, a reengenharia de software, os diferentes métodos da reengenharia, assim como, será apresentado a reengenharia de software na FDTI.

O objectivo geral do trabalho visa aplicar o processo de reengenharia de software nos diversos sistemas da FDTI. No entanto, neste documento apresenta-se a aplicação das fases iniciais do processo de RS, com especial ênfase em duas aplicações do sistema existente. Neste estudo, foi adoptado o modelo de RS proposto por Pressman, tendo sido realizadas as seguintes actividades: análise de inventário, reestruturação de documentos, engenharia reversa, reestruturação de dados e a engenharia progressiva.

Como resultado foi proposta uma solução que integra numa plataforma única os serviços existentes na FDTI.

São ainda definidos os seguintes objectivos complementares para o trabalho:

- Analisar uma plataforma de serviços comuns;
- Uniformizar as regras de negócio;
- Adicionar novas funcionalidades;
- Melhorar a manutenção do sistema;
- Reestruturar os dados com base em critérios de modelação.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta secção apresenta sucintamente a estrutura e conteúdo do trabalho, na qual é possível identificar 6 capítulos, independentes entre si, mas que formam um todo coeso, seguindo um fio condutor desde um conjunto de reflexões iniciais até uma série de considerações finais e perspectivas do seu desenvolvimento futuro.

O primeiro capítulo, introdutório, começa por caracterizar globalmente a realidade das organizações, salientando a importância do processo de reengenharia de software no seu desenvolvimento. Os objectivos a atingir, a abordagem seguida e a organização da dissertação, encerram este capítulo.

Relativamente ao segundo capítulo é apresentado o conceito de reengenharia nas organizações, na qual se destaca, o conceito de reengenharia de software, a reengenharia dos processos de negócio, as etapas dos modelos de reengenharia, evidenciando o modelo seleccionado. Por fim, apresentam-se as principais razões da adopção e do fracasso da reengenharia de software.

No terceiro capítulo é abordada a análise de requisitos de software, no qual se destacam os requisitos funcionais e não funcionais, o levantamento de requisitos e as técnicas de modelação, uma das quais será utilizada no caso de estudo da FDTI.

O quarto capítulo, caso de estudo da FDTI, no qual é apresentada a organização, assim como é aplicado o processo de reengenharia do sistema, na qual é descrito a análise de inventário, é feita a reestruturação de documentos, engenharia reversa, e é mostrada toda a situação actual dos sistemas da FDTI.

No quinto capítulo é aplicado o método de engenharia progressiva, sendo apresentada uma proposta para o sistema futuro da FDTI, situação que foi possível através da realização de levantamento de requisitos e da aplicação das técnicas de modelação, assim como a reestruturação de dados.

Por último, no capítulo de conclusões e trabalho futuro, destacam-se os resultados obtidos com a realização deste trabalho, assim como o trabalho a realizar no futuro.

CAPÍTULO 2

2 REENGENHARIA NAS ORGANIZAÇÕES

Neste capítulo descrevem-se os conceitos de reengenharia do processo de negócio e reengenharia de software. São apresentados os modelos de reengenharia de software assim como as etapas que mais se destacam. Evoluindo na revisão, descrevem-se as razões de sucesso e de fracasso do processo de reengenharia.

A tecnologia e a organização devem ser ajustadas entre si até que se obtenha uma consonância entre os dois domínios, essa conformidade dá-se a partir de diversas alternativas de ambos os lados, até se chegar ao design final, como apresenta a Figura 1 (LANDON & LANDON, 2003).

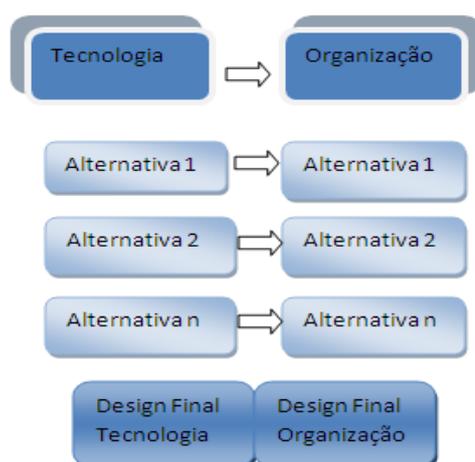


Figura 1 - Perspectiva sociotécnica (LANDON & LANDON, 2003)

Esta perspectiva sociotécnica, sugere que tanto a tecnologia como a organização sofrem modificações. Assim, a tecnologia deve ser adaptada e desenhada de modo a ajustar-se às necessidades da organização, em que todas as modificações devem ser devidamente planeadas de modo a que as vantagens na utilização de sistemas de software sejam alcançadas (LANDON & LANDON, 2003).

O processo de reengenharia na organização pode ocorrer em dois níveis de abstracção: ao nível do negócio, onde a reengenharia é focada nos processos de negócio com o intento de alterá-los, aumentando a competitividade da organização em determinada área; e ao nível do software, onde a reengenharia actua, analisando o sistema procura reestruturá-lo ou reconstruí-lo com maior qualidade (PRESSMAN, 2000).

2.1 REENGENHARIA DO PROCESSO DE NEGÓCIO

A reengenharia de processo de negócio é definida como a reavaliação e redesenho dos processos de negócio para alcançar melhorias de performance ao nível dos custos, da qualidade de serviço e da velocidade (JACOBSON, ERICSSON, & JACOBSON, 1994).

Esta deve ocorrer em todas as áreas do negócio, identificando os grandes objectivos da organização assim como, o detalhe do processo de negócio, sub-processos de negócio e as tarefas relacionadas com cada processo.

O seu objectivo visa a automatização das operações existentes, assim como melhorar e redesenhar os processos de negócio enfatizando o entendimento das expectativas dos clientes com os novos produtos e serviços.

Como na maioria das actividades de engenharia, a reengenharia do processo de negócio é interactiva. As metas de negócio e os processos que as alcançam precisam de ser adaptados ao ambiente de negócios em constante modificação. Por essa razão, não existe um ponto inicial nem um ponto final para a Reengenharia do Processo de negócio (RPN), é um processo evolutivo (PRESSMAN, 2000).

A Figura 2 apresenta as 6 actividades da RPN (PRESSMAN, 2000):

- **Definição do negócio** - Esta fase destaca os objectivos do negócio tais como, a redução de custos, redução do tempo, aumento de qualidade, desenvolvimento e o aperfeiçoamento dos recursos humanos.
- **Identificação dos processos** - Os processos são ordenados por importância, necessidade de mudança ou por outro critério.
- **Avaliação de processos** - Os processos existentes são analisados e avaliados, as tarefas dos processos são identificadas, assim como o custo, tempo consumido e os problemas de qualidade são isolados.
- **Design e especificação de processos** - São elaborados casos de uso para cada processo que deve ser reestruturado. A partir dos casos de uso são definidas novas tarefas para cada processo.
- **Prototipagem** – Os processos devem ser testados para que sejam realizadas melhorias nos mesmos, antes de serem integrados nos novos modelos.
- **Refinamento e instanciação** - Com base nos resultados dos testes da prototipagem os processos devem ser melhorados e implementados no negócio.

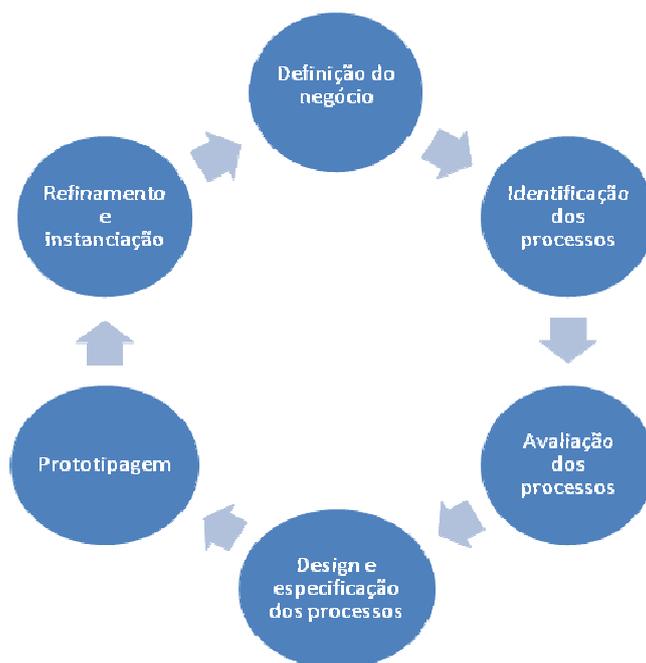


Figura 2 – Modelo de reengenharia do processo de negócio (PRESSMAN, 2000)

A reengenharia de processo de negócio pode resultar, se for aplicado por recursos humanos motivados e experientes que sabem que o processo de reengenharia é uma actividade contínua, logo pode ser avaliada dentro do contexto da estratégia de todo o negócio, assim como, as prioridades da reengenharia de software podem ser estabelecidas de forma inteligente (PRESSMAN, 2000).

2.2 REENGENHARIA DE SOFTWARE

Quando um software começa a demonstrar sinais de que já não corresponde eficientemente aos requisitos da organização, ou seja, quando os requisitos da organização mudam muito em relação ao que foi concebido para o software, surge a necessidade de se construir um sistema similar ou adaptar o sistema existente, adicionando funcionalidades, melhorando a performance e a sua consequente manutenção (FLYNN, 1994).

Os sistemas de software requerem constantes modificações, tais como, a correção de erros, melhoria do desempenho das aplicações, adição de novas funcionalidades, assim como a adaptação de novas plataformas de hardware e de software.

A dificuldade em manter o software evidencia-se uma vez que as organizações querem realizar esta manutenção a um custo relativamente baixo. Este facto tem motivado os investigadores a estudarem soluções que diminuam os custos de desenvolvimento, garantam a durabilidade do sistema e facilitem a sua manutenção (FUKUDA, 2000) (PRADO & NOVAIS, 2001).

A actividade de reengenharia de software absorve muitos recursos, por isso a organização necessita de uma estratégia prática (PRESSMAN, 2006). Antes de iniciar o processo de reengenharia de software é necessário evidenciar alguns aspectos, tais como, os que se seguem:

- Analisar o software criando uma lista de critérios, de modo a que o processo de reengenharia seja sistemático.
- Avaliar o nível de reengenharia, demorando muito menos tempo, assim o custo é reduzido.
- Analisar o sistema antigo, pois serve de base para a aplicação do processo de reengenharia.

A reengenharia é uma solução que deve ser considerada pelas empresas, pois os riscos de desenvolvimento e implementação de um novo sistema podem ser muito avultados, inviabilizando a implementação do mesmo. Além disso estudos têm demonstrado que a reengenharia, quando aplicada apropriadamente, geralmente promove custo efectivo e menos riscos que desenvolver um novo sistema (Ranson & Waren, 2002).

2.3 MODELOS DE REENGENHARIA DE SOFTWARE

Nesta secção são apresentados dois modelos de RS, o modelo defendido por Pressman e o modelo defendido por Lemos. Um destes será usado como técnica de RS no caso de estudo da FDTI.

Segundo Pressman, o modelo de RS pode ser dividido em 6 actividades (Figura 3):

- Análise de Inventário;
- Reestruturação de Documentos;
- Engenharia Reversa;
- Reestruturação de Código;
- Reestruturação de Dados.
- Engenharia Progressiva.

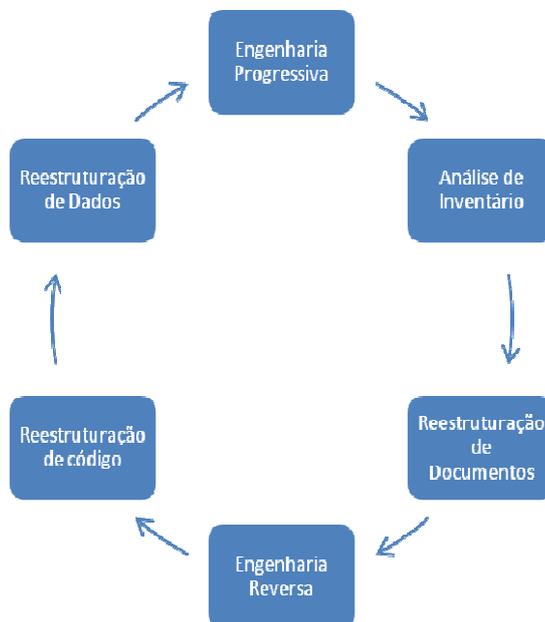


Figura 3- Modelo de reengenharia de software (PRESSMAN, 2006)

Estas actividades, usualmente, ocorrem em sequência linear, mas nem sempre é o caso. O paradigma apresentado na Figura 3 é um modelo cíclico, isso significa que cada uma das actividades apresentadas como parte do paradigma pode ser revisitada. Para

qualquer ciclo particular, o processo pode terminar depois de qualquer uma dessas actividades (PRESSMAN, 2006).

De seguida apresenta-se uma descrição de cada uma das 6 actividades.

- **Análise do Inventário** -As organizações devem ter um inventário de todas as aplicações de software. O inventário de software não é nada mais do que uma lista com alguns detalhes dos softwares de uma organização. A lista pode conter informações como tamanho, idade, nível de necessidade da ferramenta para a organização, etc. Ao analisar o inventário pode-se chegar a uma lista dos sistemas que precisam passar por um processo de reengenharia e até mesmo quais são os mais prioritários, podendo-se assim direccionar esforços para os sistemas mais críticos.

- **Reestruturação de Documentos** - A criação de documentação é uma actividade demorada. Em alguns casos essa abordagem é correcta, uma vez que não é possível recriar documentação para centenas de programas de computador. Nesta actividade existem 3 opções (PRESSMAN, 2006):
 1. Se o sistema é estável, as alterações a realizar são inexistentes, este é apenas mais um de entre outros sistemas a serem analisados, logo não é necessário perder tempo na documentação do sistema antigo.

 2. Se o sistema não se encontra completamente mudado, apenas parte dele está a ser analisado, não existe necessidade de documentar todo o sistema, apenas parte do mesmo.

 3. Se o sistema é crítico para o negócio como um todo, então é altamente recomendado que se produza documentação para todo o sistema antigo.

- **Engenharia Reversa** - A engenharia reversa teve origem na indústria de hardware, onde organizações desmanchavam produtos de concorrentes a fim de entender os seus segredos. Porém, sem a documentação de projecto dos produtos era extremamente difícil entender o seu funcionamento, sendo necessário

realizar uma análise detalhada e gerar a documentação necessária para a compreensão do produto. Em essência, a engenharia reversa, parte de um sistema existente, analisando-o de forma a identificar os seus componentes e as inter-relações.

A engenharia reversa consiste numa actividade similar à tradicional engenharia de software, onde os dados iniciais provêm da análise de um sistema antigo e o objectivo não é criar um novo sistema ou simplesmente dar uma cara nova ao sistema antigo, mas sim adaptar o sistema antigo à realidade actual dos utilizadores adicionando novas funcionalidades atendendo aos novos requisitos.

O processo de engenharia reversa inicia-se no código mal estruturado ou simplesmente não documentado. Quando necessário, o primeiro passo deve ser uma estruturação do código a fim de facilitar a sua compreensão. O passo seguinte é a extracção de abstracções, esta actividade consiste em se extrair do código informações significantes sobre o processamento, a interface com o utilizador e a base de dados do sistema.

Para se criar a abstracção do processamento do sistema, o código pode ser analisado em diferentes níveis, o sistema e os seus módulos, para compreensão de componentes e padrões.

A abstracção da base de dados pode ser feita em dois níveis. Em primeiro lugar são identificados os conceitos armazenados de uma forma separada. Em seguida, parte-se para a identificação das ligações entre os conceitos com o fim de se ter uma visão geral da estrutura da base de dados para poder, enfim, remodelá-la num paradigma moderno de bases de dados.

