



Universidade do Porto
Faculdade de Engenharia

FEUP



André Fidalgo de Morais Lobão Moniz

Gestão de Processos Jurídicos na Área de Recuperação de Crédito na Indra cpc

3) LEIC
005/MONa

LEIC
Setembro, 2004

**Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação**



**Gestão de Processos Jurídicos na Área de Recuperação de Crédito
na
Indra cpc**

Relatório do Estágio Curricular da LEIC 2004

André Fidalgo de Morais Lobão Moniz

Orientador na FEUP: Prof. António Lucas Soares
Orientador na Indra cpc: Engenheiro João Lima Pinto

Outubro de 2004



004(047.3) LCC/ECC 5202 2005/MONA

| | |
|-------------------------|----------------|
| Universidade do Porto | |
| Faculdade de Engenharia | |
| Biblioteca | |
| Nº | 81524 7 |
| CDU | 004.44 (047.3) |
| Data | 21/03/2006 |

Aos meus pais e Rita

Resumo

O presente relatório documenta o projecto denominado “Gestão de Processos Jurídicos na Área de Recuperação de Crédito” efectuado por André Fidalgo de Moraes Lobão Moniz na empresa Indra cpc, no âmbito da realização do estágio curricular da Licenciatura em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto no ano lectivo de 2003/2004.

O estágio realizado tem como plano de fundo uma área da banca intitulada “Recuperação de Crédito”. Esta é uma componente muito importante para uma instituição bancária e requer uma análise cuidada uma vez que actua sobre a recuperação de valores em dívida para com a instituição.

O departamento de recuperação de crédito é então vital para uma instituição bancária. O facto de um cliente de um banco deixar de pagar um empréstimo gera um enorme volume de informação (no âmbito de um processo jurídico associado a esse incumprimento) que tem que ser armazenada e analisada. Hoje em dia não faz sentido que todo este trabalho de documentação e gestão do processo seja efectuado manualmente. A informatização dessa informação trará enormes ganhos em termos de rapidez e eficiência, no desenrolar do processo, que poderão traduzir-se numa mais valia para a recuperação eficiente do crédito mal parado do banco.

Assim surge a necessidade do departamento de Recuperação de Crédito da Instituição bancária cliente solucionar o problema através de um sistema informático. Esse sistema foi desenvolvido pelo cliente e por questões de confidencialidade não é possível denominar o sistema com um nome. No entanto existia muita informação que não estava reflectida na aplicação e muita gestão que ainda era efectuada manualmente. Assim surge a fase três do sistema a implementar e o seu desenvolvimento foi adjudicado à Indra cpc com o objectivo de implementar novos módulos e efectuar melhorias nos módulos já existentes.

O projecto deste estágio curricular, é assim o desenho e a implementação de uma solução, que visa a evolução da informatização das áreas de pré-contencioso e contencioso de uma instituição bancária. As componentes implementadas gerem todo o processo, desde a preparação do dossier, atribuição de responsáveis, gestão de informação de devedores e controlo do processo judicial.

Extremamente abundante em experiências, não só do ponto de vista técnico mas também humano (pelo contacto estabelecido com equipas internas da instituição bancária e colegas na empresa), abordando uma série de aspectos extremamente actuais na área da Banca, dando uma perspectiva bastante abrangente do negócio.

Agradecimentos

Gostaria de agradecer a todas as pessoas da Indra cpc que me ajudaram e me apoiaram desde o início do meu estágio, nomeadamente o meu orientador de estágio Eng. João Lima Pinto, Carlos Costa, Noélia Peixoto, Carla Azevedo, Manuel Silva, Ruben de Sousa, António Monteiro, Margarida Pinho, Pedro Barros, Luís Reis e Paulo Oliveira.

Gostaria de agradecer ao Professor e orientador António Lucas Soares pelo apoio prestado na elaboração do relatório.

Agradeço à minha família pois sem eles não estaria hoje a terminar o meu curso. Ao meu avô José Morais pelo gosto que me inculuiu na matemática, aos meus outros avós pela ternura e apoio incondicional. Agradeço também à minha bisavó Francisca Lobão pela ilha maravilhosa que me deu.

À Nicky por me dar tudo o que nunca sonhei que existisse e por estar sempre ao meu lado.

Aos meus amigos agradeço pelo apoio e momentos de amizade inesquecíveis em todos os momentos e principalmente durante a minha deslocação por motivos de trabalho. Ao Mário, Pedro, Dois, Late, Rodrigo, Filipe, João Nuno, Ana, Sílvia, Ariana, Steve, Giulio, Rafa, Gonçalo e Luísa.

Índice de Conteúdos

| | | |
|--------|---|----|
| 1 | Introdução..... | 1 |
| 1.1 | Apresentação da Indra cpc..... | 1 |
| 1.2 | O Projecto "Gestão de Processos Jurídicos na Área de Recuperação de Crédito"..... | 2 |
| 1.3 | Organização e Temas Abordados no Presente Relatório..... | 3 |
| 2 | Análise do problema..... | 4 |
| 2.1 | Contexto do projecto..... | 4 |
| 2.2 | Visão geral do processo jurídico..... | 5 |
| 2.2.1 | Entidades envolvidas..... | 6 |
| 3 | Planeamento do Projecto..... | 8 |
| 3.1.1 | Metodologia..... | 8 |
| 3.1.2 | Calendarização..... | 9 |
| 3.1.3 | Análise de Risco..... | 9 |
| 3.1.4 | Organização..... | 10 |
| 4 | Especificação detalhada do sistema..... | 12 |
| 4.1 | Requisitos Funcionais..... | 12 |
| 4.1.1 | Actores..... | 13 |
| 4.1.2 | Visão Geral..... | 13 |
| 4.1.3 | Gerir Penhoras..... | 15 |
| 4.1.4 | Pesquisar Bens..... | 16 |
| 4.1.5 | Notas de Liquidação..... | 18 |
| 4.1.6 | Pareceres..... | 22 |
| 4.1.7 | Cauções..... | 25 |
| 4.1.8 | Avaliação de Bens..... | 27 |
| 4.1.9 | Créditos Reclamados..... | 28 |
| 4.1.10 | Gestão de Bens..... | 29 |
| 4.2 | Requisitos suplementares (Não funcionais)..... | 30 |
| 4.2.1 | Usabilidade..... | 30 |
| 4.2.2 | Desempenho..... | 30 |
| 4.2.3 | Segurança..... | 31 |
| 4.2.4 | Fiabilidade..... | 31 |
| 4.2.5 | Entrega..... | 31 |
| 5 | Desenvolvimento do Projecto..... | 32 |
| 5.1 | Arquitectura do sistema..... | 32 |
| 5.2 | Tecnologias utilizadas..... | 35 |
| 5.3 | Dependências..... | 36 |
| 5.3.1 | Transacções com o sistema central do Banco..... | 36 |
| 5.4 | Modelo de dados..... | 37 |
| 5.4.1 | Desnormalizações..... | 38 |
| 5.5 | Problemas e Soluções..... | 39 |
| 5.5.1 | Migração de dados..... | 39 |
| 5.5.2 | Performance nas consultas..... | 42 |
| 5.5.3 | Desnormalizações com resultados negativos..... | 44 |

| | |
|--|----|
| 6 Conclusões e perspectivas de trabalho futuro | 46 |
| Referências e Bibliografia | 48 |

Índice de Figuras

| | |
|--|----|
| Figura 2-1 Visão geral de um processo jurídico..... | 5 |
| Figura 2-2 Modelo de alto nível de classes do domínio | 7 |
| Figura 3-1 Modelo evolucionário | 8 |
| Figura 3-2 Diagrama de Gantt | 9 |
| Figura 3-3 Equipas envolvidas no projecto | 11 |
| Figura 4-1 Diagrama de pacotes de casos de uso | 13 |
| Figura 4-2 Diagrama de casos de uso do sistema..... | 14 |
| Figura 4-3 Diagrama de actividades da Gestão de Penhoras..... | 15 |
| Figura 4-4 Diagrama informal da sequência de evolução dos estados de penhora | 16 |
| Figura 4-6 Diagrama de actividades da Pesquisa de Bens. | 17 |
| Figura 4-7 Diagrama de actividades para a criação de Notas de Liquidação..... | 18 |
| Figura 4-8 Diagrama de actividades para a criação de Cheques no âmbito da Nota de Liquidação | 20 |
| Figura 4-9 Diagrama de actividades para a criação de Cheques no âmbito do Tribunal | 21 |
| Figura 4-10 Diagrama de actividades da Gestão de Pareceres..... | 22 |
| Figura 4-11 Diagrama de actividades da Gestão de Cauções..... | 25 |
| Figura 4-11 Diagrama de actividades da Avaliação de Bens | 27 |
| Figura 4-12 Diagrama de actividades dos Créditos Reclamados | 28 |
| Figura 4-13 Diagrama de actividades da Gestão de Bens | 29 |
| Figura 5-1 Arquitectura lógica | 33 |
| Figura 5-2 Diagrama genérico de Componentes | 33 |
| Figura 5-3 Diagrama detalhado de Componentes | 35 |
| Figura 5-4 Diagrama Informal do sistema..... | 37 |
| Figura 5-5 Diagrama informal da migração de dados no início do projecto | 40 |
| Figura 5-6 Diagrama informal da migração de dados de simulação final..... | 41 |
| Figura 5-7 Modelo Entidade Relação de tabelas referência | 44 |
| Figura 5-8 Modelo entidade relação de tabelas de referência desnormalizado | 45 |
| Figura 5-9 Proposta para a modelação das tabelas de referência | 45 |

1 Introdução

O presente relatório tem como principal propósito descrever o trabalho realizado pelo aluno André Fidalgo de Morais Lobão Moniz (doravante no presente relatório identificado como o estagiário) durante o estágio na Indra cpc. Assim serão descritas as diversas fases do desenvolvimento do projecto “Gestão de Processos Jurídicos na Área de Recuperação de Crédito”.