A reestruturação do software modifica o código-fonte e/ou os dados num esforço para tornar as modificações futuras mais fáceis. Em geral, a reestruturação não modifica a arquitectura global do programa, concentra-se no detalhe de módulos individuais e nas estruturas de dados locais definidas nos módulos.

- **Reestruturação de código** - Alguns dos sistemas antigos têm uma arquitectura de programa relativamente sólida, mas alguns módulos foram codificados de um modo que se torna difícil entendê-los, testá-los e mantê-los. Em alguns casos, o código dos módulos pode ser reestruturado. Para realizar essa actividade, o código fonte é analisado usando uma ferramenta de reestruturação. A estrutura dos programas é registada e o código é então reestruturado, revisto e testado para garantir que nenhuma anomalia foi inserida, sendo a documentação interna do código actualizada.
- **Reestruturação de dados** - A arquitectura de dados actual deve ser analisada e os modelos de dados necessários definidos, deste modo, quando a estrutura de dados é ineficiente, os dados passam por reengenharia. Como a arquitectura de dados tem uma forte influência na arquitectura do programa e nos algoritmos que o constituem, as modificações resultarão invariavelmente tanto em modificações arquitecturais quanto de código.

Um programa com uma arquitectura de dados fraca é bastante difícil de se adaptar. A reestruturação de dados consiste em se fazer uma engenharia reversa da estrutura de dados para, posteriormente, se adaptar ou recriar toda a estrutura de forma a gerar uma arquitectura mais sólida. É claro que normalmente alterações no nível de dados resultarão na necessidade de alterações no código que os manipula.

- **Engenharia Progressiva (Forward)** - A engenharia progressiva também designada de renovação ou recomposição, recupera a informação de projecto do software existente e usa essa informação para alterar ou reconstruir o sistema existente num esforço para aperfeiçoar a sua qualidade global. Na maioria dos casos, o software reimplementa a função do sistema existente e também adiciona novas funções contribuindo para uma melhoria no desempenho geral (CHIKOFSKY & CROSS, 1990).

Na Figura 4 apresenta-se um modelo alternativo de processo de reengenharia proposto por Lemos (LEMOS, 2002). A metodologia baseia-se em 7 etapas: definição dos objectivos da reengenharia, constituição da equipa de reengenharia, análise do sistema antigo, especificação dos novos requisitos do sistema, definição do processo de implementação, desenvolvimento, testes e formação.

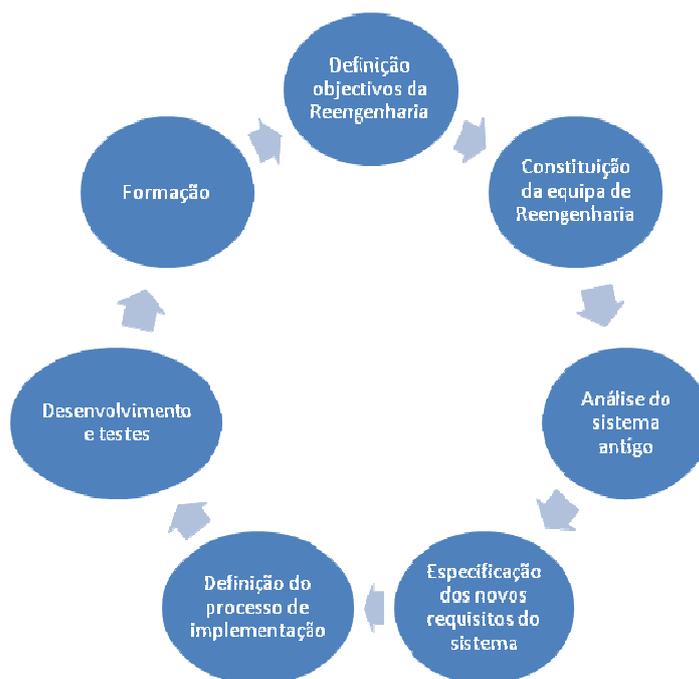


Figura 4 – Modelo de reengenharia (LEMOS, 2002)

Segue-se a descrição das 7 fases do modelo de reengenharia:

- **Definição dos objectivos da reengenharia e do negócio** - Nesta fase é fundamental o envolvimento da administração para que os objectivos táticos/estratégicos sejam bem difundidos no projecto de reengenharia.
- **Constituição da equipa de reengenharia** - Deve ser uma equipa integrada e multidisciplinar composta por programadores, utilizadores e gestores de negócio do departamento de Sistemas de Informação (SI). É essencial que os utilizadores

finais estejam bem representados para que todas as necessidades de informação sejam bem definidas e representativas da realidade operacional.

- **Análise do Sistema Antigo** - É uma das fases mais longas do projecto, dado que a maior parte das vezes parte-se de uma fraca documentação e do mapeamento dos SI existentes na organização, sentindo-se muitas dificuldades na compreensão do funcionamento desses sistemas. Esta fase é crucial pois determina e condiciona a execução da reengenharia.
- **Especificação dos novos requisitos do sistema** - Identificados os objectivos e efectuado o re-mapeamento dos programas, dá-se início à especificação funcional e técnica do novo sistema com os elementos funcionais e técnicos do projecto.
- **Definição do processo de implementação** - O processo de implementação pode ser efectuado segundo duas formas distintas, através de um processo em bloco, onde todas as mudanças no sistema antigo são postas em prática, validadas de uma só vez, ou através de um processo incremental onde à medida que as mudanças são desenvolvidas estas são validadas e colocadas em produção. A escolha do processo de implementação depende da especificidade e risco do projecto em causa.
- **Desenvolvimento e testes** - É uma etapa crítica na medida em que é difícil provar que o novo sistema é funcionalmente equivalente ao software antigo.
- **Formação** - A formação é uma peça fundamental para a consolidação do projecto a todos aqueles que irão utilizar o novo sistema.

Apesar de as tapas apresentarem designações diferentes do modelo proposto por Pressman, algumas têm grandes semelhanças em termos de objectivos e abordagem seguida. Os objectivos de ambos são idênticos, visto que pretendem compreender o funcionamento do sistema antigo, para assim encontrar a solução mais adequada ao problema que se pretende resolver.

O modelo seleccionado para abordarmos o sistema da FDTI, é o modelo de Pressman devido ao facto de neste modelo ser dada uma maior ênfase à análise do sistema antigo, conseguindo desta forma compreender o sistema complexo que se tem em mãos.

No modelo proposto por Lemos, este evidencia a relevância da definição dos objectivos da reengenharia, assim como a constituição da equipa de desenvolvimento, dando mais importância ao sistema novo do que ao sistema antigo.

2.4 MOTIVAÇÕES E LIMITAÇÕES PARA ADOÇÃO DA REENGENHARIA

Nesta secção são clarificadas as razões que levam as organizações a adoptarem esta metodologia de reengenharia nas organizações, assim como as razões que levam ao fracasso da mesma.

Segundo Flynn (FLYNN, 1994) alguns factores para o sucesso de um projecto de reengenharia são os seguintes:

- **Motivação** - Os motivos para iniciar um projecto de reengenharia devem estar bem definidos, devido a este facto, a administração deve estar absolutamente convencida de que os esforços em reengenharia levarão a consideráveis melhorias, assim como, devem entender que algumas estruturas serão eliminadas por bons motivos. Para garantir o sucesso da reengenharia, a administração deve-se comprometer e a organização deve ser totalmente envolvida no processo de reengenharia. Além disso, os colaboradores envolvidos devem ter plena consciência das razões pelas quais o projecto está a ser desenvolvido, ou seja, devem compreender os problemas a resolver, assim como também aceitar as suas novas tarefas.
- **Liderança** - Na organização o gestor de projecto deve liderar o mesmo e assumir a responsabilidade, antes de começar o projecto, o líder deve ter consciência das dificuldades do mesmo, devendo resistir às pressões da organização e deve convencer todos de que o projecto não é apenas vital, mas essencial para a sobrevivência da organização.
- **Autoridade em toda a organização** - A organização deve sentir-se envolvida, os colaboradores devem ser aconselhados a assumir o trabalho que trará as alterações. Alguns podem trazer problemas, seja por resistirem a mudanças em processos definidos anteriormente, seja por estarem acomodados com os

processos antigos. Estes serão os mais difíceis de se convencer da necessidade e da importância das mudanças.

- **Visão** - Os novos objectivos da organização devem ser eloquentes e facilmente compreendidos por todos os envolvidos.
- **Foco** - As mudanças na organização salientam os objectivos de maior prioridade e os recursos destinados atingir os mesmos.
- **Produtos tangíveis** - Os resultados da reengenharia são concretos, executáveis, como os objectivos, missões, novos fluxos de trabalho, modelos de processo, plano da organização e modelo de dados.
- **Suporte tecnológico** - Suporte na forma de métodos e ferramentas são indispensáveis para o trabalho de reengenharia. A engenharia de negócios geralmente envolve a construção de um sistema de informação para o novo negócio, é uma área de risco, pois o sistema de informação deve ser construído para evitar futuros problemas.
- **Aconselhamento de um consultor** - A consultoria pode auxiliar no processo de reengenharia, porém deve-se tomar cuidado pois o consultor deve conhecer muito bem o modelo de negócio da organização.
- **Aceitar o risco** - Os riscos de um projecto de reengenharia podem ser diminuídos drasticamente se for usado um processo formal, a orientação auxilia no processo de reengenharia de negócio.

Analisando o descrito acima, pode-se dizer que as motivações (ou objetivos pretendidos) que levam as organizações a adoptarem a reengenharia estão relacionadas com a motivação dos intervenientes e a gestão de todo o projecto, assim como o suporte tecnológico.

Os motivos que levam os projectos de reengenharia ao fracasso são variados, mas os que se destacam são os que se seguem (Prado & Novais, 2001):

- **Adopção de uma estratégia de reengenharia defeituosa** - Recorrendo-se ao uso inapropriado de consultoria externa ou serviços. Estes serviços podem ser de extrema utilidade para a resolução de questões técnicas, aquisição de conhecimento do negócio, etc, porém, em geral o trabalho de outsourcing deve ser monitorizado mais de perto para evitar a ocorrência de problemas. A não contratação de um consultor ou serviço outsourcing quando necessário, pode ser tão determinante para o fracasso quando uma contratação mal elaborada.
- **Tecnologias antigas cujo programa de formação é inadequado** - O processo de reengenharia é acompanhado por mudanças de tecnologia, metodologias, equipamentos, paradigmas e até mesmo vocabulários. É extremamente importante um eficaz programa de formação para deste modo preparar e motivar o pessoal da equipa para realizar a migração para uma nova plataforma.
- **A organização não possui o sistema antigo sob controlo** – É necessário o conhecimento das mudanças feitas no sistema, dos tipos de requisitos alterados, do grau de dificuldade e da importância das alterações assim como da existência de métricas sobre as actividades de manutenção, características de um sistema sob controlo.
- **Inexistência de levantamento e validação de requisitos** – Por vezes surgem dificuldades nesta tarefa pois o sistema, em geral, ajuda pouco, já que normalmente foi baseado em requisitos antigos, deste modo deve-se levantar os novos requisitos para o sistema.
- **Avaliação apropriada da arquitectura do software** - Serve de base para a definição da abordagem da reengenharia, se a arquitectura antiga for considerada inviável, deve-se construir um novo sistema completo, caso contrário, deve ser considerada parcial ou totalmente viável, desde que a compreensão da arquitectura seja profunda, de outro modo o melhor é construir uma nova arquitectura.

- **Inexistência de documentação** - A falta de documentação para a realização da reengenharia aumenta muito as hipóteses do projecto tornar-se um fracasso. Sem um eficaz processo dificilmente uma boa equipa com boas ferramentas de desenvolvimento produzirá um bom resultado.
- **Inexistência do planeamento adequado** - O plano pode ser feito mas não é documentado, ficando apenas na mente de alguns recursos humanos. O planeamento pode ser feito mas não comunicado a todos os interessados. Também pode ocorrer que o plano seja incompleto, ou que não se disponibilizem recursos para implementá-lo no tempo acordado. Em qualquer um desses casos os riscos de falha crescem bastante.
- **A gestão de projecto necessita de comprometimento a longo prazo** - Uma gestão de projecto de reengenharia é muito importante para diminuir a hipótese de ocorrência de erros que, se encontrados apenas em etapas mais avançadas do desenvolvimento, serão extremamente caros para corrigir, por essa razão o gestor de projecto deve estar concentrado no projecto.
- **A gestão de projecto predetermina questões técnicas** - Quando um membro da gestão indica as tecnologias usadas, custos, agenda e questões de performance sem a intervenção do pessoal da área técnica, a hipótese de tais decisões falharem é bastante elevada, podendo prejudicar todo projecto.

CAPÍTULO 3

3 ANÁLISE DE REQUISITOS DO SOFTWARE

3.1 ANÁLISE DE REQUISITOS

O principal objectivo da análise é produzir uma pormenorização do sistema na qual se define a estrutura do problema a ser resolvido de acordo com a visão do utilizador.

Esta actividade envolve o conjunto de trabalho da equipa técnica com os utilizadores do sistema a fim de estabelecer o domínio da aplicação do sistema, assim como os serviços que este sistema deve prever e as respectivas restrições.

A análise de requisitos do software é uma fase crítica no desenvolvimento de sistemas de informação, visto que afecta as fases de desenvolvimento seguintes. É considerada como uma fase crítica, devido a problemas de comunicação entre os utilizadores e a equipa de desenvolvimento, às mudanças sucessivas dos requisitos do sistema assim como a aplicação de técnicas inadequadas de avaliação (GUEDES, 2008).

Muitas das vezes a linguagem entre o utilizador e o responsável pelo desenvolvimento é tão diferente tornando a comunicação ineficaz. Os requisitos, apresentam um carácter inconstante pois são modificados ao longo do desenvolvimento do sistema de informação.

A análise de requisitos envolve os utilizadores finais, os que participam e utilizam o sistema. Esta actividade é difícil de levar a cabo especialmente devido aos problemas que se seguem:

- Os utilizadores sabem em linhas gerais o que o sistema deverá satisfazer, mas por vezes não conseguem expressar-se do melhor modo.
- Os utilizadores procuram transmitir os requisitos com os seus próprios termos e conhecimentos implícitos da sua actividade, o que pressupõe que o analista saiba do que eles estão a falar.
- Diferentes utilizadores apresentam requisitos idênticos, mas o modo como os expressam é diferente podendo deste modo enganar o analista.
- Para o analista é complicado a compreensão do negócio da organização apenas com a informação do utilizador e assim determinar os requisitos do sistema;
- O analista pode ficar sobrecarregado de detalhes que não são relevantes para o novo sistema;
- O documento que define os detalhes de um novo sistema (especificação funcional) efectivamente é um acordo entre os departamentos envolvidos e o grupo de desenvolvimento de sistema. Se o documento de especificação puder ser escrito de forma a fazer sentido para os utilizadores, poderá não ser muito útil para os analistas e programadores que irão construir o sistema.

Mesmo utilizando as melhores ferramentas analíticas possíveis, alguns dos problemas acima estarão sempre presentes, pois não existem ferramentas analíticas que possibilitem ao analista saber o que o utilizador pensa mas não diz. Não há como mostrar um modelo concreto e claro do sistema para os utilizadores, pois é difícil para os utilizadores imaginar o que o novo sistema lhes fornecerá até que esteja realmente em funcionamento (PILONE & PITMAN, 2006).

A análise sob diferentes perspectivas é importante pois não existe uma única forma de se analisar os requisitos do sistema. Assim é necessário compreender o domínio do sistema, validar os requisitos do sistema, definir e especificar os requisitos, resolver os conflitos que possam surgir ao longo da especificação, classificar os requisitos, como sendo requisitos funcionais ou não funcionais (Figura 5).



Figura 5- Análise de requisitos

Os requisitos, de modo geral, podem ser classificados em dois grandes grupos: os requisitos funcionais e os não funcionais.

- **Requisitos funcionais** - Os requisitos funcionais são aqueles que descrevem o comportamento do sistema, as suas ações para cada entrada, ou seja, é aquilo que descreve o que tem que ser feito pelo sistema. São a razão do projecto, já que descrevem as funcionalidades que o sistema deve dispor.

- **Requisitos não funcionais** - Os requisitos não funcionais são aqueles que expressam como deve ser feito (não confundir requisitos não funcionais com design). Em geral se relacionam com padrões de qualidade como confiabilidade, performance, robustez, etc. São muito importantes, pois definem se o sistema será eficiente para a tarefa que se propõe a fazer ou não. Um sistema ineficiente certamente não será usado. Neles também são apresentados restrições e especificações de casos-de-uso para os requisitos funcionais (WAZLAWICK, 2005).

Os requisitos funcionais, apresentam as funções a serem desempenhadas, tais como:

- Funções aplicacionais;
- Funções de controlo;
- Funções para estabelecer interfaces.

Nos requisitos funcionais é necessário saber quais os tipos de fluxos de dados de entrada e de saída que devem ser considerados, os que devem obedecer a tratamento de erros, como e quando deve ser feito o controlo de integridade, a que sobrecarga de utilização deve dar resposta, qual a taxa de crescimento que o sistema deverá suportar, exigências de recuperação em caso de falha, necessidade de reconhecimento automático de erros, entre outros.

Relativamente às funções que se estabelecem entre a interface e o utilizador, deve-se ter a descrição por cada interface, um conjunto de informações tais como, o tipo de interface, informação a ser transmitida, desempenho na transmissão dos dados, protocolo de comunicação, serviços e produtos de comunicação e mecanismos para acessos concorrentes.

Os requisitos não-funcionais representam a qualidade global de um software, tal como a manutenção, usabilidade, desempenho, custos entre outros. Normalmente estes requisitos são descritos de maneira informal, sendo difíceis de validar.

Para realizar o levantamento de requisitos, é necessário que o analista produza alguns documentos, os quais visam aspectos, tais como, os que se seguem (WAZLAWICK, 2005):

- **Visão geral do sistema ou sumário executivo** - Descreve as principais ideias do cliente sobre o sistema, não precisa ser longo. Em geral, mais do que duas páginas já inclui detalhes desnecessários.
- **Requisitos funcionais e não funcionais** - Regista-se, respectivamente, o que o sistema deve fazer e como devem ser feitas. É desejável que esta actividade seja realizada pela equipa de desenvolvimento juntamente com clientes e utilizadores.

- **Glossário** - Documento em que são definidos os termos técnicos do domínio da aplicação. Muitas vezes uma mesma palavra pode induzir a diferentes interpretações em diferentes situações, duas palavras que para o senso comum são consideradas sinónimos podem não sê-lo no domínio da aplicação.
- **Análise e controlo de riscos** - Documento que demonstra os principais riscos no desenvolvimento. Muitas vezes não é possível prever se um risco pode ser neutralizado logo no início do desenvolvimento, mas pode-se pelo menos prever a existência de mecanismos de controlo.
- **Protótipos e provas** - Quando for necessário verificar e esclarecer alguns requisitos, pode-se produzir um protótipo do tipo “throw-away”, ou seja, cujo código não será usado no desenvolvimento. O levantamento de requisitos pode ser feito, numa reunião com a equipa de desenvolvimento, utilizadores e clientes.

3.2 TÉCNICAS DE MODELAÇÃO

As técnicas de modelação abordadas nesta secção são a análise estruturada e UML. A análise estruturada é uma actividade de construção de modelos com a finalidade de retratar o fluxo e o conteúdo das informações utilizadas pelo sistema, decompor o sistema em partições funcionais e comportamentais e descrever a essência daquilo que será construído. A Unified Modeling Language (UML) é uma linguagem de modelação orientada a objectos, auxilia a visualizar o modelo e a comunicação entre objectos. A UML é a base para muitas ferramentas de desenvolvimento, incluindo modelação visual, simulações e ambientes de desenvolvimento.

3.2.1 ANÁLISE ESTRUTURADA

A análise estruturada de sistemas compõe-se de um conjunto de técnicas e ferramentas, em constante evolução. O seu conceito fundamental é a construção de um modelo lógico do sistema, utilizando técnicas gráficas capazes de levar utilizadores e analistas a formarem um quadro claro e geral do sistema e de como as suas partes se encaixam para atender às necessidades daqueles que dele precisam (NUNES & O'NEILL, 2003).

Na Figura 6 apresenta-se um esquema dos modelos da análise estruturada, assim como os respectivos diagramas. Assim o Diagrama Fluxo de Dados (DFD), especifica os processos, fazendo parte do modelo funcional do sistema. O Diagrama Transições e Estados (DTE/DCVE), permite a especificação do controlo, correspondendo ao modelo comportamental. Por fim o Diagrama Entidades Relacionamentos (DER), permite especificar os dados, fazendo parte do modelo de dados.

O modelo funcional, DFD, apresenta um desenvolvimento do geral para o particular do sistema. Começa-se com um diagrama geral de fluxo de informações e depois parte-se

para um refinamento sucessivo, através da construção de diagrama de fluxo de informações detalhadas, definindo o que o sistema deve fazer. Esta técnica é muito útil quando se pretende determinar as entradas e saídas do sistema.

O modelo de dados, DER, é usado para especificar os dados e representação conceptual do negócio.

O modelo comportamental, DTE/DCVE é uma representação do estado ou situação no decorrer da execução de processos de um sistema.

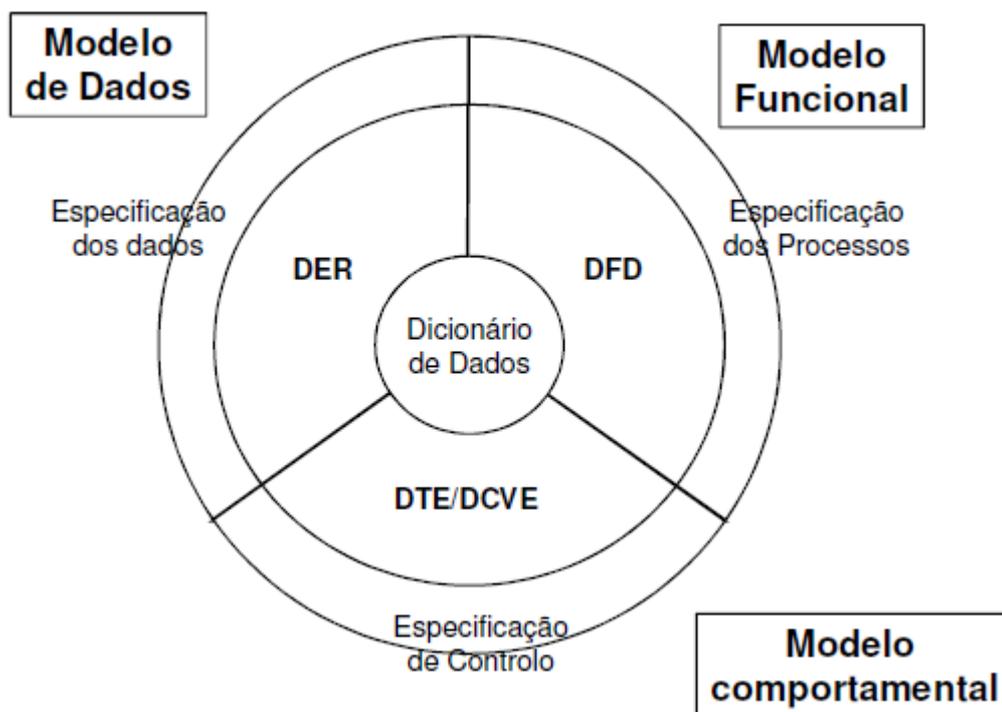


Figura 6 - Modelo de análise estruturada

3.2.2 UML

A UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma da orientação a objectos. Esta linguagem tornou-se nos últimos anos a linguagem padrão de modelação de software adoptada pela Engenharia de software.

Esta linguagem de modelação, tem como objectivo auxiliar os analistas e engenheiros de software a definir as características do software, tais como os seus requisitos, o seu comportamento, a sua estrutura lógica, a dinâmica dos seus processos e até mesmo as necessidades físicas em relação ao equipamento sobre o qual o sistema está a ser implementado. Todas essas características são definidas por meio de UML antes de o software começar realmente a ser desenvolvido.

A UML surgiu de três metodologias de modelação: o método de Booch, o método Object Modeling Technique (OMT) de Jacobson e o método Object.Oriented Software Engineering (OOSE) de Rumbaugh (Figura 7).

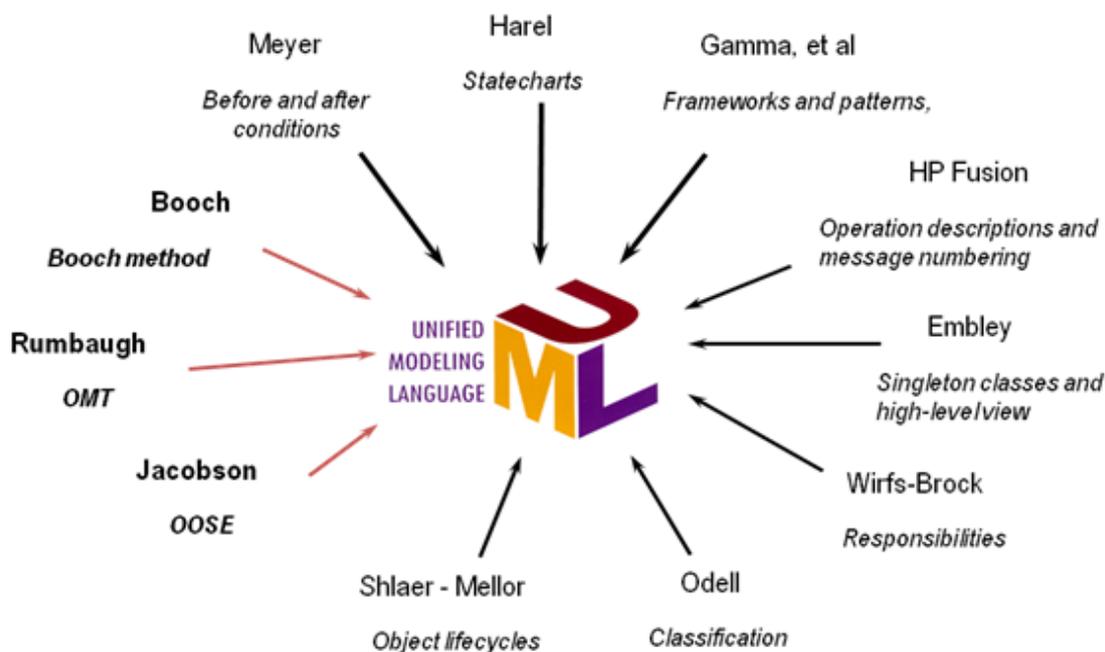


Figura 7- Origens do UML (SILVA & VIDEIRA, 2001)

A razão da existência de tantos diagramas é fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos. Procura-se deste modo que cada diagrama complete os outros. Cada diagrama UML analisa o sistema ou parte dele, sob uma determinada perspectiva.

Na Figura 8 tem-se os diagramas UML, os quais destacam-se o diagrama de casos de uso, diagramas de classes, diagrama de sequência. Alguns diagramas focam o sistema de forma geral, apresentando uma visão externa do sistema, como é o objectivo do diagrama de casos de uso. Paralelamente, outros diagramas apresentam uma visão mais pormenorizada do software, apresentando deste modo uma focalização mais técnica assim como as características específicas do sistema.



Figura 8 – Diagramas da UML (SILVA & VIDEIRA, 2001).

Verifica-se um grande défice de comunicação entre as comunidades de engenharia de software e de negócio, conduzindo a uma situação em que a documentação gerada pelas equipas de engenharia de software não é exactamente aquilo sugerido pelas equipas de negócio. Uma forma de solucionar esse problema é estabelecer uma linguagem comum às duas comunidades, recorrendo a diagramas de casos de uso. Desta forma garante-se um entendimento comum entre todos os interessados (RATIONAL, 2000).

O diagrama de casos de uso apresenta uma linguagem simples e de fácil compreensão para que os utilizadores possam ter a ideia geral de como o sistema se vai comportar. Este pretende identificar os actores (utilizadores e outros softwares que interagem com o sistema ou até mesmo algum hardware especial) que vão utilizar de algum modo o software, bem como os serviços, ou seja as operações que o sistema vai disponibilizar aos actores. Este diagrama é o mais geral e informal da UML, é utilizado para auxiliar no levantamento e análise de requisitos, em que são determinadas as necessidades do utilizador, e na compreensão de um sistema como um todo, embora venha a ser consultado durante todo o processo de modelação e serve de base para a construção dos outros diagramas.

Os diagramas de casos de uso representam os principais processos de negócio da organização e abrangem, em geral, um considerável número de requisitos funcionais. Inicialmente elabora-se uma lista de casos de uso, cuja elaboração começou na modelação do negócio, a qual contém os actores e a lista dos requisitos funcionais.

Os diagramas de classes ilustram atributos e operações de uma classe e as restrições como que os objectos podem ser ligados, descrevem também os tipos de objectos no sistema e os relacionamentos entre estes objectos que podem ser, associações e abstracções.

Este é o diagrama mais utilizado e com maior importância da UML, serve de apoio para a maioria dos outros diagramas, como o próprio nome indica, este diagrama define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos possuídos por cada classe, além de estabelecer com classes e relacionam-se entre si.

As informações são tratadas pelo sistema, em geral, os conceitos sofrem operações de manutenção, que podem ser inserção, alteração, remoção e consulta.

A correcta identificação dos conceitos e operações de manutenção pode ser facilitada, com a construção de um diagrama de classes que contém os conceitos e associações que constituem a informação a ser armazenada pelo sistema (WAZLAWICK, 2005).

O diagrama de sequência dá ênfase a ordenação temporal em que as mensagens são trocadas entre os objectos de um sistema. Este diagrama é construído a partir do Diagrama de Casos de Usos, definindo-se, primeiro, qual o papel do sistema (Caso de Uso), e só depois, é definido como o software realizará o seu papel (Sequência de operações).

Os diagramas de sequência, representam a sequência de processos (mais especificamente as mensagens transmitidas entre objectos) num programa de computador, estes diagramas representam essa informação de forma simples e lógica, descrevendo o modo como os objectos colaboram ao longo do tempo, registando o comportamento de um caso de uso exibindo as mensagens tocadas entre ambos.

CAPÍTULO 4

4 REENGENHARIA DO SISTEMA DA FDTI

Após no capítulo 2 terem sido abordados os vários conceitos relacionados com a Reengenharia de Software, no presente capítulo apresenta-se o modelo de reengenharia de software aplicado na FDTI.

Iniciou-se o processo de reengenharia com a Análise de Inventário para identificar as aplicações existentes e seleccionar aquelas onde se iria actuar. De seguida realizou-se a Reestruturação dos Documentos, assim como a Engenharia Reversa, descritos neste capítulo.

A FDTI é uma organização representada por várias áreas organizacionais, das quais neste trabalho se destaca a área de Sistemas de Informação, sobre o qual incide este trabalho. Como este trabalho tem um carácter complexo e uma dimensão considerável, foi subdividido em tarefas pelos elementos da equipa de desenvolvimento responsáveis pelo projecto. Neste documento apenas se aborda uma pequena parte do trabalho realizado, em virtude da dimensão do mesmo e de ter sido da minha responsabilidade a sua concretização. O trabalho aqui apresentado incide essencialmente na análise do sistema antigo e na análise do novo sistema.