Resumidamente o objectivo deste estágio é a integração numa equipa de desenvolvimento e a implementação e melhoria de diversas componentes de uma aplicação de recuperação de crédito numa Instituição Bancária cliente da Indra cpc.

1.1 Apresentação da Indra cpc

Com a missão de se tornar no melhor aliado dos seus clientes no uso intensivo de TI (Tecnologias de Informação), a Indra cpc surge em Abril de 2002, resultado da aliança entre a INDRA (60%) e a CPC IS (40%). Com uma carteira de clientes estável de elevado potencial de crescimento e munida de 450 profissionais, 90% dos quais com formação superior ou elevada especialização certificada, coloca-se entre as primeiras companhias no mercado português de Serviços de Tecnologias de Informação.

A actividade da Indra cpc está baseada numa clara orientação para a área de negócio do cliente, nomeadamente para as Telecomunicações, Banca e Seguros, Energia, Indústria e Comércio, Administração Pública, Tráfego e Transportes. Com experiência adicional nas mais recentes tecnologias a Indra cpc tem uma capacidade de oferta completa e inovadora de soluções próprias, conferidas pelos seus Centros de Competência e investimento que faz na investigação e desenvolvimento. Esta empresa dá especial atenção às soluções de procura emergente que oferecem maior valor acrescentado e conteúdo tecnológico.

Criando e implementando soluções globais para clientes que pretendem investir nas melhores soluções tecnológicas que impõem os melhores resultados no seu negócio, a Indra cpc rege-se pelos seguintes factores:

- Inovação Tecnológica
- Oferta de serviços com valor acrescido
- Apoio na gestão de oportunidades (suporte comercial)
- Excelência no desenvolvimento do projecto: qualidade, rapidez e fiabilidade
- RH com as capacidades e conhecimentos adequados a cada situação

A nível de Internet é já uma das principais empresas portuguesas no desenvolvimento de soluções informáticas. Tem como referências:

- *homebanking* e *corporatebanking* do grupo *BCPAtlântico* (NovaRede, Banco7, Atlântico, BCP Particulares e Negócios, Banco Universitário)
- Leilão Virtual da TAP
- CidadeBCP.pt – portal financeiro

- TituloDirecto.pt – corretora on-line do Finibanco
- Catálogo Telemático de Aprovisionamento Público, da Direcção-Geral do Património
- BPI *Equity Research*, site de publicação de informação sobre bolsa

1.2 O Projecto “Gestão de Processos Jurídicos na Área de Recuperação de Crédito”

O estágio realizado tem como plano de fundo uma área da banca intitulada “Recuperação de Crédito”. Esta é uma área extremamente importante para uma instituição bancária e requer uma análise cuidada. É importante pois, na prática, representa o crédito incobrável, ou seja o dinheiro que é devido à instituição mas que na realidade ainda não foi entregue.

O departamento de recuperação de crédito é vital para uma instituição bancária. O facto de um cliente de um banco deixar de pagar um empréstimo gera um enorme volume de informação (no âmbito de um processo jurídico associado a esse incumprimento) que tem que ser armazenada e analisada. Nos dias que correm não faz sentido que todo este trabalho de documentação e gestão do processo jurídico seja executado manualmente. A informatização dessa informação trará enormes ganhos em termos de rapidez e eficiência, no desenrolar do processo jurídico, que poderão traduzir-se numa mais valia para a recuperação eficiente do crédito mal parado do banco.

Actualmente a área Jurídica da instituição bancária cliente da Indra cpc, está dotada de um sistema informático que proporciona o controlo e execução de funções e actividades descritas na missão deste órgão da Instituição.

O surgimento deste projecto está relacionado com diversos factores:

- A aplicação em questão não é suficiente, nem completa no que se refere à manutenção da informação relevante de um processo jurídico. O controlo da actividade contenciosa, desenvolvida pelos advogados externos, não abrange todas as necessidades desta área de negócio. Assim surge a necessidade de criar novas componentes para responder às necessidades dos utilizadores finais do sistema. Actualmente, ainda existem muitos processos geridos manualmente, que poderiam ser automatizados e informatizados.
- Os resultados dos relatórios de contas da recuperação de crédito demonstram que de ano para ano o volume de capital recuperado não evoluiu, demonstrando as lacunas da versão corrente do sistema.
- A gestão de um processo jurídico, assenta sob uma estrutura de workflow, que determina acções e fases de execução. As regras de negócio estão definidas como condições do referido workflow. Assim é necessário que a aplicação obrigue os utilizadores a realizarem as acções de forma correcta e numa sequência lógica.
- A necessidade de criar uma camada de serviço de dados, tendo em conta as inúmeras alterações ao modelo de dados, centralizando assim o acesso a dados numa camada só. Este ponto é importante também para auxiliar o desenvolvimento de um projecto paralelo, que inclui o desenvolvimento de um site que engloba parte das funcionalidades da aplicação.

Contextualizando historicamente, pode-se dizer que este projecto teve o seu início no ano de 1998 e teve já diversas equipas de *outsourcing* envolvidas. A Indra cpc faz parte do projecto na fase corrente, denominada fase três. Em directa colaboração com uma equipa interna de desenvolvimento o projecto foi realizado nas instalações do cliente.

Dentro deste contexto a equipa de desenvolvimento não efectuou um estudo quanto às tecnologias a usar pois estas já estavam definidas e o objectivo do cliente era continuar o projecto sem mudar de tecnologia.

Embora a equipa da Indra cpc tivesse como objectivo desenvolver diversas componentes, as componentes desenvolvidas pelo estagiário foram as seguintes:

- Gestão de Penhoras
- Pesquisa de Bens
- Notas de Liquidação
- Pareceres
- Cauções
- Créditos Reclamados
- Avaliação de Bens
- Gestão de Bens

Concluindo, o objectivo final do projecto é colocar, de uma forma estruturada e eficiente, a nova versão do sistema em ambiente de produção.

1.3 Organização e Temas Abordados no Presente Relatório

Este relatório encontra-se dividido nos seguintes capítulos:

No primeiro capítulo é apresentado o estágio efectuado, enunciando os seus objectivos, assim como a apresentação da empresa na qual decorreu o estágio.

O segundo capítulo destina-se a efectuar uma exposição do problema, enquadrando-o num contexto mais global do negócio da banca.

Numa perspectiva de análise do problema e de previsão de concretização, está incluído no terceiro capítulo o planeamento, análise de risco, assim como a metodologia utilizada durante o estágio.

O quarto capítulo destina-se a especificar o problema detalhadamente, contendo assim a visão global do problema, os actores envolvidos e os requisitos funcionais acompanhados dos casos de uso de cada uma das componentes implementadas, assim como diagramas de actividades para uma análise funcional mais detalhada. São ainda referidos os requisitos não funcionais do sistema.

No quinto capítulo, denominado desenvolvimento do projecto, é descrito a fase de implementação do projecto assim como os problemas e soluções encontradas.

Por fim, no capítulo Conclusões e perspectivas de trabalho futuro, são apresentadas as conclusões de todo o trabalho realizado e da experiência que foi este estágio na Indra cpc.

2 Análise do problema

2.1 Contexto do projecto

Dando uma definição muito simples sobre o negócio poderemos dizer que o problema começa quando um cliente do banco com um empréstimo falha no cumprimento do contrato.

Frente a este cenário, existem duas áreas pelas quais o processo irá passar. Em primeiro lugar o Pré-Contencioso, ou seja existe uma tentativa por parte do banco em restabelecer o cumprimento do contrato, formalizando um novo plano de pagamentos. No caso de isto não ser possível, é dada a sua entrada em tribunal, sendo assim designado como Contencioso. A partir desse momento tudo é feito pelo banco para reaver o dinheiro que está em dívida. A informática poderá ser valiosa nesta área. É neste contexto que o sistema a implementar é uma mais valia no armazenamento/análise da informação e poderá trazer inúmeras vantagens para o cliente.

Assim o projecto compreende especificar e desenvolver uma solução, que visa a informatização das áreas de pré-contencioso e contencioso de uma instituição bancária. As componentes implementadas gerem todo o processo, desde a preparação do dossier, atribuição de responsáveis, gestão de informação de devedores e controlo do processo judicial.

O projecto foi realizado por uma equipa de desenvolvimento da Indra cpc em colaboração directa com uma equipa da instituição bancária cliente.