4.1 ORGANIZAÇÃO

A FDTI foi constituída em Outubro de 1990 pelo Instituto Português da Juventude e pelo Instituto do Emprego e Formação Profissional com o objectivo de difundir conceitos técnicos e científicos, nomeadamente os que dizem respeito às Tecnologias de Informação (TI), como meio de contribuir para preparar, formar e apoiar a comunidade, em especial os jovens, no sentido de responder aos desafios da sociedade contemporânea. Trata-se de uma fundação de direito privado que se rege pelos seus estatutos e pela legislação aplicável.

A Fundação tem desempenhado um papel preponderante na formação e certificação da população portuguesa em Tecnologias de Informação e Comunicação (TIC) e no contributo para a qualificação e desenvolvimento tecnológico das organizações. Pretende, deste modo, contribuir decisivamente para a inclusão digital de toda a população, com o objectivo de colocar Portugal no grupo dos primeiros países Europeus, oferecendo serviços inovadores e abrangentes, baseados em elevados critérios de qualidade, assim como satisfazer as necessidades dos clientes internos e externos.

Anualmente abrange muitas pessoas nas suas actividades de divulgação e formação em sistemas de informação, para além de um conjunto alargado de entidades públicas e privadas.

Algumas das iniciativas passadas e presentes mais emblemáticas da FDTI são:

- Programa InforJovem, formação de utilizadores e técnicos de informática;
- Programa Diálogo, videoconferências;
- Programa Galileu, Ciência e Tecnologia para a Juventude;
- Simpósios A Geração da Informação;
- Geração Millennium, formação de jovens dos 8 aos 18;
- Competências Básicas em TIC, formação e certificação dos cidadãos em geral;

- Um sorriso com as TIC, dirigida a crianças internadas nas unidades pediátricas;
- Tek Party, evento tecnológico.

A FDTI é constituída pelos Serviços Centrais, onde se incluem o Conselho de Administração, 6 Áreas (Sistemas de Informação, Qualidade, Pessoas, Formação, Marketing e Financeira), 3 Gabinetes (Jurídico, Controlo e gestão, e Infra-estrutura) e 1 Secretariado, por 18 Delegações Distritais e uma Delegação Regional, sediadas, respectivamente, nas capitais de distrito e no Funchal (Figura 9).



Figura 9- Organograma da FDTI (Gonçalinho, 2009)

No sentido de cumprir a sua missão de disseminar as TI, contribuindo de forma significativa para a info-inclusão da população portuguesa e lusófona, em especial dos jovens e dos mais carenciados, a FDTI detém uma vasta rede de espaços que possibilitam o contacto com as tecnologias de informação. São cerca de 150 espaços com características permanentes ou móveis, distribuídos por Portugal.



Figura 10- Rede de apropriação tecnológica da FDTI (Gonçalinho, 2009)

Como se pode observar na Figura 10, a rede da FDTI, tem diversos espaços dos quais se destacam os CDTI, locais onde são divulgadas as Tecnologias de Informação, através de formações, assim como os CDTI portáteis que são bastante úteis, quando não existe um local físico onde se possam difundir os conhecimentos em TI. Não menos importantes são os Cybercentros, PBL e por fim o CDTI móvel, necessário para organizar eventos em estabelecimentos públicos, em particular em escolas.

4.2 ANÁLISE DO SISTEMA ACTUAL

4.2.1 ANÁLISE DE INVENTÁRIO

A análise de inventário, permitiu identificar as aplicações da FDTI o respectivo funcionamento assim como os principais problemas que estas apresentavam, elaborando-se o inventário das mesmas.

As aplicações existentes na Fundação são as que se seguem:

- GesCDTI - aplicação que permite gerir toda a informação relativa ao sistema de CDTI da FDTI, a localização do CDTI bem como todas as salas que o compõem;
- GesStocks - aplicação que permite a gestão de stocks das diversas unidades orgânicas;
- Agenda - aplicação que permite gerir todos os contactos de todas as entidades relacionadas com a organização;
- Equipam - aplicação desenhada para efectuar a gestão do parque informático da FDTI, permitindo a caracterização do equipamento, bem como aferir da sua localização e estado.
- GesRecibos - aplicação que emite todos os recibos da organização, inclusivé os recibos dos colaboradores;
- GesPolos - aplicação que permite a gestão de pólos, centros de alocação de custos para meios contabilísticos;
- GesTurmas - aplicação que permite a gestão de turmas por parte dos formadores e delegados distritais;

- GesPagamentos - aplicação que permite gerir os pagamentos de todos os colaboradores externos e internos da FDTI;
- GesAlunos - aplicação que permite a gestão de alunos, ou seja a inscrição dos alunos em turmas;
- GesColabora - aplicação que permite gerir os dados pessoais dos colaboradores da FDTI;
- GesCaixa - aplicação que permite gerir os fundos permanentes dos delegados distritais;
- GesInquéritos - esta aplicação permite a realização e gestão de inquéritos efectuados pela FDTI. Os inquéritos são confidenciais e anónimos para os diversos projectos da organização;
- GesFormacao - aplicação que permite gerir as formações dadas em determinado CDTI;

As bases de dados do sistema da FDTI, são as seguintes, GesInqueritos, GesAlunos, GesCaixa, GesCDTI, FDTI, GesColabora, GesPagamentos, GesStocks, GesTurmas e Gesformacao. A base de dados BaseDadosFDTI, serve 4 aplicações, a aplicação Agenda, Equipam, GesRecibos e GesPolos. Esta base de dados encontra-se também relacionada com outras bases de dados como por exemplo a BaseDadosGesCDTI, BasedadosGesStocks, BaseDadosGesTurmas e BaseDadosGesColabora.

Na Figura 11 apresenta-se a relação que existe entre as actuais aplicações e as bases de dados da FDTI. Este esquema exemplifica a complexidade do sistema existente, que integra várias bases de dados interligadas entre si, as quais servem as aplicações da FDTI.

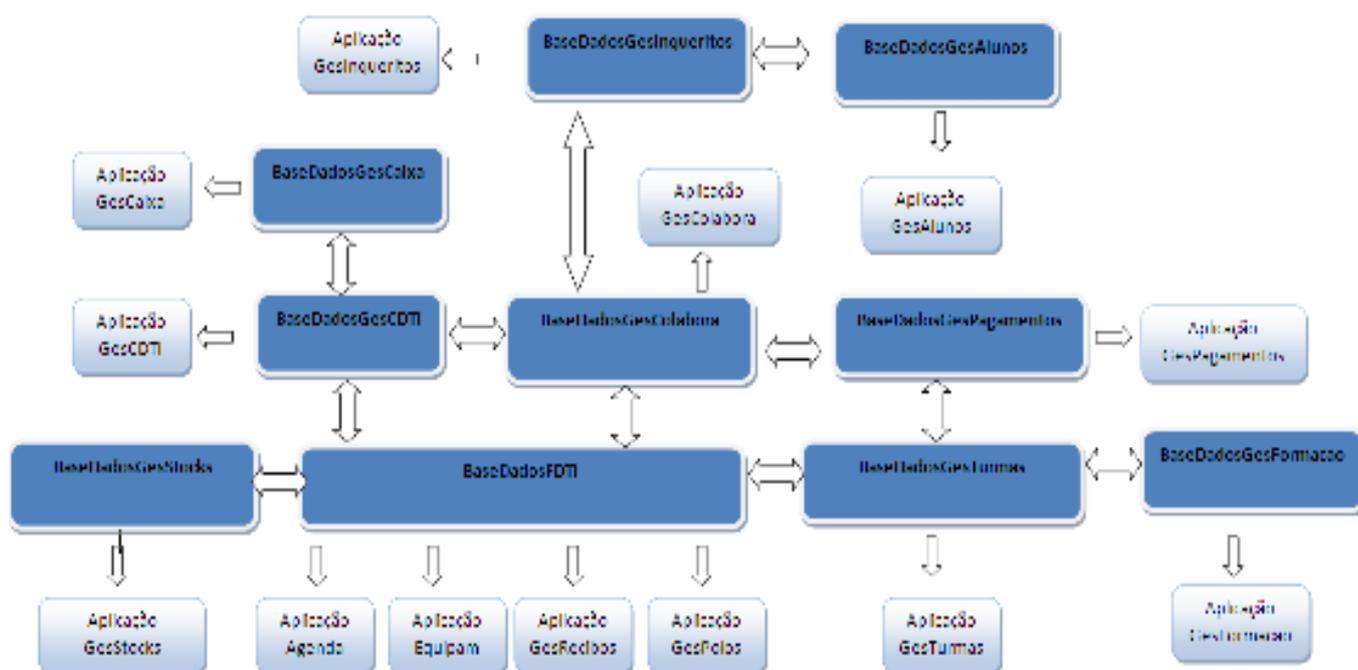


Figura 11- Sistema de informação da FDTI

Ao realizar o inventário das aplicações existentes na organização, verificou-se que estas apresentavam vários problemas, os quais se resumem ao seguinte:

- Existência de várias aplicações e várias bases de dados;
- Inexistência de Framework de serviços comuns;
- Regras de negócio espalhadas pelas várias aplicações, fazendo com que uma alteração de negócio numa aplicação, tenha implicações nas restantes;
- Falta documentação das aplicações;
- Aplicações desenvolvidas internamente, sem critérios de análise e desenvolvimento de sistemas;
- Bases de dados construídas sem critérios de modelação;
- Tabelas da base de dados sem critérios de construção, não existindo políticas correctas na definição dos identificadores ou inexistência destes;

- Inexistência de conhecimento das aplicações por todos os técnicos de desenvolvimento, dado que quem desenvolveu não elaborou documentação, nem efectuou a passagem de conhecimento das mesmas;
- Aplicações residentes em máquinas ultrapassadas;
- Inexistência de planeamento estratégico do sistema de informação;
- Versão antiga para o sistema de gestão de base de dados SQL Server 2000;
- Utilização de programação ASP.NET 1.1 e VB6;
- Queries complexas;
- Inexistência de um sistema de apoio a decisão;
- Falta de flexibilidade na inserção de novos requisitos;
- Dependência de técnicos da área de Sistemas de Informação para a realidade das tarefas básicas;
- Execução manual das operações relacionadas com os pagamentos;

Para delimitar o trabalho e atendendo às limitações temporais foram seleccionadas apenas 2 aplicações (Figura 12). A escolha destas duas aplicações (GesPolos, GesCDTI), uma vez que assumem uma importância extrema para um pleno funcionamento da organização, sendo muito utilizadas pelos delegados distritais, formadores, área financeira e técnicos dos serviços centrais, no negócio da formação em Tecnologias de Informação (TI).

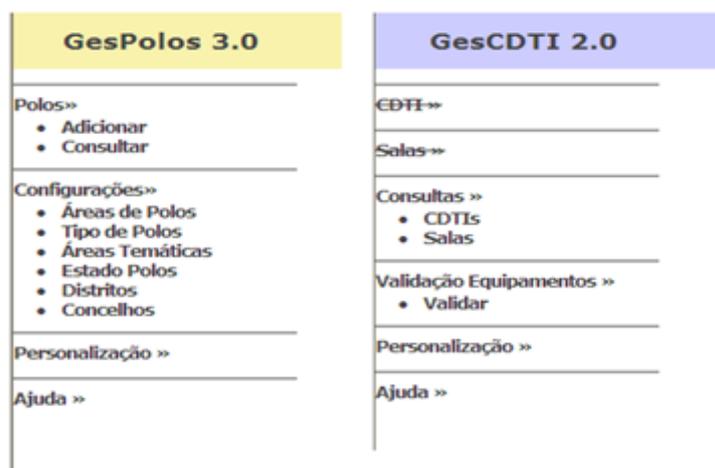


Figura 12- Aplicações GesPolos e GesCDTI

4.2.2 REESTRUTURAÇÃO DE DOCUMENTOS

A documentação das aplicações é de extrema importância, mesmo que esta tarefa seja demorada é determinante para o melhor conhecimento da funcionalidade actual do processo de negócio da organização, assim como para o futuro, visto que os processos de negócio estão em constante evolução, e assim para novos recursos humanos a sua tarefa está facilitada.

Ao analisar o estado actual das aplicações produziu-se um documento onde se descreve o contexto em que se insere o sistema antigo, assim como os objectivos a atingir, visto que até então não existia nenhum documento que explica-se o funcionamento do sistema actual.

Através do documento elaborado é mais fácil de entender o funcionamento do sistema, assim como o relacionamento entre as diferentes aplicações.

Relativamente às opções apresentadas no ponto 2.4 do capítulo 2 adoptou-se pela segunda opção apenas se está a analisar parte do sistema e assim não existe necessidade de elaborar documentação de todo o sistema. A documentação produzida teve em conta as normas de qualidade pela qual a organização se rege e pela qual é certificada, a norma ISO 9001:2000.

No caso extremo do sistema ser crítico para o negócio como um todo recomenda-se a elaboração da documentação de todo o sistema antigo, mas esta opção não se aplica à situação do sistema da FDTI.

4.2.3 ENGENHARIA REVERSA

Como foi afirmado anteriormente sem documentação das aplicações é necessário realizar uma análise detalhada, na qual a engenharia reversa assume destaque, visto que essa análise é iniciada nos sistemas existentes, analisando-os de forma a obter componentes e relacioná-los, com base no código das aplicações.

Nas próximas secções são descritas as aplicações GesPolos e GesCDTI, nas quais explica-se o funcionamento geral das aplicações, assim como os modelos de dados de cada uma delas.

Os intervenientes nas aplicações seleccionadas são o colaborador, delegado distrital, o gestor de área, o conselho de administração e o técnico.

- **Delegado Distrital** - Utilizador registado responsável por uma delegação distrital;
- **Gestor de Área** - Utilizador registado responsável por uma área organizacional;
- **Conselho de Administração** - Utilizador registado responsável pela área de conselho de administração;
- **Técnico** - Utilizador registado responsável pela área de sistemas de informação;
- **Colaborador** - Utilizador registado responsável pela área financeira;

4.3 APLICAÇÃO GESPOLOS

O conceito de pólos, refere-se à localização física de espaços da FDTI. Neste caso, a aplicação GesPolos permite a gestão de pólos, ou seja da rede de espaços da FDTI. Os pólos, podem ser do tipo que se segue:

- Armazéns (locais onde armazenam os equipamentos obsoletos, avariados entre outros);
- Centro de ciência e tecnologia;
- Centro de Divulgação de Tecnologias da Informação (CDTI);
- Centro de Novas Oportunidades (CNO);
- Centro TIC pediátrica (actividades em hospitais para crianças);
- Delegação distrital;
- Direcção;
- Feiras e exposições;
- Gabinetes;
- Loja já (centros de divulgação de informática);
- Pontos de banda larga (locais onde é possível aceder a computadores com acesso á Internet);

Na Figura 13 apresenta-se as funcionalidades existentes na aplicação GesPolos, assim como os actores que intervêm na aplicação. Esta aplicação permite gerir pólos sendo usada pelos utilizadores delegado distrital, técnico e director financeiro para realizar as seguintes operações:

- **Efectuar registo** - O utilizador colaborador tem a possibilidade de efectuar o seu registo, preenchendo o respectivo formulário, inserindo os seus dados pessoais, inserindo o seu username e a respectiva password, tendo acesso às respectivas funcionalidades.
- **Consultar pólo** - Os utilizadores delegado distrital, técnico e director financeiro podem visualizar o formulário de consulta geral de pólos. Após o devido

preenchimento do formulário é apresentada a listagem de pólos. Assim é seleccionado um pólo sendo visualizado todo o detalhe desse pólo.