2.2 Visão geral do processo jurídico

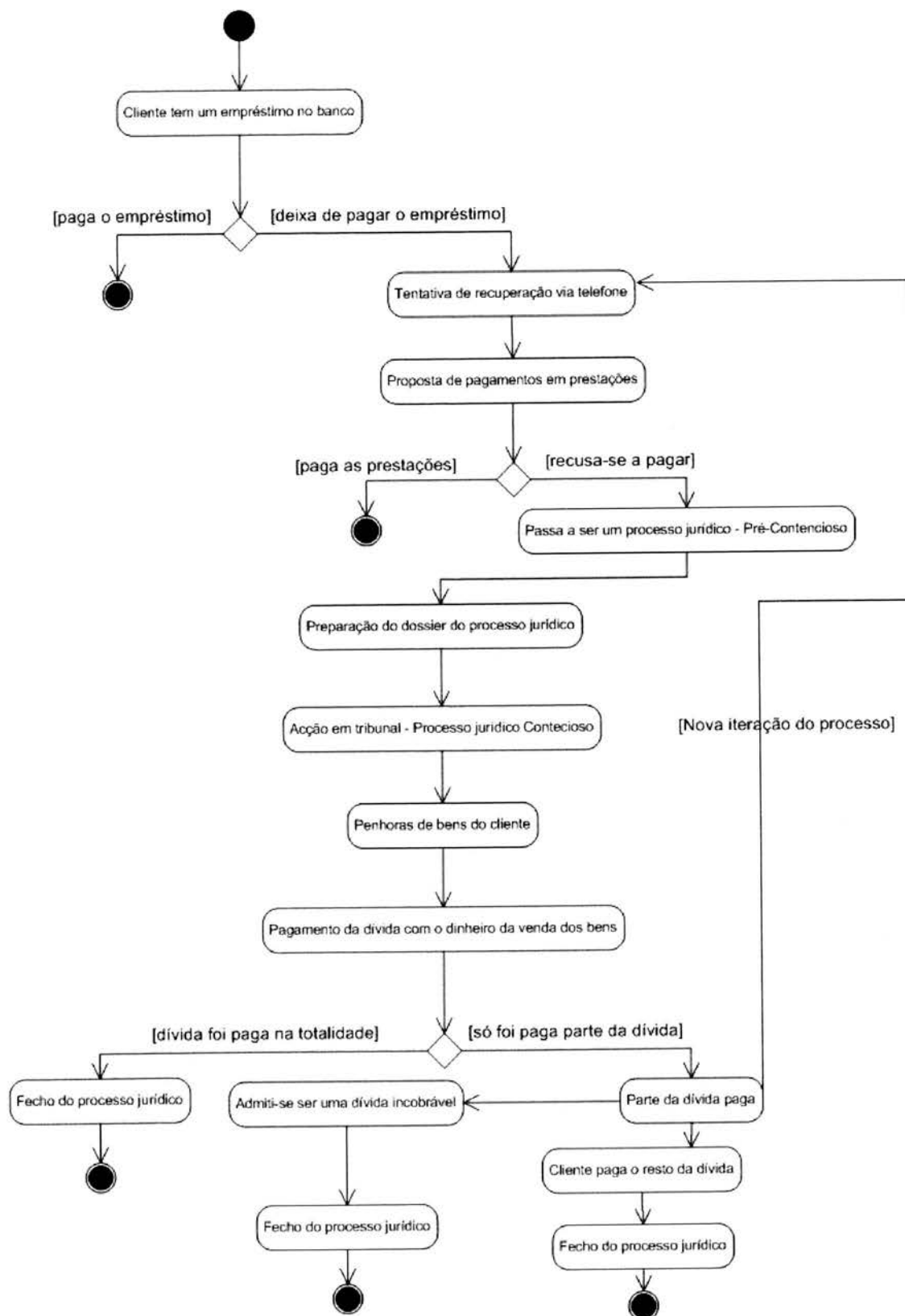


Figura 2-1 Visão geral de um processo jurídico

Assim como a figura anterior demonstra, o processo parte de uma situação de não pagamento da dívida por parte do cliente do banco. De seguida o departamento de recuperação de crédito da instituição bancária tenta reaver o seu dinheiro através de uma proposta de um plano de pagamentos em prestações. No caso de o cliente não ter possibilidade ou simplesmente não quiser pagar é iniciado o processo jurídico dentro deste órgão da instituição. É então necessário preparar todo o dossier do processo, que envolve catalogar os contratos, os clientes, os bens do cliente, bem como outras entidades relevantes para a entrada em execução de um processo em tribunal.

É emitida uma nota de débito com o valor em dívida pelo cliente para entrega em tribunal. Nesta situação temos o processo em contencioso, fase que envolve a gestão de um conjunto de tarefas que envolvem gerir informação como a penhora de bens, os leilões na praça pública, gerir os bens reclamados, criar notas de liquidação, arrematar contratos, etc.

Clarificando, o controlo do processo judicial funciona do seguinte modo: o tribunal passa à penhora de bens do cliente, seguindo estes para leilão na praça pública. O dinheiro obtido pela venda dos bens é pago às instituições (por exemplo ministério público) que tem créditos reclamados naquele bem e o dinheiro correspondente aos créditos reclamados do banco é subtraído ao valor da dívida.

Se nessa altura a dívida ainda se mantiver positiva, o banco, consoante os custos de todo o processo jurídico e os valores (outros bens, por exemplo) que poderão ser usados na recuperação do capital em dívida, decide se é viável continuar o processo, congelar o processo até que existam novas variáveis ou simplesmente considerar que a dívida é incobrável.

2.2.1 Entidades envolvidas

Tendo em conta a visão geral do processo jurídico, poderemos identificar diversas entidades, que fazem parte da preparação do dossier do processo e se relacionam entre si. Um processo jurídico tem associado a si inúmeras entidades, no entanto as que tem relevância para as componentes a especificar pelo estagiário no projecto são as seguintes:

- Clientes
- Bens
- Penhoras
- Tribunais
- Contratos
- Notas de liquidação
- Notas de Débito
- Cheques
- Pareceres

No diagrama seguinte poderemos ver as relações entre estas classes e de uma forma geral perceber o contexto do problema. Este diagrama dá uma noção mais específica do que envolve um processo jurídico na área de recuperação de crédito.

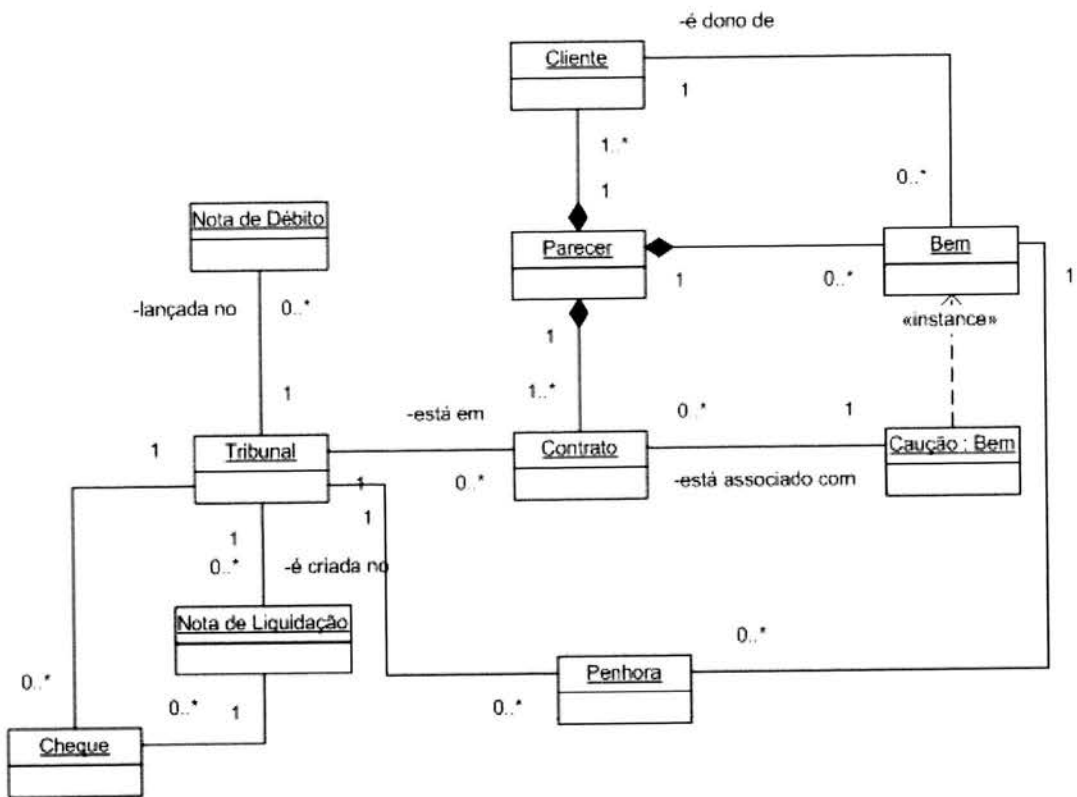


Figura 2-2 Modelo de alto nível de classes do domínio

Este diagrama não é uma modelação de dados do trabalho efectuado, mas sim uma visão genérica contextualizada do problema.

3 Planeamento do Projecto

Nesta secção são abordadas questões relativas ao planeamento do projecto, como este foi conduzido e a razão das decisões tomadas.

3.1.1 Metodologia

A metodologia usada foi o modelo evolucionário como mostra a figura seguinte.

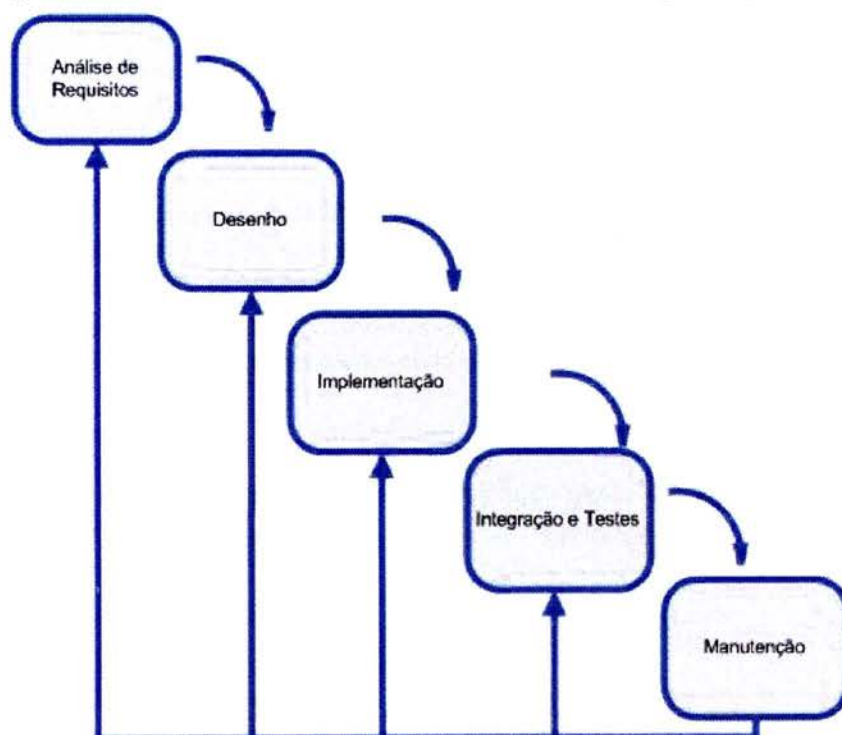


Figura 3-1 Modelo evolucionário

Este foi o modelo adoptado pois os requisitos desta fase do sistema a implementar eram ainda mal entendidos pelos utilizadores do sistema. Na realidade não sabiam exactamente como queriam que o sistema se comportasse nem que entidades queriam associar. Desta forma procedeu-se a primeira análise de requisitos, depois ao desenho, à implementação e por final à integração e testes. Em conformidade com a calendarização, todas as componentes foram desenvolvidas até final de Julho, altura em que começaram os testes dos utilizadores finais, assim como as respectivas correcções e melhorias.

A divisão das fases é descrita de seguida:

- Análise e Especificação de requisitos

Esta fase foi conduzida pela gestora de projecto durante os 2 meses anteriores ao início da implementação. Foram recolhidas as informações relativas às novas componentes a criar assim como as melhorias às componentes já existentes. Quando o projecto voltou a esta fase depois dos testes os requisitos foram definidos em conjunto pelo estagiário, pela gestora de projecto e pelos utilizadores finais.

- Desenho

Durante esta fase foram definidos os modelos de dados a usar nas novas componentes assim como as modificações a fazer ao modelo antigo. Esta fase foi na realidade elaborada em conjunto com o gestor de projecto da equipa da Indra cpc e com o gestor de projecto da equipa da Instituição bancária cliente. Cada fase de implementação de uma nova componente era iniciada pela fase de desenho da mesma. O objectivo foi definir todos os aspectos necessários para construir o sistema que cumprisse a especificação de requisitos levantada anteriormente.

- Implementação

Nesta fase o objectivo era implementar a aplicação de acordo com os requisitos fornecidos e o modelo de dados definido.

- Testes – Melhorias e Correções

Devido à natureza da aplicação, esta fase foi bastante extensa, tendo ocupado cerca de 40 % do tempo total do projecto. Nesta fase os utilizadores finais ocupavam parte do dia a testarem a nova versão da aplicação identificando erros e enviando-os por e-mail. Estes eram catalogados e corrigidos. No caso de melhorias identificadas eram realizadas reuniões para se discutirem as razões para tais alterações, assim como conflitos e tempos de implementação.

3.1.2 Calendarização

No diagrama seguinte é apresentada a calendarização definida para o projecto, onde é organizada temporalmente a execução das actividades do projecto.

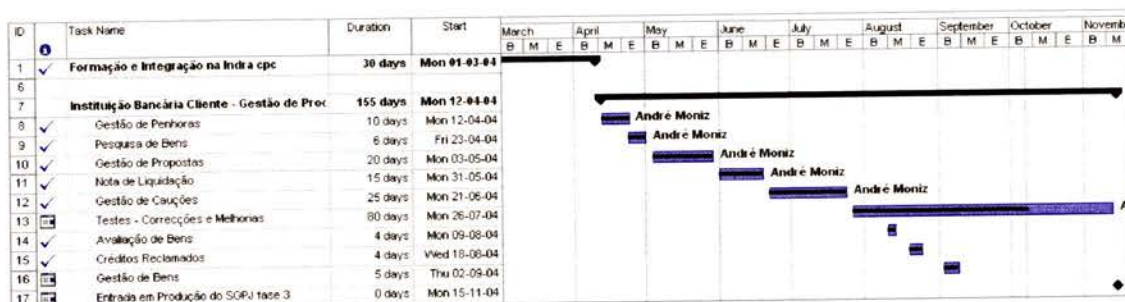


Figura 3-2 Diagrama de Gantt

Existiu, como se pode identificar, um período de adaptação à empresa e métodos de trabalho no qual foram pedidos 3 protótipos de pequena dimensão em diversas tecnologias. Durante este período o estagiário recebeu também formação interna da empresa intitulada “**Good practices using SQL in application development**”. No dia 12 de Abril foi transferido para a sede de Informática da Instituição bancária cliente, em Lisboa, local onde realizou o projecto em questão.

3.1.3 Análise de Risco

Em todos os projectos é imprescindível efectuar uma análise de riscos para tentar antever algumas das dificuldades que se poderão encontrar. Assim num projecto desta dimensão e com o elevado grau de responsabilidade associado, os riscos são grandes e requerem uma análise cuidadosa. Isto porque estamos a falar de uma aplicação de recuperação de crédito e se

a informação não é coerente, ou se os cálculos são mal efectuados isso poderá resultar numa má gestão dos processos jurídicos e resultar em milhares de euros perdidos para o banco.

Assim para minimizar o risco, foram identificados os seguintes riscos:

- Prazos

A questão do cumprimento das datas definidas no planeamento é também muito importante. Apesar de ser de esperar que ocorram ligeiros atrasos, é muito importante garantir que o seu efeito seja minimizado. O cumprimento dos prazos é deveras importante neste projecto, uma vez que dos prazos depende o início dos testes e a entrada da aplicação em ambiente de produção.

O facto de estar a trabalhar junto do cliente aumenta muito a pressão que existe em cumprir os prazos, tornando-se praticamente imperativo cumpri-los a qualquer custo. Num mundo de competição como o de hoje, qualquer contratempo ou atraso pode dar uma imagem negativa da empresa, portanto este ponto é de extrema importância no planeamento do projecto.

É importante ainda referir que, estando envolvidas várias entidades no projecto, o período de férias pode ser problemático.

- Implementação

A migração de dados da aplicação antiga representa um dos principais riscos da implementação, pois o modelo de dados irá ser alterado de forma a ir de encontro aos requisitos estabelecidos.

A aplicação a implementar, como já foi referido, tem inúmeros valores indicativos do estado do processo jurídico. Valores esses, que sendo monetários, na realidade influenciam a forma como o processo irá decorrer de uma forma bastante significativa. Dada a importância destes valores, durante a fase de testes foi necessário fazer uma verificação minuciosa, junto dos utilizadores finais, dos cálculos efectuados. Passo a passo foram identificados erros e corrigidos.

Como também já foi referido, existe uma ligação e sincronização de dados da base de dados local com a base de dados central do banco. Não poderemos deixar que os dados estejam incoerentes, e tendo em conta isto é necessário que a sincronização seja feita de forma eficiente, para que não esteja a ser fornecida informação desactualizada aos utilizadores finais.

3.1.4 Organização

O diagrama seguinte apresenta os elementos envolvidos no projecto.