- **Alterar pólo** - O utilizador delegado distrital tem a possibilidade de alterar o pólo, que seleccionou, podendo alterar apenas a designação, o tipo de pólo, a área do pólo e o estado do pólo.
- **Afectar colaboradores** - O utilizador director financeiro tem a possibilidade de afectar colaboradores, por unidade orgânica, a um determinado pólo.
- **Consultar configurações do pólo** - Os utilizadores delegado distrital, técnico e delegado distrital, nesta opção podem consultar as configurações dos pólos. É possível visualizar a listagem das áreas dos pólos, dos tipos de pólos, áreas temáticas, distritos e concelhos.
- **Alterar configurações do pólo** - O utilizador director financeiro tem a possibilidade de alterar a área temática, o tipo de pólo, as áreas dos pólos e o estado do mesmo.
- **Adicionar pólo** - O utilizador director financeiro e delegado distrital tem a possibilidade de adicionar novos pólos. Para o efeito deve ser preenchido o respectivo formulário, no qual é inserido o código do pólo, a designação, o tipo de pólo, a área do pólo e o estado do pólo.

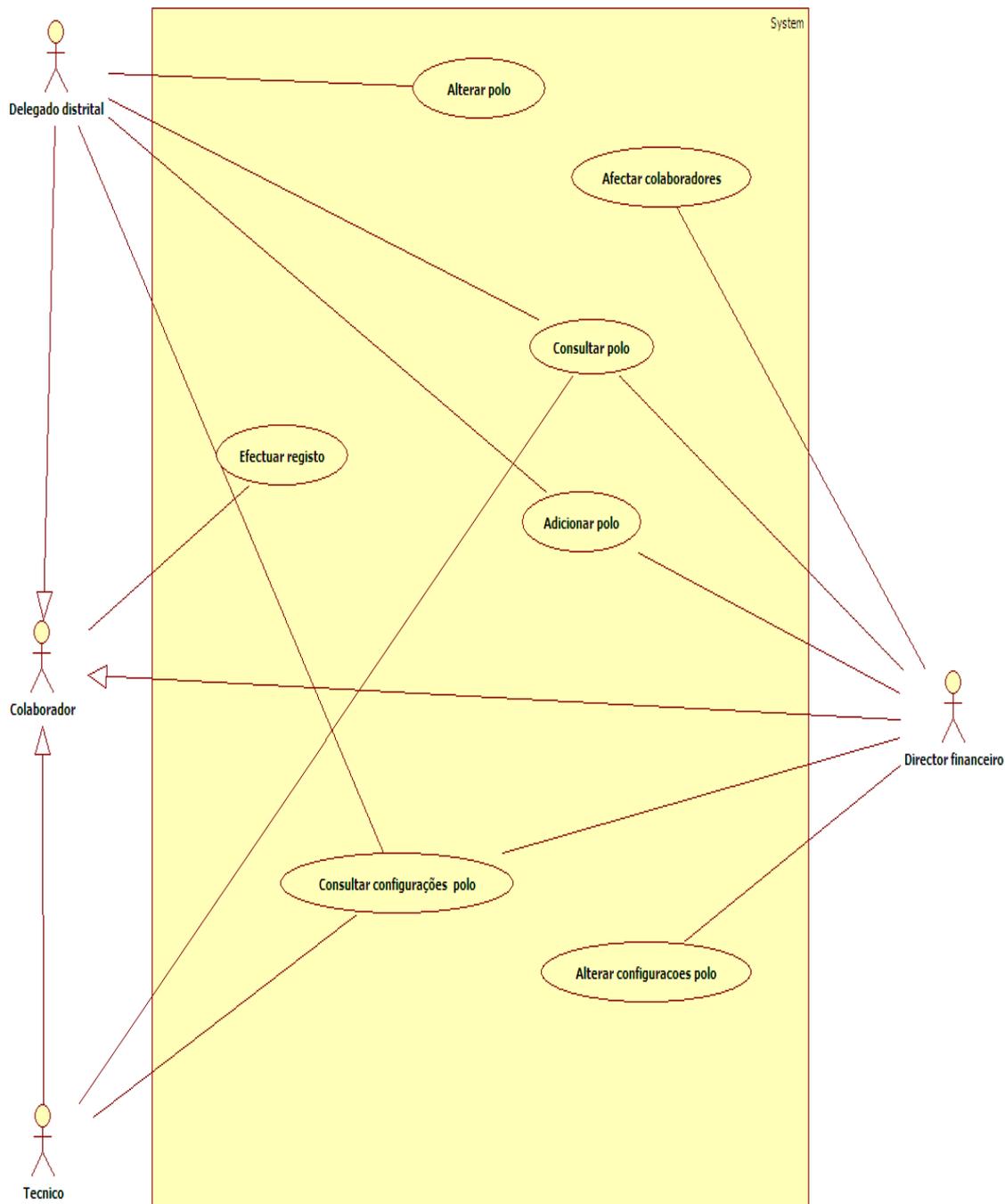


Figura 13- Diagrama casos-de-uso aplicação GesPolos

A Figura 14 representa o modelo de dados da aplicação GesPolos. Pode-se verificar que as tabelas principais são TPolo1, TAreaPolo1, TTipoPolo1, TEstadoPolo1, TCDTIGeral, TCirculoTecInf, TTipoCirculoTecInf, TDistritos e TConcelhos.

A tabela TPolos1 é uma das mais representativas, visto que toda a informação relativa aos pólos é armazenada nesta tabela. As tabelas TCDTIGeral, TCirculoTecInf, TTipoCirculoTecInf e TDelegacoes guardam informações relativas ao tipo de pólos que existe.

Consequentemente a cada pólo corresponde um determinado Distrito, logo a tabela TPolo1 está relacionada com a tabela TDistritos1 e esta por sua vez está relacionada com a tabela TConcelhos1.

A tabela TPolos1 encontra-se relacionada com a tabela TAreaPolo1, ou seja cada pólo corresponde a uma determinada área. A tabela TAreaPolo1 está relacionada com a tabela TPolosPermissoes, visto que mediante o código do colaborador assim este pode efectuar determinadas operações na área do pólo a que este tenha acesso. A TEstadoPolo guarda todas as informações relativas ao estado do pólo, ou seja se o estado do pólo é activo, inactivo, encerrado temporariamente ou encerrado definitivamente.

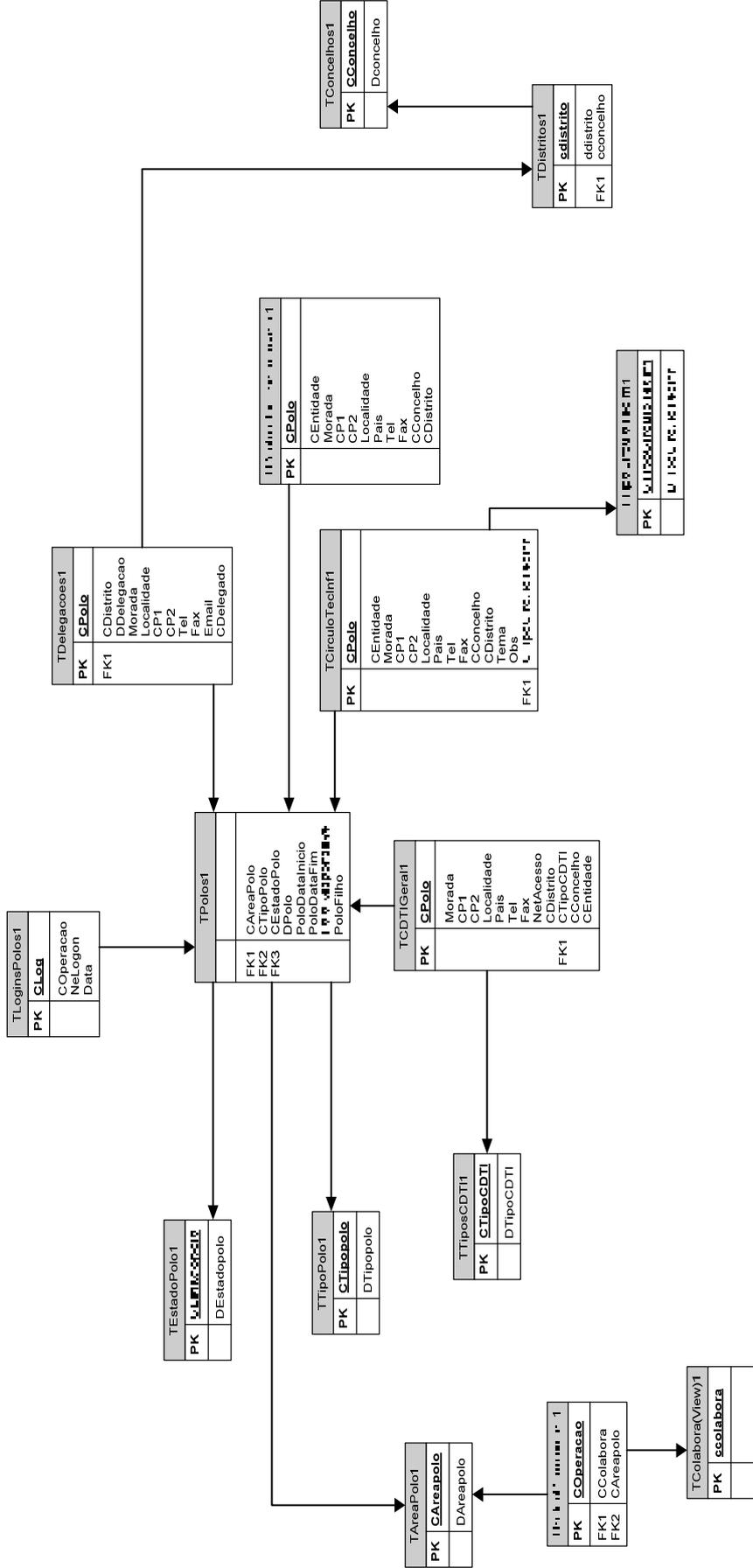


Figura 14- Modelo de dados da aplicação GesPolos

4.4 APLICAÇÃO GESCDTI

A aplicação GesCDTI, permite a gestão de CDTI, que são os locais onde existe formação de várias áreas relacionadas com as TI. Os CDTI's podem ser em local fixo, em autarquias, associações, carrinhas ou no camião da Fundação. Nesta aplicação é possível a consulta de CDTI, tendo que preencher os critérios tais como, área/ distrito, descrição do CDTI, o número de salas, tipo de CDTI, estado de CDTI, assim como visualizar a listagem de CDTI e as correspondentes características. Selecionando um CDTI é possível visualizar o seu detalhe, a sua localização, identificação e estado, acessibilidades, áreas, acesso Web, equipamento afecto ao CDTI, assim como a possibilidade de afectar equipamento às salas.

O CDTI pode ser dos tipos que seguem:

- Permanente (sitio fixo);
- Itinerante (algum tempo nas autarquias, associações, etc...);
- Portátil (carrinhas);
- Móvel (camião).

O estado do CDTI pode ser os que se seguem:

- Activo;
- Inactivo;
- Encerrado.

O diagrama de caso de uso, apresenta as funcionalidades existentes na aplicação GesCDTI, assim como os actores que intervêm na mesma. Esta aplicação permite gerir CDTI's (adicionar, alterar e consultar CDTI's) sendo usada pelos utilizadores delegado distrital, técnico e gestor de área (Figura 15).

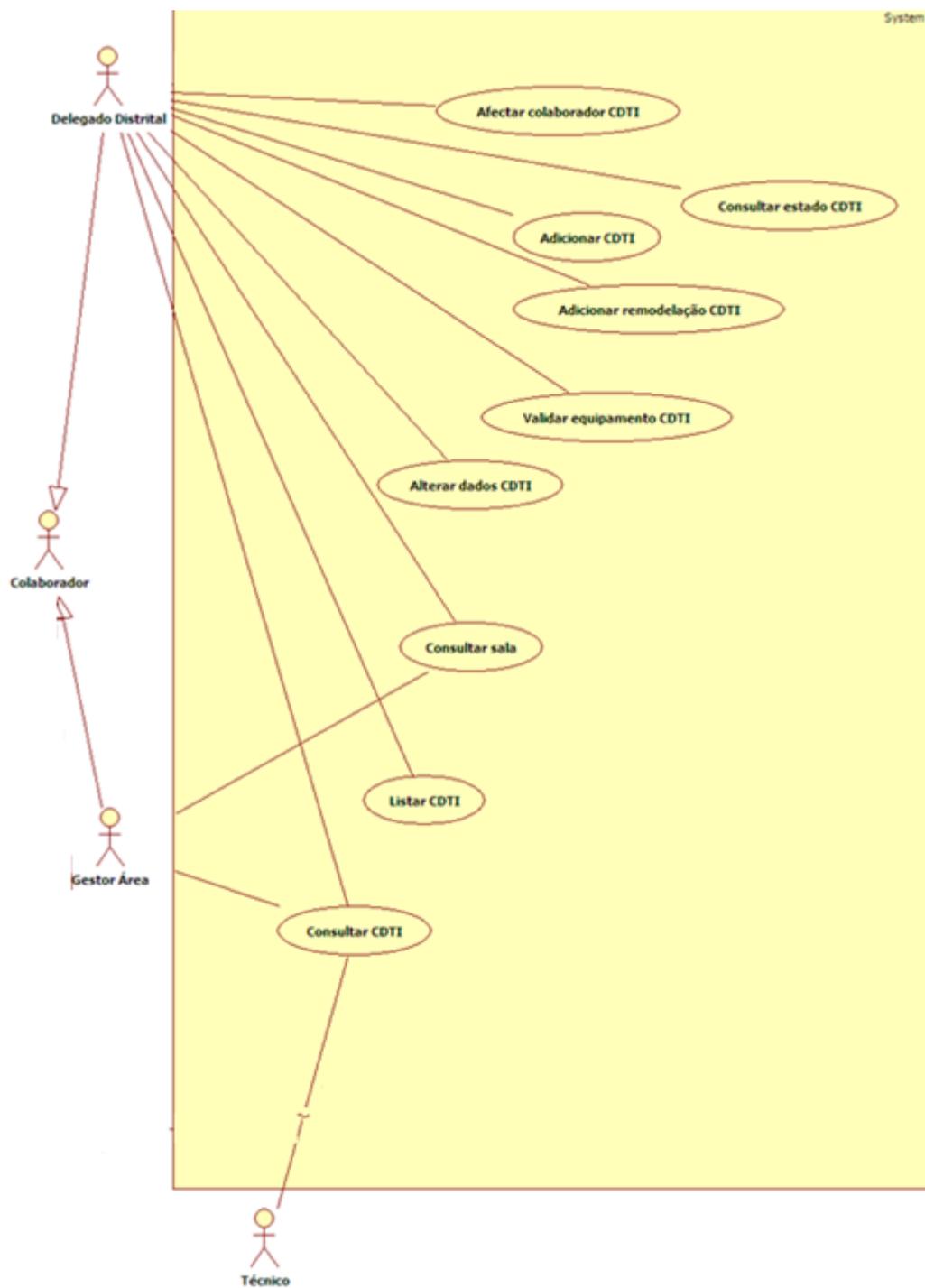


Figura 15- Diagrama casos-de-uso aplicação GesCDTI

As funcionalidades existentes na aplicação GesCDTI são as seguintes:

- **Adicionar remodelação do CDTI** - O utilizador delegado distrital tem a possibilidade de efectuar o registo das remodelações efectuadas no CDTI, na qual temos uma descrição da remodelação e a data em que ocorreu.
- **Afectar colaborador CDTI** - O utilizador delegado distrital tem a possibilidade de afectar colaboradores ao CDTI, para este poderem aceder a certas operações.
- **Consultar estado do CDTI** - O utilizador delegado distrital tem a possibilidade de consultar o estado do CDTI, onde é possível visualizar por distrito o número do CDTI e o correspondente estado dos mesmos. O estado pode ser activo, inactivo, encerrado definitivo, encerramento temporário.
- **Adicionar o CDTI** - O utilizador delegado distrital tem a possibilidade de adicionar um CDTI, devendo preencher os respectivos campos correspondentes, assim como anexar o protocolo se assim for o caso.
- **Validar equipamento CDTI** - O utilizador delegado distrital tem a possibilidade de validar o equipamento existente no CDTI. Após a visualização a listagem de equipamentos existentes no CDTI, deve ser confirmada a sua existência.
- **Consultar sala** - Os utilizadores delegado distrital e gestor de área tem a possibilidade de visualizar a listagem de salas em geral, assim como o detalhe das mesmas se assim o desejar. A cada sala pode-se afectar o equipamento que se pretenda, desde que ela pertença ao CDTI em questão.
- **Alterar dados do CDTI** - O utilizador delegado distrital tem a possibilidade de alterar os dados do CDTI, assim que seja necessário.
- **Listar CDTI** - o utilizador delegado distrital tem a possibilidade de apresentar a listagem de CDTI, após ser seleccionado o concelho ou distrito.
- **Consultar CDTI** - O utilizador delegado distrital assim como o gestor de área, têm a possibilidade de consultar um CDTI, de um modo geral, ou seja uma listagem de CDTI por distrito, assim como o detalhe do CDTI.