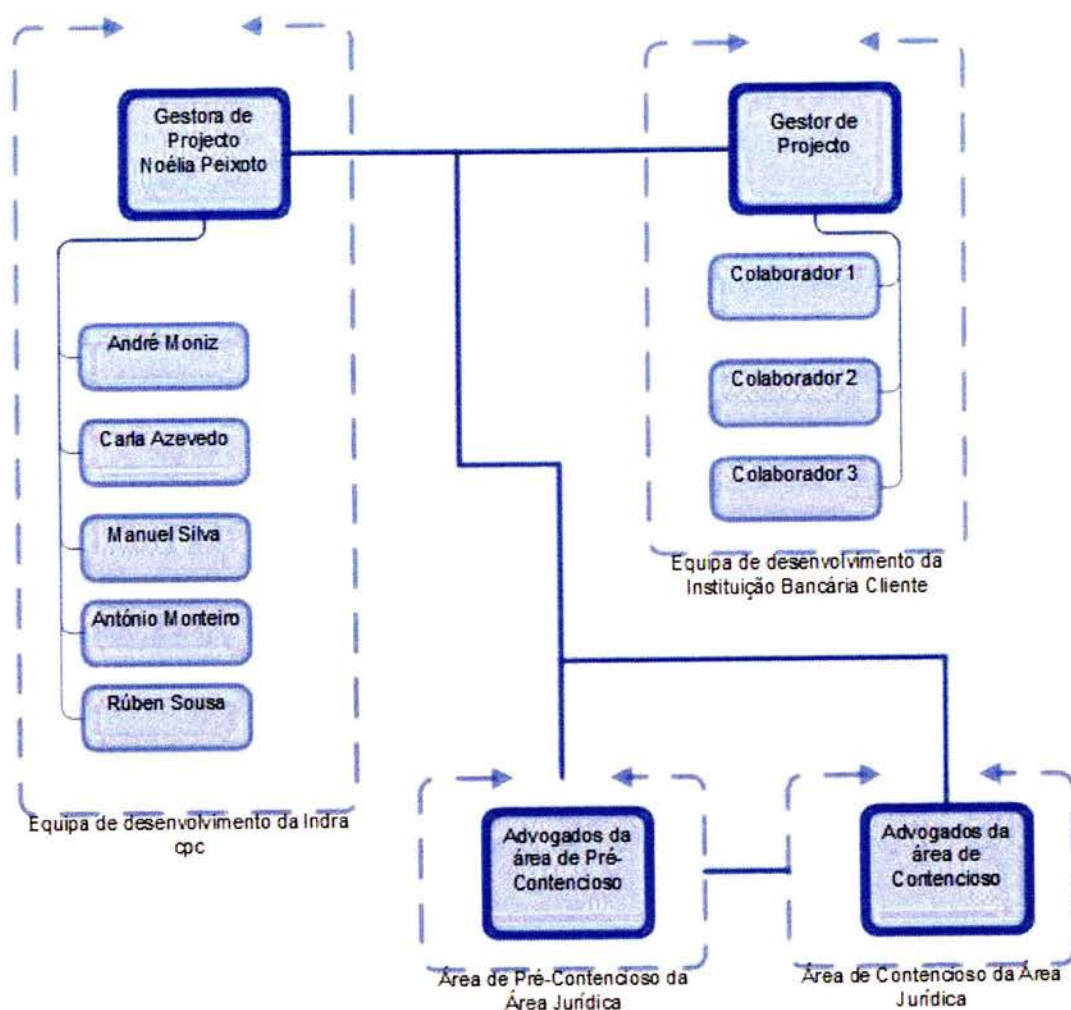


Figura 3-3 Equipas envolvidas no projecto

Como já foi referido o projecto foi elaborado em colaboração directa com uma equipa interna da instituição bancária cliente.

Na fase de testes a equipa apresentava-se com os seguintes elementos Noélia Peixoto, André Moniz, Carla Azevedo e Manuel Silva. Esta equipa teve que interagir no seu dia a dia com a equipa de desenvolvimento do cliente, assim como com o seu gestor de projectos. A condução do projecto foi efectuada em conjuntos pelos dois gestores de projecto. Foi sua tarefa assegurar que as acções fossem efectuadas segundo os prazos estipulados assim como auxiliar os restantes elementos no desenho e implementação da aplicação.

Outras entidades estiveram também envolvidas: a área jurídica do banco, nomeadamente elementos da área de pré-contencioso e de contencioso.

4 Especificação detalhada do sistema

4.1 Requisitos Funcionais

Percebidos alguns dos conceitos do negócio foi necessário começar uma análise mais profunda do que era requerido para a nova versão do sistema a implementar.

As componentes a implementar serão descritas em pormenor juntamente com os seus casos de utilização e diagramas de actividades. Por questões de confidencialidade do negócio do cliente certos casos de uso, assim como certos processos, não foram abordados, no presente relatório, a um nível tão profundo quanto na realidade o foram durante a realização do projecto. São diversos os objectivos de cada componente mas essencialmente podem ser resumidos da seguinte forma:

- Gestão de Penhoras

Gerir os bens quanto ao seu estado de penhora consiste em colocar um bem num determinado estado. Associado a este acto está também um fluxo de estados, a definição de um fiel depositário e de Ónus/Encargos do bem, e a caracterização de um conjunto de dados que lhe estão associados.

- Pesquisa de Bens

Resumidamente, pretende-se que o sistema dê ao utilizador a possibilidade de efectuar pesquisas bastante detalhadas de bens.

- Notas de Liquidação

Criar e manter Notas de Liquidação. Isto implica a possibilidade de entrega de recebimentos dos clientes do banco assim como recalculo da dívida, ou seja, a dívida real que o banco tem a recuperar do cliente.

- Pareceres

Um parecer é uma proposta de pagamento de uma parte da dívida. O sistema requer que seja possível manter os pareceres criados, assim como o acompanhamento do workflow do sistema. Isto implica criação e modificação dos dados da proposta assim como geração de documentos associados às propostas.

- Cauções

O sistema actual não permitia a gestão de cauções. Pretende-se que o sistema mantenha a informação das cauções, assim como possibilite a criação de todos os dados complementares. É necessário ainda a sincronização com os dados do sistema central do banco.

- Avaliação de Bens

Um bem pode ser avaliado de forma a que o advogado tenha uma noção dos valores pertinentes para o desenrolar do processo jurídico. Assim o sistema irá permitir gerir os pedidos de avaliação de bens.

- Créditos Reclamados

Manter os dados sobre os créditos reclamados de um bem num determinado tribunal. Assim como manter a informação da classificação dos créditos.

- Gestão de Bens

Permitir a gestão de toda a informação de um determinado bem, assim como a inserção de novos bens e associação com os clientes do processo jurídico.

4.1.1 Actores

4.1.1.1 Advogados Administrativos

São eles os principais utilizadores finais do sistema e a sua função é gerir toda a fase de Pré-Contencioso, assim como parte de Contencioso, dos processos jurídicos.

4.1.1.2 Solicitadores e Advogados

Os solicitadores e os advogados são utilizadores das componentes do sistema mais interligadas com a área de contencioso de um processo jurídico. Como para a aplicação estes terão o mesmo papel, são identificados nos diagramas com um só actor.

4.1.2 Visão Geral

No diagrama seguinte poderemos ter uma visão geral do sistema.

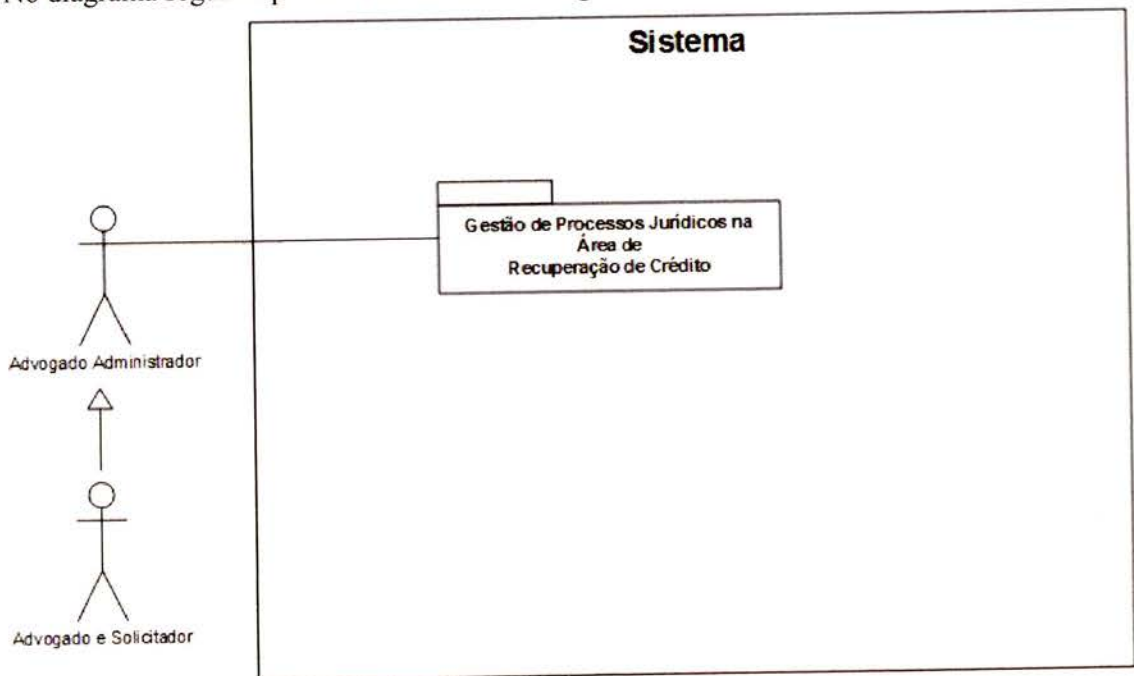


Figura 4-1 Diagrama de pacotes de casos de uso

Este pacote inclui todos os casos de uso definidos pelo estagiário. Na imagem seguinte teremos identificados os casos de uso para as componentes a implementar. Para definir melhor os requisitos foram utilizados diagramas de actividades que descrevem com mais profundidade a análise funcional e acompanham a descrição de cada caso de uso.

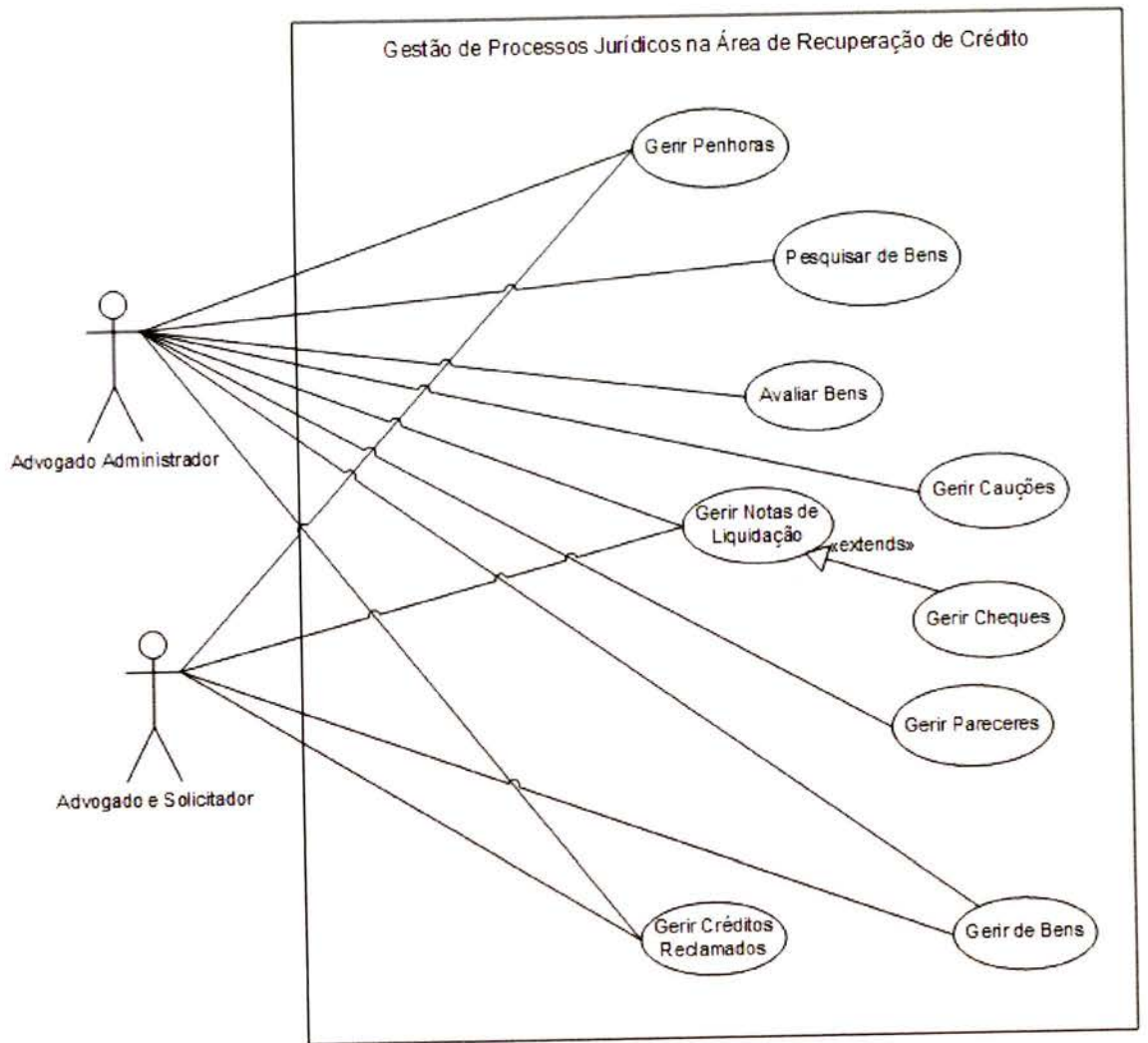


Figura 4-2 Diagrama de casos de uso do sistema

4.1.3 Gerir Penhoras

A Gestão de Penhoras é uma componente que tem como principal objectivo colocar determinado bem num determinado estado de penhora. Os bens dizem respeito aos clientes do processo e os estados de penhora possíveis são: Requerido, Penhorado, Registrado, Cancelado, Insucesso, Desistido.

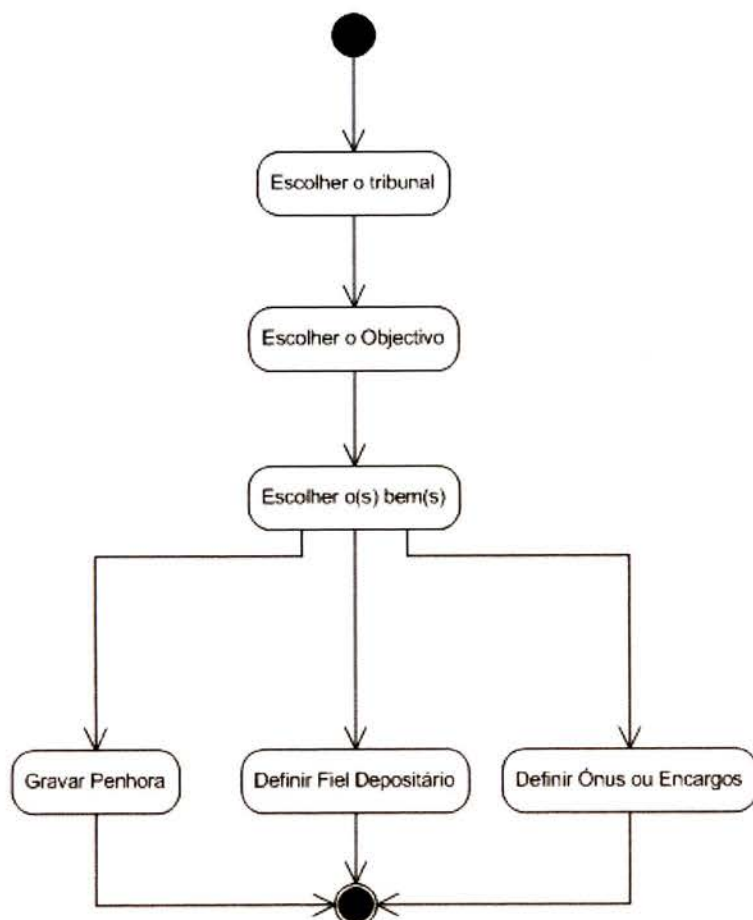


Figura 4-3 Diagrama de actividades da Gestão de Penhoras

- Escolher o tribunal

Para se definir um estado de penhora é necessário que o utilizador do sistema a implementar defina o tribunal no qual o bem irá ter o seu estado associado. Sem essa definição de tribunal não é possível definir um estado de penhora para o bem. Um estado de penhora está assim sempre associado a um tribunal.

- Escolher o objectivo

Depois de definido o tribunal é necessário definir um objectivo para o bem. Esse objectivo é o estado de penhora no qual se quer colocar o bem. Como existe um conjunto de regras lógicas para a definição do estado de penhora, a ideia principal é definir o objectivo para podermos visualizar quais os bens que poderão ser colocados nesse estado de penhora. Assim, por exemplo, se seleccionarmos o estado “Penhora” iremos visualizar todos os bens que podem ser penhorados no tribunal seleccionado anteriormente.

O utilizador final decide em que estado de penhora quer colocar o bem consoante as restrições que as seguintes regras impõem. A figura seguinte explica a sequência lógica dos estados possíveis de um bem.