Na Figura 16 observa-se que a tabela TCDTI, guarda todos os dados relativos ao CDTI. Por exemplo, as informações sobre a localização, descrição dos acessos ao CDTI. Esta tabela está relacionada com a tabela TCDTISalasEquipamento, a qual armazena as informações dos equipamentos alocados ao respectivo CDTI.

A tabela TCDTISala encontra-se relacionada com a tabela TCDTI, visto que um CDTI tem uma ou várias salas de formação. Esta tabela armazena todos os dados que dizem respeito á sala.

Cada CDTI pode necessitar de remodelação, estando a informação relativa às remodelações na tabela TRemodelacoes. Uma vez que as remodelações podem ser de vários tipos a tabela TRemodecoes está relacionada com a TTipoRemodelacoes.

Na tabela THistorico são armazenadas todas as informações relativas às operações efectuadas nos CDTI's, inclusivé as datas e os autores das operações.

Um CDTI pode ainda ter valências, ou seja informações específicas do CDTI. A tabela TCDTIValencias, guarda os dados relativos às valências de um determinado CDTI. A tabela TEstadoSala armazena o estado da sala do CDTI.

CAPÍTULO 5

5 SISTEMA PROPOSTO PARA A FDTI

5.1 ENGENHARIA PROGRESSIVA

O esquema da Figura 17 apresenta a solução proposta para uma plataforma única, subdividida por módulos, na qual é possível agrupar e gerir os serviços comuns às várias aplicações da FDTI.

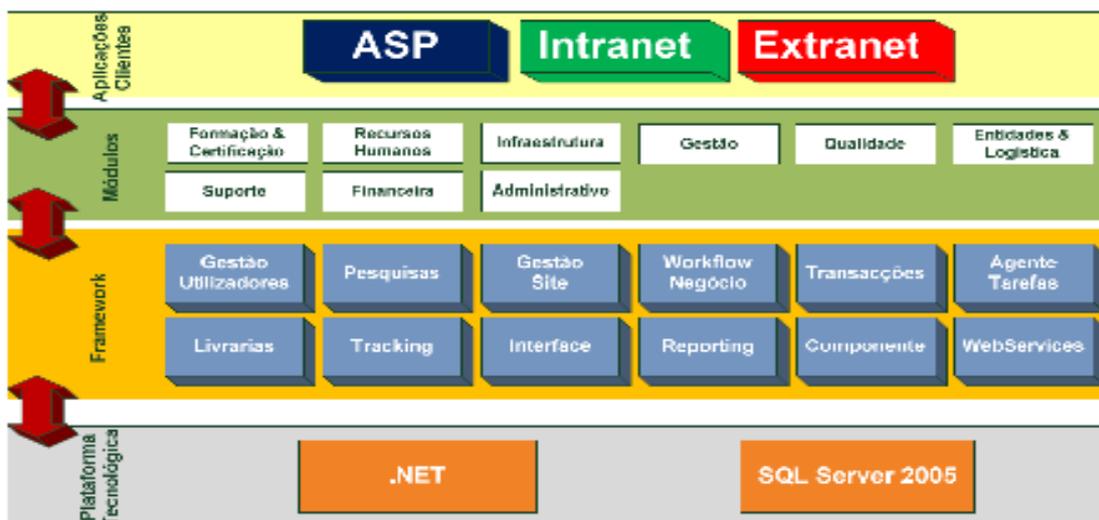


Figura 17 – Plataforma única

A nova solução é baseada no conceito de uma Framework de serviços comuns, na qual cada um dos módulos acede aos serviços que necessitar.

Os serviços disponíveis pela Framework permitem identificar o módulo de origem do pedido de acesso ao sistema da Framework, respondendo a mesma ao respectivo pedido.

A Framework será ainda responsável por deter a gestão de perfis, passwords, utilizadores e toda a gestão de regras de negócio dos módulos. Os módulos do sistema de informação são controlados pela Framework.

Com a utilização da Framework é possível uniformizar as regras de negócio, implementar um sistema pedido/resposta, permitindo uma melhor qualidade de serviço, adicionar novas funcionalidades permitindo assim a manutenção futura mais fácil.

A plataforma tecnológica usada na Framework é o Microsoft .Net, o qual visa o desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código é gerado pelo .Net, podendo ser executado em qualquer dispositivo que possua a Framework .Net que interage com um conjunto de bibliotecas, possibilitando a reutilização de componentes, em particular classes. Quanto ao Microsoft SQL Server 2005, é um sistema de base de dados robusto usado por várias áreas de actuação.

De seguida descrevem-se os novos módulos propostos na plataforma:

- **Módulo de Formação e Certificação** - Este módulo da aplicação tem como objectivo gerir todo o processo de Formação e Certificação, passando pela gestão dos Projectos de Formação e respectivos Módulos, Gestão de Alunos (internos e externos), Turmas e Certificados. Este módulo deverá assentar sobre uma base única de tabelas, que possam ser utilizada pelos restantes módulos da aplicação, bem como, utilizar as tabelas destes. Deve também permitir a existência de vários utilizadores, com vários níveis de permissão. A aplicação deverá permitir, ao nível da administração da mesma, criar perfis e associar a estes as funcionalidades existentes na aplicação, às quais se pretende que aquele perfil tenha acesso. As permissões de acesso as várias funcionalidades ao nível da Formação e Certificação será posteriormente definida pelo Programa Formação.
- **Módulo de Recursos Humanos** - Este módulo vai incluir funcionalidades das aplicações GesColabora, GesPagamentos e bolsa de formadores.

- **Módulo de Infra-estrutura** - O módulo Infra-Estrutura, pretende englobar as aplicações GesPolos e GesCDTI.
- **Módulo Financeiro** - Este módulo é responsável pela área financeira, contendo funcionalidades relacionadas com a aplicação GesTransfer e GesCaixa.
- **Módulo Qualidade** - Este módulo é responsável pela área da qualidade, contendo funcionalidades relacionadas com informação estatística do GesInqueritos e pelos indicadores de gestão.
- **Módulos Entidades e Logística** - Este módulo engloba a aplicação Agenda, na qual temos as informações relativas às entidades e contactos da FDTI.
- **Módulo Administrativo** - Este módulo é responsável pela parte relativa a aplicação GesRegulamentacao, mantendo as mesmas funcionalidades.
- **Módulo Gestão** - Este módulo é responsável pela gestão da Framework, tendo as funcionalidades de criação, actualização ou eliminação de utilizadores, gestão de perfis de utilizador e criação de módulos.
- **Módulo Suporte** - Este módulo é responsável pelo suporte técnico. Este módulo poderá ser substituído por aplicação externa.

Em seguida será analisado em detalhe o módulo Infra-Estrutura que engloba as aplicações GesPolos e GesCDTI.

Seleccionou-se abordar este módulo devido ao facto ser uma das áreas que mais influencia o negócio da formação, visto que os outros módulos estão relacionados com este. O módulo Infra-estrutura é responsável pela rede de infra-estruturas da FDTI, ou seja pelos CDTI, nos quais ocorrem as formações em TI. As formações ocorrem num determinado CDTI que é um tipo de pólo. No final de cada formação é necessário o preenchimento dos inquéritos relativos á formação dada pelo formador.

Nas secções seguintes descreve-se a análise de requisitos efectuada no módulo infra-estrutura, a modelação do sistema, em particular, os casos-de-uso, diagramas de classes, diagramas sequência e diagramas de actividades realizados. Por fim a reestruturação de dados realizada no módulo seleccionado.

5.2 ANÁLISE DE REQUISITOS

Os intervenientes na utilização do módulo Infra-estrutura, são o colaborador, delegado distrital, o gestor de área, o conselho de administração e o técnico.

- **Delegado Distrital** - Utilizador registado responsável por uma delegação distrital;
- **Gestor de Área** - Utilizador registado responsável por uma área organizacional;
- **Conselho de Administração** - Utilizador registado responsável pela área de conselho de administração;
- **Técnico** - Utilizador registado responsável pela área de sistemas de informação;
- **Colaborador** - Utilizador registado responsável pela área financeira;

Na Figura 18 são apresentados os requisitos funcionais que são suportados pelo novo sistema.

- **Efectuar registo** - O utilizador colaborador tem a possibilidade de efectuar o seu registo, preenchendo o respectivo formulário com os seus dados pessoais, no qual insere o seu username e a respectiva palavra-chave tendo acesso as respectivas funcionalidades.
- **Efectuar login** - O utilizador colaborador após ter efectuado o registo, inicia a sua sessão, na qual tem que inserir o respectivo username e palavra-chave para ter acesso ao sistema.
- **Efectuar logout** - O utilizador colaborador após ter iniciado a sua sessão tem a possibilidade de finalizar a mesma, seleccionando a opção efectuar logout.
- **Consultar Pólo** - Os utilizadores delegado distrital e técnico podem visualizar o formulário de consulta geral de pólos. Após o devido preenchimento do formulário é apresentado a listagem de pólos. Assim é seleccionado um pólo sendo visualizado todo o detalhe desse pólo.

- **Alterar pólo** - O utilizador delegado distrital tem a possibilidade de alterar o pólo, que seleccionou, podendo alterar apenas a designação, o tipo de pólo, a área do pólo e o estado do pólo.
- **Afectar colaborador** - O utilizador delegado distrital tem a possibilidade de afectar um colaborador ao pólo, para este poder efectuar aceder a certas operações.
- **Adicionar pólo** - O utilizador delegado distrital tem a possibilidade de adicionar um pólo, devendo para tal preencher um formulário, no qual é inserido o código do pólo, a designação, o tipo de pólo, a área do pólo e o estado do pólo. Caso seja seleccionado o tipo pólo “CDTI”, então é possível Adicionar o tipo de pólo CDTI, preenchendo-se o formulário do CDTI, assim como se pode anexar o protocolo se assim for o caso.
- **Adicionar remodelação do CDTI** - O utilizador delegado distrital tem a possibilidade de efectuar o registo das remodelações efectuadas no CDTI, a qual contém uma descrição da remodelação e a data em que foi realizada.
- **Candidatar abertura de CDTI** - É necessário preencher uma ficha de candidatura no qual temos os critérios de aceitação das instalações, no qual temos os critérios de cumprimento obrigatórios e os opcionais. Efectivamente, caso sejam reunidas as condições de abertura do CDTI, após essa confirmação dos critérios de abertura, é necessário preencher outra ficha de “Candidatura abertura de CDTI”. Neste caso a entidade candidata a abertura de CDTI preenche os respectivos dados do CDTI, cabendo ao concelho de administração a tomada de decisão. O delegado distrital preenche a ficha de candidatura, após o preenchimento é enviado um e-mail ao Gabinete de Infra-estruturas. O responsável pelo Gabinete analisa a proposta de abertura, validando a mesma. Após esta validação é enviado um e-mail ao CA e estes aprovam a abertura do mesmo.
- **Consultar estado do CDTI** - O utilizador delegado distrital tem a possibilidade de consultar o estado do CDTI, onde é possível visualizar por distrito o número de CDTI e o correspondente estado dos mesmos. O estado pode ser activo, inactivo, encerrado definitivo, encerramento temporário.

- **Validar equipamento CDTI** - O utilizador delegado distrital tem a possibilidade de validar o equipamento existente no CDTI. Deste modo tem-se acesso à listagem de equipamentos existentes no mesmo, e por fim é confirmada a sua existência.
- **Consultar sala** - O utilizador delegado distrital e o gestor de área tem a possibilidade de visualizar a listagem de salas em geral, assim como o detalhe das mesmas se assim o desejar. Assim como pode afectar equipamento à sala que pretender desde que esta pertença ao CDTI em questão.
- **Alterar dados do CDTI** - O utilizador delegado distrital tem a possibilidade de alterar os dados do CDTI, assim que seja necessário.
- **Consultar CDTI** - O utilizador delegado distrital assim como o gestor de área tem a possibilidade de consultar o CDTI, de um modo geral, ou seja uma listagem de CDTI por distrito, assim como o detalhe. É possível afectar equipamento ao CDTI, assim como afectar salas ao mesmo.
- **Listar CDTI** - o utilizador delegado distrital tem a possibilidade de listar os CDTI's, após a selecção do concelho ou distrito.
- **Analisar candidatura de abertura de CDTI** - O utilizador conselho de administração ao receber um e-mail com a proposta de abertura de CDTI, analisa-a e se entender aprova a sua abertura. Neste caso é emitido mesma, emitindo um despacho de abertura do CDTI que é enviado ao respectivo Delegado distrital que solicitou a abertura do mesmo.
- **Candidatar encerramento CDTI** - Efectivamente pode ser necessário o encerramento de CDTI. Nesse caso é necessário a FDTI, preencher a proposta de encerramento do CDTI, o qual tem o nome do CDTI, localidade, o tipo de encerramento, a razão do encerramento, a afluência dos últimos anos, destino do equipamento, a reafecção dos formandos. A entidade que propõe o encerramento, em geral é a FDTI, preenche o respectivo formulário, no qual refere o “parecer acerca do encerramento”, a avaliação do desempenho da FDTI, aspectos positivos da FDTI, os aspectos negativos da FDTI, o nome do inquirido, cargo, data e assinatura do mesmo.

- **Aprovar candidatura abertura CDTI** - É necessário entregar ao CA o questionário referindo as condições do espaço onde que vai ser instalado o CDTI. O pedido é analisado pelo CA, podendo ser aprovado a criação do CDTI. Só após esta validação do CA é emitido o despacho no qual se aprova a abertura do CDTI. Nesse momento o delegado ou o gestor do gabinete das infra-estruturas pode adicionar o CDTI na aplicação.

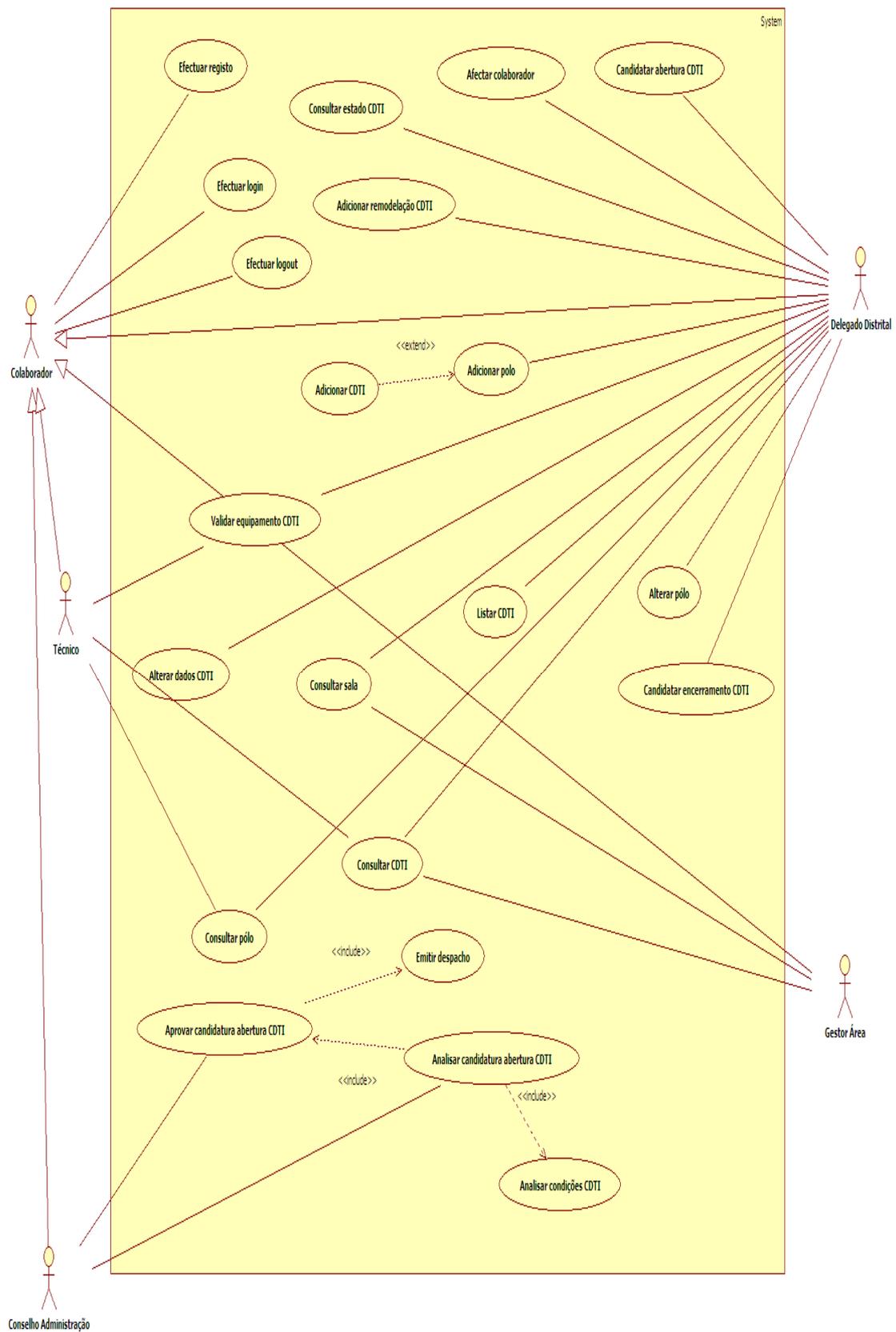


Figura 18- Diagrama casos-de-uso módulo infra-estrutura

5.3 DIAGRAMAS DO SISTEMA

Os diagramas utilizados no novo sistema são os diagramas de casos-de-uso, diagramas de classes, diagramas de sequência e diagramas de actividades recorrendo à técnica de UML.

Seleccionou-se apenas estes quatro modelos de diagramas, de entre os disponíveis, visto que era necessário saber-se quais os requisitos funcionais do sistema. Deste modo o diagrama de casos-de-uso explica o funcionamento dos requisitos funcionais do sistema assim como os seus intervenientes, este diagrama foi utilizado na secção anterior quando realizada a análise de requisitos.

Através do modelo de dados determinaram-se as classes importantes assim como os diagramas de sequência e actividades. Algumas das funcionalidades do sistema evidenciaram muitos pormenores, sendo muito importantes os diagramas de sequência e de actividades para compreender seu o correcto funcionamento.

5.3.1 DIAGRAMAS DE CLASSES

Na Figura 19 apresenta-se o modelo de classes do sistema do módulo infra-estrutura, no qual são apresentadas as classes principais, tais como, Polo, CDTI, Sala, FichaAbertura, Valencias, TipoSala, EstadoSala, TipoPolo.

Como se pode observar na figura tem-se uma generalização da classe Delegacoes e da classe CDTI, ou seja ambas as classes partilham os mesmos atributos gerais e funcionalidades da classe Polo, mas apresentam algumas particularidades em termos de atributos e métodos. Os CDTI podem ter uma ou várias salas de formação, por sua vez a sala tem um tipo de sala.

A sala tem um estado de sala, dentro da sala temos equipamentos, que por sua vez estão relacionados com um CDTI.

A grande alteração nestas classes verificou-se nos atributos da classe Polo, visto que não tinha os atributos morada, código postal, localidade, país e telefone etc. Estes atributos estavam armazenados na classe CDTI, que assim eram replicados na classe Delegacoes. Foram também criadas mais duas classes a FichaAbertura e FichaEncerramento, para armazenar todos os dados relativos às fichas dos CDTI, algo que não era realizado anteriormente.

5.3.2 DIAGRAMAS DE SEQUÊNCIA

Nesta secção são apresentados dois diagramas de sequência (Figura 20 e 21), relativos às seguintes funcionalidades:

- Inserir Pólo;
- Candidatar abertura de CDTI.

Seleccionaram-se estes 2 diagramas uma vez que é necessário conhecer o funcionamento pormenorizado destas funcionalidades, assim como o actor responsável pelo evento, uma vez que são as mais solicitadas na aplicação.

Neste primeiro diagrama (Figura 20) tem-se a representação da funcionalidade Inserir Polo. O actor delegado distrital selecciona a opção “Inserir polo” e o sistema interno, verifica as permissões de acesso deste utilizador, inserindo o respectivo código do pólo. Os dados são preenchidos no respectivo formulário. Por fim será apresentada no ecrã uma mensagem confirmando a inserção do novo pólo.

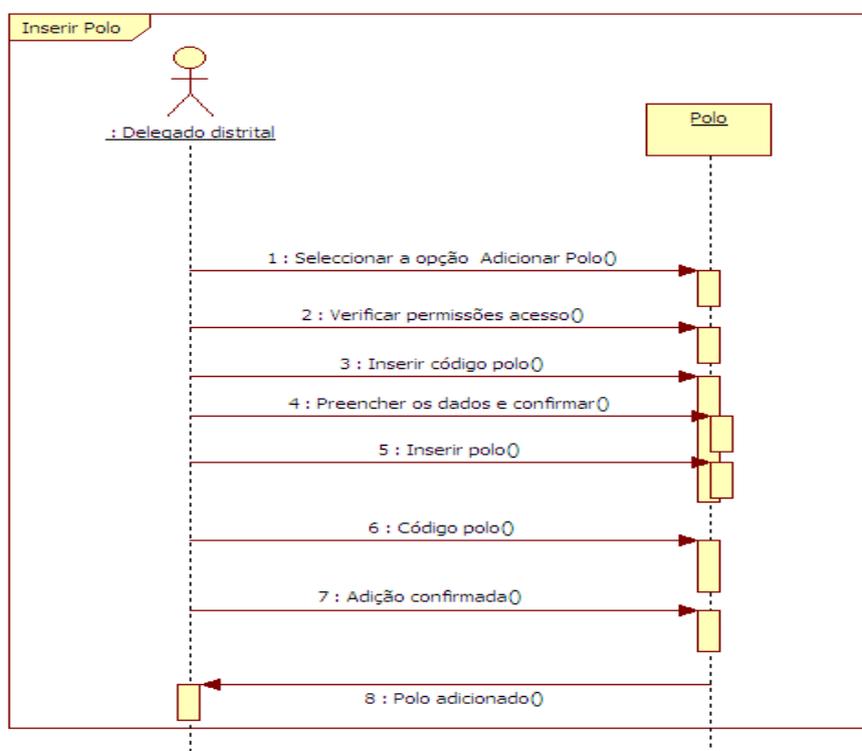


Figura 20- Diagrama de sequência - inserir pólo

Na Figura 21 o utilizador delegado distrital selecciona a opção Candidatar CDTI, verifica os critérios de aceitação das instalações do CDTI, sendo estes confirmados. Após a verificação dos critérios de abertura, realiza-se o preenchimento da candidatura do CDTI. Sendo confirmados os dados da proposta após o preenchimento do formulário. O actor conselho de administração analisa a abertura de CDTI, elabora um parecer favorável à abertura do CDTI, selecciona as valências do mesmo, aprova a candidatura de abertura do CDTI, emite um despacho, sendo finalizada a candidatura de abertura de CDTI.

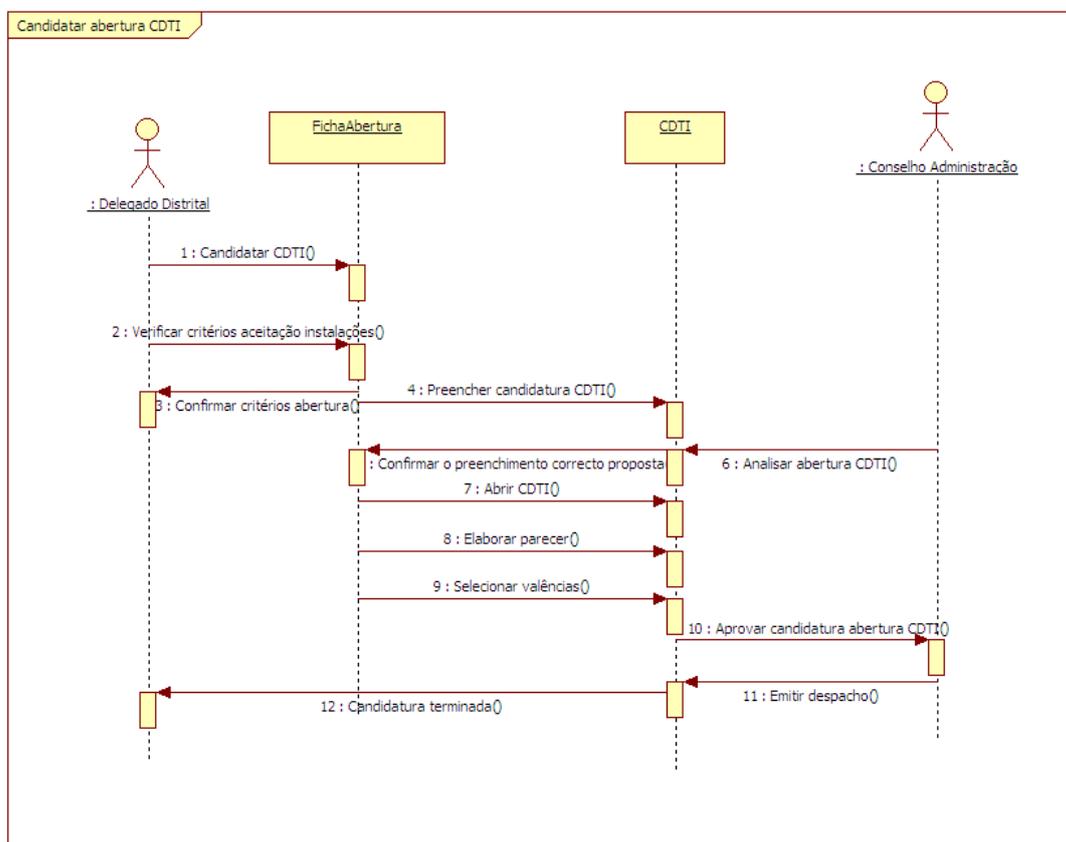


Figura 21 - Diagrama de Sequência - candidatar abertura de CDTI

5.3.3 DIAGRAMAS DE ACTIVIDADES

Os diagramas de actividades representados (Figura 22 e Figura 23), pretendem explicar o funcionamento das funcionalidades Inserir pólo e Candidatar Abertura do CDTI, anteriormente referidos no diagrama de sequência.

Através da Figura 22, o colaborador efectua o registo na aplicação, iniciando a sua sessão. Caso tenha o perfil do delegado distrital existe a possibilidade de aceder a opção inserir pólo.

Ao escolher a opção “Inserir pólo”, deve-se indicar o tipo de pólo que se pretende inserir e depois preencher devidamente todo o formulário do Pólo. No final, logo depois da confirmação do correcto preenchimento do formulário, os dados são guardados sendo o pólo inserido imediatamente. Por fim é finalizada a operação de adição de pólo e caso o utilizador queira é feito o logout do sistema.

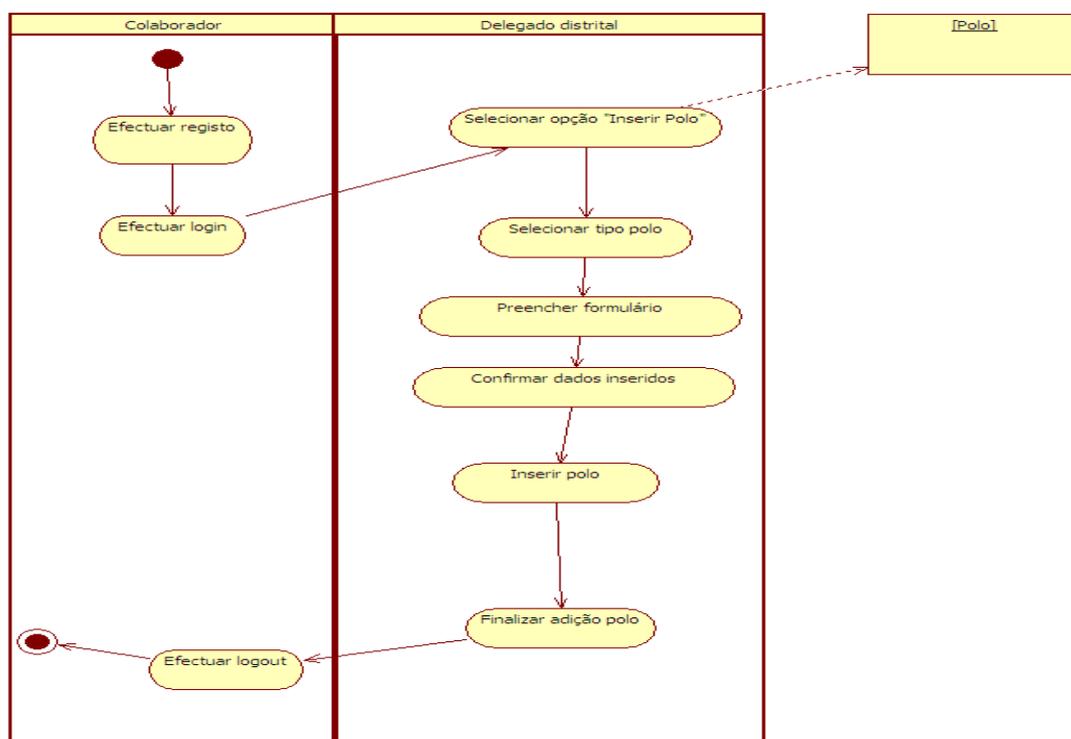


Figura 22- Diagrama de actividades - inserir pólo

Na Figura 23 é apresentado o diagrama de actividades relativo à funcionalidade “Candidatar abertura do CDTI”. O delegado distrital tem a possibilidade de candidatar abertura do CDTI, verifica os critérios de aceitação das instalações, estes critérios podem ser obrigatórios ou opcionais. Após a confirmação dos critérios de abertura, este preenche a candidatura de abertura do CDTI, confirma o preenchimento correcto do formulário. Paralelamente o conselho de administração analisa a abertura do CDTI, aprova a candidatura decidindo a sua abertura. O delegado distrital elabora um parecer, selecciona as valências do CDTI, enquanto que o conselho de administração emite um despacho, finalizando a candidatura de abertura do CDTI.