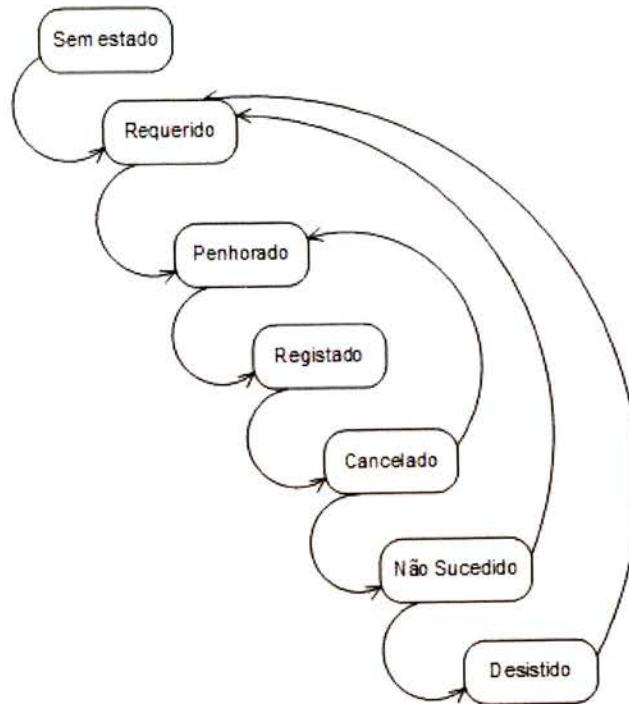


Figura 4-4 Diagrama informal da sequência de evolução dos estados de penhora

Foi ainda definido pelos utilizadores finais que ao seleccionar um determinado objectivo (diferente do “Todos”), aparecesse uma listagem de todos os bens que pudessem ser colocados nesse estado seleccionado e que houvesse a possibilidade de seleccionar nessa listagem determinados bens e aplicar esse estado aos mesmos. Ou seja, foi requerido que houvesse a possibilidade de multi-selecção de bens. No caso do estado “Todos” o estado é seleccionado pelo utilizador e os bens são modificados individualmente.

- Gravar o estado de penhora

Depois de seleccionar os bens e o estado de penhora, é necessário definir a data da aplicação do estado e assim é possível gravar o estado de penhora para os bens seleccionados.

- Definir fiel depositário

Ao escolher esta opção o utilizador poderá associar ao bem seleccionado um fiel depositário.

- Definir Ónus/Encargos

O sistema a implementar terá que dar a possibilidade de definir ónus ou encargos para um determinado bem.

4.1.4 Pesquisar Bens

Os clientes de um processo jurídico poderão ter bens associados que poderão fazer parte do pagamento da sua dívida. Esta componente tem como objectivo, como o nome indica, efectuarmos a pesquisa de bens. Um bem pode ser de diversos tipos, como por exemplo, uma

casa ou um automóvel que por sua vez tem características bens distintas. Assim é necessário ajudar o utilizador a pesquisar bens com características tão distintas.

Basicamente para pesquisar bens o utilizador tem que definir filtros para a sua pesquisa. Estes poderão ser de dois tipos: dados gerais a todos os tipos de bens ou dados específicos a cada tipo de bem.

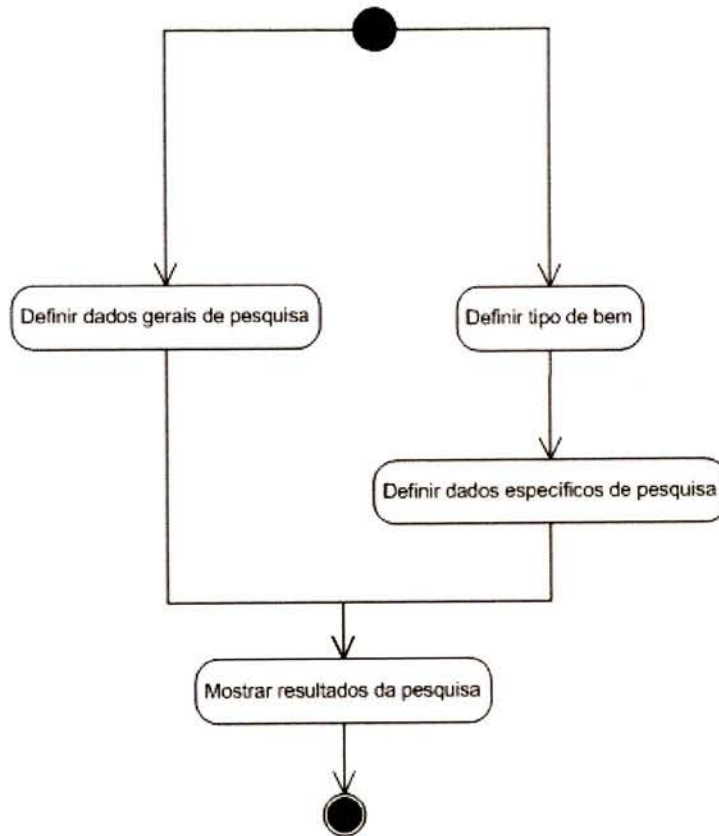


Figura 4-5 Diagrama de actividades da Pesquisa de Bens.

- Definir critérios de pesquisa

Para se efectuar uma pesquisa é indispensável definir filtros de pesquisa. Pois são estes que definem que tipo de bem estamos a pesquisar. Existem então critérios específicos e gerais.

- Definir dados gerais

Os dados gerais são os dados comuns a todos os bens. Assim o utilizador pode definir os critérios de pesquisa gerais.

- Definir tipo de bem a pesquisar

Para o utilizador do sistema definir os critérios de pesquisa específicos de um determinado bem terá sempre que definir que tipo de bem pretende pesquisar. Desta forma o utilizador terá disponível todos os dados específicos desse tipo de bem.

- Definir os dados específicos

Cada dado específico pode ser definido para tornar a pesquisa ainda mais precisa.

- Definir valores para o campo específico seleccionado

Como existem valores de dados específicos que já tem valores pré-definidos, como é o caso por exemplo da freguesia de um imóvel, como alternativa à inserção do nome da freguesia pelo utilizador, terá que ser dada a possibilidade de inserção de uma forma consistente e uniforme. Neste caso deverão aparecer os distritos, os concelhos e as freguesias em listagens de usabilidade amigável. Este pequeno exemplo serve para ilustrar que a definição dos valores dos dados específicos da pesquisa de um bem, deve ser suportada pelo sistema de uma forma intuitiva e que facilite a pesquisa do bem pretendido.

- Visualizar resultado da pesquisa

Depois de definidos os critérios de pesquisa é necessário visualizar que bens foram encontrados. Mais uma vez a listagem de bens é a solução pretendida. Nesta listagem devem estar contidos os valores dos dados gerais mais relevantes.

4.1.5 Notas de Liquidação

Uma nota de liquidação é uma entidade relacionada com o tribunal e com os contratos do processo. O primeiro diagrama apresenta as actividades que fazem parte do processo de criação de uma nota de liquidação.



Figura 4-6 Diagrama de actividades para a criação de Notas de Liquidação

- Escolher tribunal

Como a nota de liquidação só faz sentido no âmbito de um tribunal é sempre necessário seleccionar o tribunal antes de efectuar qualquer operação. Assim um processo jurídico sem tribunal associado não poderá ter notas de liquidação.

- Visualizar total de recebimentos sem Nota de Liquidação

Quando se selecciona o tribunal o utilizador poderá visualizar o total do valor dos recebimentos que ainda não estão associados a nenhuma nota de liquidação. Este valor é um indicativo importante para os advogados que gerem o processo jurídico e ficará agregado à visualização das notas de liquidação já criadas no âmbito desse tribunal seleccionado.

- Visualizar Notas de liquidação do tribunal

Em conjunto com o ponto referido anteriormente, é requerido que exista uma listagem de todas as notas de liquidação já criadas assim como a data de criação e o valor das mesmas.

Ao seleccionar uma Nota de Liquidação o utilizador terá a possibilidade de ver os detalhes dessa nota de liquidação. Esses detalhes incluem a listagem dos recebimentos associados (Petição Inicial, Taxas, Entregas e Vendas), listagem dos créditos privilegiados do tribunal, os encargos do tribunal e os valores associados à Nota de Liquidação.

- Visualizar recebimentos

O utilizador terá a possibilidade de visualizar os recebimentos relativos à nota de liquidação seleccionada.

- Efectuar entrega de recebimentos

O utilizador do sistema poderá efectuar entrega de recebimentos. Poderá definir o montante e a data de entrega do recebimento.

- Modificar e eliminar recebimentos

O utilizador poderá eliminar recebimentos e poderá ainda modificar os montantes dos recebimentos já entregues.

- Visualizar Encargos do tribunal

Este componente deve ainda contemplar a possibilidade de visualizar os encargos do tribunal no qual a Nota de Liquidação está a ser criada.

- Visualizar créditos privilegiados do tribunal

Será possível visualizar os créditos privilegiados do tribunal no qual a nota de liquidação está a ser criada.

- Gerir créditos reclamados do tribunal

Existência de ligação à componente de gestão de créditos reclamados no tribunal no qual a nota de liquidação está a ser criada.

- Definir restantes valores da Nota de Liquidação

Existe a possibilidade de o utilizador modificar os valores (já descritos anteriormente) associados à nota de liquidação a ser criada.

- Criar Nota de Liquidação

Para se criar uma nota de liquidação é necessário definir o plano de recebimentos. Esse plano parte das petições iniciais dos contratos que estão associados às notas de débito do processo

jurídico. De seguida terão que ser aplicadas as alterações das taxas de juro. E por sua vez são aplicados os recebimentos que o utilizador desejar inserir. Este procedimento de inserção de recebimentos e cálculo da dívida exequenda era efectuado numa folha de cálculo *Excel* e é pretendido que faça parte do sistema a desenvolver.