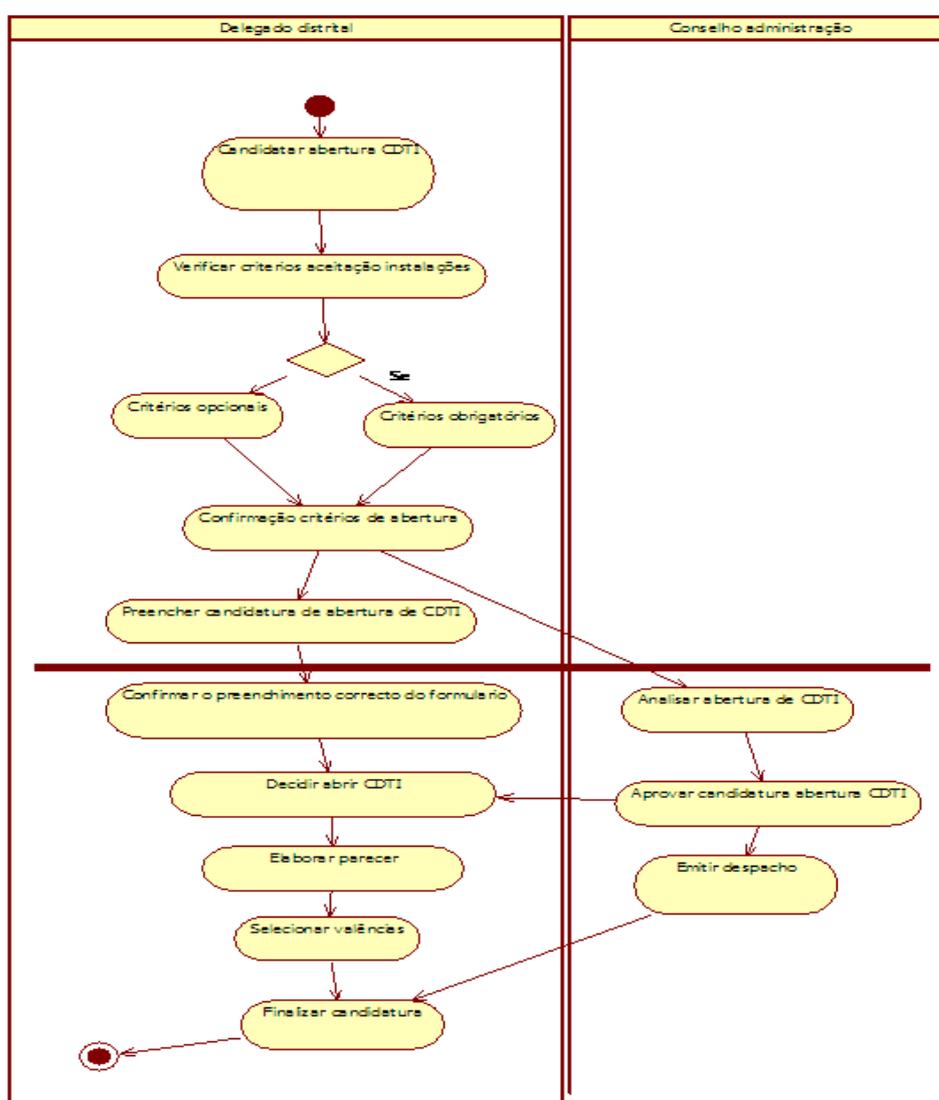


Figura 23- Diagrama de Actividades - candidatar abertura de CDTI

5.4 REESTRUTURAÇÃO DE DADOS

Na Figura 24 apresenta-se o modelo de dados do módulo Infra-estrutura, que representa a integração entre as tabelas da base de dados do GesPolos e GesCDTI. Esta reestruturação baseou-se nos critérios de modelação, no qual as tabelas já tem identificadores assim como estão todas relacionadas, algo que não se verificava anteriormente.

Foram adicionadas 2 novas tabelas, a tabela TFichaAberturaCDTI e TFichaEncerramentoCDTI, visto que os formulários eram preenchidos em formato papel. A tabela TCDTI esta relacionada com a tabela TPolo, TCDTISala, TFichaAberturaCDTI, TFichaEncerramentoCDTI, TTipoAcessoWeb, TCDTIValencias, TRemodelacoes, TTipoCDTI e TEstadoCDTI.

Como se pode verificar e comparando com o modelo de dados da aplicação GesPolos, eliminaram-se 4 tabelas, uma vez que não eram usadas, TCirculoTecInf, TpostoInformacaoJuvenil e TTipoCirculoTecInf, assim como a tabela TTipoRemodelacao da aplicação GesCDTI.

A tabela TCDTI, permite guardar todas os dados relativos ao pólo do tipo CDTI, por exemplo, as informações sobre a localização, descrição dos acessos ao CDTI. A tabela TCDTISala encontra-se relacionada com a tabela TCDTI, visto que um CDTI tem uma ou várias salas de formação, esta tabela armazena todos os dados que dizem respeito á sala. Cada CDTI pode necessitar de remodelação, a informação relativa às remodelações dos CDTI, encontra-se na tabela TRemodelacoes, deixando de existir TTpoRemodelacao.

Na tabela THistorico são armazenadas todas as informações relativas às operações efectuadas nos CDTI's, inclusive as datas e os autores das operações.

Um CDTI pode ainda ter valências, ou seja informações específicas do CDTI. A tabela TCDTIValencias, guarda os dados relativos às valências de um determinado CDTI. A tabela TEstadoSala armazena o estado da sala do CDTI.

6 CONCLUSÃO E TRABALHO FUTURO

Dados os montantes significativos geralmente envolvidos nos investimentos em engenharia de software, torna-se particularmente relevante identificar o que move as organizações no que concerne à reengenharia de software. As empresas recorrem actualmente a múltiplos softwares para suportar e potenciar o seu negócio assim como as razões que os levam a investir neles. O custo/benefício da reengenharia deve ser determinada quantitativamente isto é, o custo associado com o suporte e a manutenção actuais de uma aplicação existente é comparado com os custos projectados da reengenharia e da redução resultante em custos de manutenção. Em quase todos os casos nos quais um programa tem uma vida longa, e no momento apresentam uma baixa manutenção a reengenharia representa uma estratégia de negócio eficiente em termos de custo.

No caso da FDTI, foi mais fácil compreender o software actual recorrendo ao processo de reengenharia de software, e assim conseguir compreender e implementar um novo software, baseado no sistema antigo.

O sistema futuro apresenta um grau de complexidade elevado, deste modo, descreveu-se apenas um módulo da solução encontrada. Recorreu-se a técnicas de modelação tais como casos de uso, diagramas de classes, diagramas de sequência e diagrama de actividades para compreender o seu funcionamento.

Ao longo da realização deste projecto foram encontradas inúmeras dificuldades as quais foram ultrapassadas com sucesso. As dificuldades sentidas deveram-se à falta de documentação que explicasse o funcionamento das aplicações e das respectivas bases de dados, assim como as regras de negócio estavam embebidas nas stored procedures.

Conseguiu-se atingir os objectivos pretendidos com a realização deste trabalho, uma vez que, analisou-se uma plataforma de serviços comuns, uniformizou-se as regras de negócio, adicionou-se novas funcionalidades e reestruturou-se os dados com base em critérios de modelação.

Neste momento a fase de análise e especificação de requisitos esta concluída, tendo sido estes validados pelos responsáveis das áreas intervenientes no processo. Todo o planeamento do funcionamento da plataforma única foi analisado e elaborado, mas consequentemente poderá sofrer algumas alterações consoante estas sejam necessárias.

Como se pode verificar abordou-se apenas uma parte da plataforma única, os módulos, nos quais apenas destaquei o módulo infra-estrutura, por limitações temporais não foi possível descrever em pormenor todas as outras partes do funcionamento da plataforma.

Futuramente será realizada toda a migração dos dados da plataforma antiga para a nova plataforma, reorganização das estruturas dos dados, modificação das mesmas caso seja necessário, assim como o desenvolvimento da solução, após a finalização do desenvolvimento, serão realizados testes funcionais da aplicação, será realizada formação para os utilizadores do sistema, sendo deste modo assegurado a manutenção do novo sistema de informação da FDTI.

Pode-se concluir que adoptar o processo de RS, neste caso de estudo, foi a opção mais adequada, perante a complexidade do sistema antigo da FDTI. Caso se adopta-se pela elaboração de um sistema novo de raiz era necessário mais recursos humanos o que imputava mais custos, inviabilizando a implementação do mesmo. A razão principal da adopção deste processo foi mesmo o facto das regras de negócio estarem embutidas no sistema antigo, sendo mais fácil analisá-lo e assim melhorar os aspectos negativos que este apresentava, assim os riscos eram menores do que desenvolver um sistema novo.

BIBLIOGRAFIA

Alcântara, R. (2002). Dissertação de Mestrado: O desenvolvimento de sistemas de informação utilizando software livre. Florianópolis, Universidade Federal de Santa Catarina.

Barrere, T., & Prado, A. (2000). "Bonafé V.C Reengenharia de Software Orientada a Componentes Distribuidos". *XV Simpósio Brasileiro de Engenharia de Software* (pp. 30-34). Editora Campus.

Bergey, J. (2000). *"Why reengineering projects fails"*. Pittsburgh: Software Edition.

Booch, G. (2002). *Software Architecture and the UML*.

Booch, G. (1998). *The Unified Modeling Language User Guide*. New York: Addison-Wesley .

Booch, G., Rumbaugh, & Jacobson, I. (2000). *The Unified Modeling Language UserGuide*. New York: Addison Wesley.

Chikofsky, E. J., & Cross, J. (1990). "Reverse Engineering and design recovery- A Taxonomy". *IEEE Software* , 1,7.

Davenport, T. (2003). "Process Innovation, Reengineering Workthrought Information Tecnology". *Harvard Business Shool Press* .

Fayad, M., & Jonhson, R. (2000). *Domain specific application framework: Frameworks experience industry* (1ª ed.). Jonh Wiley e Sons.

Filho, P. (2003). *"Engenharia de software: Fundamentos, Métodos e Padrões"* (2ª ed.). Rio de Janeiro: LTC.

Flynn, K. (1994). Critical success factors for a successful business reengineering project. *Conference Proceedings*, (pp. 50-60). Boston.

Fontanette, J., Garcia, V., Perez, A., Rossanaro, B., Adriano, A., Prado, et al. (2003). "Estratégia de Reengenharia de software baseado em componentes". *5ª Sessão Técnica*. Havai: IEEE CS/Press.

Fukuda, A. P. (2000). *Refinamento automático de sistemas orientados a objectos distribuidos*. Brasil: UFSCar.

Garcia, C., Lucrecio, D., Prado, F., & Almeida, E. (2004). "Towards an effective Approach for Reverse Engineering". *Proceeding of the 11 Th Working Conference on Reverse Engineering. I*, pp. 290-299. Amsterdam: IEEE/CS Press.

Garcia, C., Lucrecio, D., Prado, F., & Almeida, E. (2005). "Towards on Approach for Aspect- Oriented Software Reengineering". *Proceedings of 7 Th International Conference on Enterprise Information Systems. 1*, pp. 274-279. Miami: IEEE/ CS Press.

Gimenes, I. (1994). "Um ambiente de apoio ao desenvolvimento de software crítico XV". *II Congresso Brasileiro de Engenharia de software* (pp. 239-354). Caxambu: NovaTec.

Gonçalinho, E. (12 de Fevereiro de 2009). Manual da qualidade. *Sistema de Gestão da Qualidade da FDTI*, pp. 4-8.

Guedes, T. (2007). *"UML 2-Guia Prático"* (2ª ed.). Brasil: NovaTec.

Guedes, T. (2008). *UML- Uma Abordagem Prática* (3ª Edição ed.). Brasil: NovaTec.

Hammer, M. (1990). "Reengineer work: Don't automate, obliterate". *Harvard Business Review*, 104-112.

Hammer, M., & Champy, J. (1994). *Reengineering the Corporation : A Manifesto for Business Revolution*. New York: North River Press Inc.

Institute, S. (2002). "CMMI for Systems Engineering/Software Engineering". Pittsburg: Carnegie Mellon University.

Jacobson, I., Ericsson, M., & Jacobson, A. (1994). *"The Object Advantage: Business Process Reengineering with Object Technology"*. New York: ACM Press.

Joia, L. A. (2004). *Geração de modelos heurísticos a partir de estudos de múltiplos casos* (2ª Edição ed.). Rio de Janeiro: FGV.

Joia, L. A. (1994). *Reengenharia e Tecnologias de informação*. Rio de Janeiro: Pioneira.

Landon, K., & Landon, J. (2003). *"Essentials of Management Information Systems: Organization and Technology"* (8ª ed.). New York: MacMillan.

Larman, C. (2002). *Utilizando UML e Padrões- Uma Introdução à análise e a projectos orientados a objectos*. Bookman.

Lemos, G. S. (2002). *Um processo de reengenharia com ênfase na qualidade*. São Carlos: Tese de Mestrado.

Martin, S., & Kendall, F. (1997). *UML Distilled: Applying the Standard Object Modelling Language*. Used & New.

Merlo, E. (2003). "Reverse engineering of user interfaces". *Working Conference on Reverse Engineering* (pp. 171-178). Toronto: Baltimore Edition.

Mills, E. (1998). *Software Metrics*, Curriculum Module Software Engineering Institute. Carnegie Mellon University.

Moris, D., & Brandon, J. (1994). *Reengenharia: Restruando a sua empresa*. São Paulo: Makson Books.

Nunes, M., & O'Neill, H. (2003). *Fundamental do UML* (3ª Edição ed.). Portugal: FCA.

- Penteado, R. A. (2000). "Reengineering of legacy Systems based os transformations using the object oriented paradigm". *V Conference on Reverse Engeneering* (pp. 144-153). Havai: Press IEEE.
- Piattinni, M. (2007). *Análise e desenho detalhado de aplicações informáticas de gestão*. Madrid: RA-MA.
- Pilone, D., & Pitman, N. (2006). *UML 2- Rápido e Prático*. Brasil: Letracia Brasil.
- Prado, A., & Novais, E. A. (2001). Reengenharia orientada a objectos de código antigo Progress 4GL. 21-36.
- Pressman. (2005). "Engenharia de Software". New York: McGraw-Hill Higher Education.
- Pressman. (2006). "Engenharia de Software". New York: McGraw-Hill Higher Education.
- Pressman. (2001). *"Software Engineering"* (5ª ed.). New York: McGraw Hill Higher Education.
- Pressan. (2000). *"Software Engineering: A pratitioner's Approach"* (5ª ed.). New York: McGraw-Hill Education.
- Rational. (2000). *Rational Unified Process: Best Pratices for Software Development Teams*. Obtido de <http://www-106.ibm.com/developerworks/rational/library/253.html>.
- Ranson,I & Waren,I. (2002). A metthod for acessing Legacy systems for Evolution. 2ª *Conference on Software Maintenance and Reengineering*, (p. 128). Toronto.
- Schimidt, C. (2004). *Sistemas Integrados de Gestão Empresarial: Uma contribuição no estudo do comportamento organizacional e dos utilizadores na implementação de sitemas*. Brasil: Tese de Doutoramento.
- Silva, A., & Videira, C. (2001). *UML, Metodologias e ferramentas CASE*. Portugal: Centro Atlântico.

Souza, D., & Wills, A. (1998). *Objects, Components and Frameworks with UML- The Catalysis Approach*. USA: Addison Wesley.

Warden R. (2000). *"Reengineering- A practical methodology with commercial applications"*. New York.

Waters, R., & Chikofsky, E. (2004). Reverse Engineering: Progress Along Many Dimensions. Communications of the ACM.

Wazlawick, R. (2005). *Análise e projecto de sistemas de informação orientados a objectos*. Rio de Janeiro: Elsevier Editora.

Yeh, A., Harris, D., & Reubenstein H. (2000). "Recovering Abstract Data Types and Object Instances from a conventional procedural language". *Proceedings of Second Conference on Reverse Engeneering*. Toronto: IEEE/CS Press.