O utilizador poderá ainda definir os diversos valores associados às notas de liquidação.

Contudo, caso os créditos privilegiados não estejam confirmados ou não tenham graduação não é possível criar a nota de liquidação.

- Seleccionar e visualizar detalhes da Nota de Liquidação

Para além da criação das Notas de Liquidação temos que ter em conta que vão existir cheques de dois tipos, precatório cheque e cheque de custas parte, que irão ser relacionados ou com o tribunal, ou com a nota de liquidação

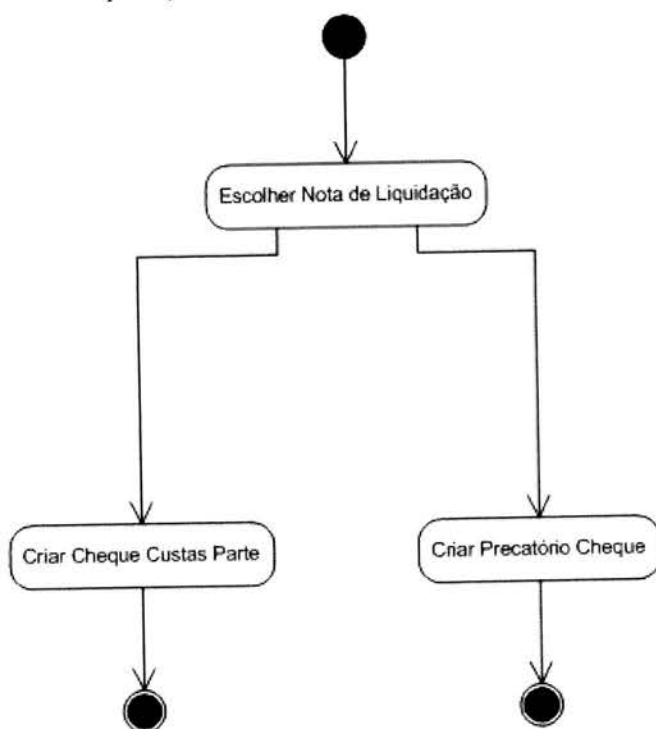


Figura 4-7 Diagrama de actividades para a criação de Cheques no âmbito da Nota de Liquidação

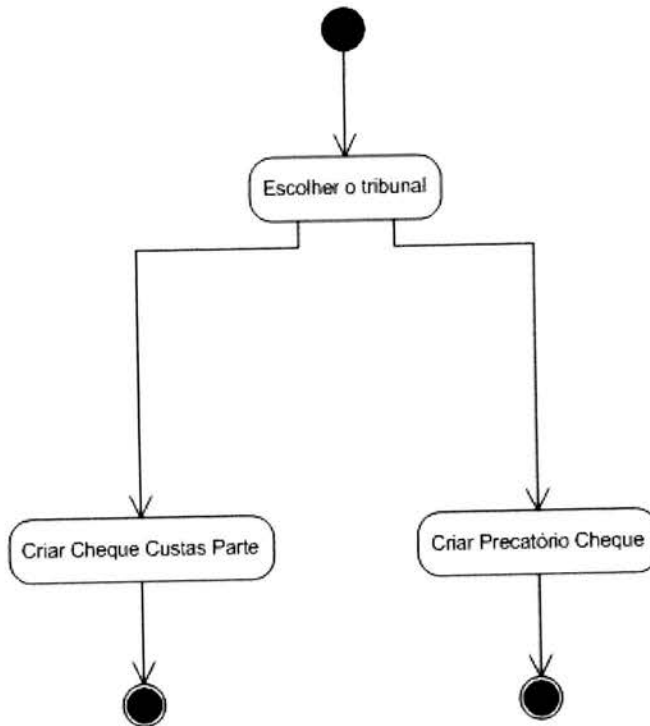


Figura 4-8 Diagrama de actividades para a criação de Cheques no âmbito do Tribunal

- Criar precatório cheque

É possível criar precatórios cheques no âmbito ou do tribunal ou de uma Nota de Liquidação criada.

- Visualizar precatórios cheques do tribunal

Os precatórios cheque de todas as notas de liquidação do tribunal seleccionado irão aparecer numa outra listagem, dando ainda a possibilidade de visualizar o estado corrente de cada um dos cheques.

- Visualizar precatórios cheques da Nota de Liquidação

Deverá existir ainda a possibilidade de seleccionar uma nota de liquidação e visualizar só o precatório cheque dessa nota de liquidação.

- Modificar precatórios cheques

Seleccionando um precatório cheque será possível alterar o estado desse mesmo. Contemplar ainda que poderão existir observações associadas à mudança desse estado. Ao executar esta acção irá ser aplicado um evento (de acordo com o workflow definido) relativo ao estado modificado.

- Criar cheques custas parte

É possível criar cheques custas parte no âmbito ou do tribunal ou de uma Nota de Liquidação criada.

- Visualizar cheques custas parte do tribunal

Para cada tribunal seleccionado irá ser possível visualizar, como no caso dos precatórios cheque, todos as custas parte associadas às notas de liquidação desse mesmo tribunal. Nesta listagem o utilizador poderá visualizar um indicador que demonstrará se as custas partes foram ou não recebidas.

- Visualizar cheques custas parte da Nota de Liquidação

Análogo ao ponto anterior, mas no âmbito da nota de liquidação seleccionada e não do tribunal.

- Efectuar a recepção dos cheques custas parte

O utilizador terá a possibilidade de indicar que as custas foram recebidas. Esta acção irá desencadear a aplicação de um evento associado, conforme o workflow.

4.1.6 Pareceres

Uma proposta é uma entidade associada a Contratos, Clientes, Pareceres e Bens. Na prática é uma proposta de pagamento do cliente ao banco, de forma a que não haja necessidade de se entrar com o processo em tribunal.



Figura 4-9 Diagrama de actividades da Gestão de Pareceres

- Criar Proposta

Caso não existam propostas para o processo seleccionado, o sistema a implementar deverá apresentar o ecrã de inserção de uma nova Proposta. Para se criar uma proposta nova é

necessário definir os dados da proposta e seleccionar os clientes, proponentes, contratos e bens que farão parte da proposta.

- Definir dados da Proposta

O utilizador poderá criar propostas sendo, para isso, necessário definir os dados da proposta.

- Escolher Clientes, Proponentes, Contratos e Bens

Para além da definição dos dados da proposta, é necessário escolher quais os contratos, proponentes, clientes e bens associados à proposta a criar. Estes dados deverão estar listados em listas individuais para cada uma das entidades. Os proponentes e clientes a listar são os associados ao processo que está aberto pela aplicação.

Os contratos e os bens a listar são os que estão relacionados com todos os clientes do processo, podendo estar ou não associados com o mesmo.

- Adicionar Proponentes e Contratos externos ao processo

O sistema a implementar deverá ainda contemplar a possibilidade de se adicionar às listagens proponentes e de contratos externos ao processo. Ou seja, entidades não associadas ao processo jurídico.

- Visualizar Propostas criadas

É requerido que o sistema tenha uma listagem das propostas criadas no âmbito do processo jurídico e para além dessas, outras propostas que estejam interligadas com o processo pelos contratos externos associados.

- Ver histórico de Propostas

Na listagem de propostas o utilizador poderá visualizar todas as propostas já eliminadas, ou seja o histórico das propostas do processo.

- Escolher Proposta

O sistema terá a possibilidade de visualizar os dados das propostas já criadas.

- Visualizar valores da Proposta

Todos os valores são calculados e obtidos a partir de formulas descritas no próximo sub capítulo.

Visualizar os valores da proposta compreende visualizar:

- Proponentes, Clientes e Bens da proposta
- Valores da Dívida Comercial da proposta
- Valores da Dívida Judicial da proposta
- Totais da proposta
- Dados da etapa actual da proposta
- Contratos da proposta
 - Dados acerca dos processos relacionados com os contratos
 - Responsáveis e Fases dos processos
 - Valores dos contratos

- Documentos a produzir (templates) e já produzidos
- Etapas da proposta
- Imprimir

O utilizador poderá imprimir o relatório relativo à proposta seleccionado com todos os dados apresentados na aplicação, denominado mapa.

- Modificar Proposta

Todos os dados da proposta, passíveis de serem modificados, poderão ser alterados.

- Eliminar Proposta

O sistema deverá contemplar a possibilidade de eliminar uma proposta. Esta não deve ser apagada por completo mas sim fazer parte do histórico de propostas.

- Definir Etapas

Uma proposta segue um ciclo de vida em que diversas etapas vão alterando o seu estado.

Existe assim um fluxo associado à definição das etapas da proposta. Assim o utilizador deverá seguir esse mesmo fluxo (definido previamente pelo utilizador). Associada a essa etapa estará uma data, uma entidade e observações, bem como o(s) documento(s) produzido pela etapa.

Todo este processo de fluxo de etapas está associado a um workflow definido. Portanto ao definir uma etapa é aplicado um evento apropriado a cada etapa. Não é possível por razões de confidencialidade colocar no presente relatório o workflow relativo às etapas de uma proposta.

- Modificar etapa

O sistema dará a possibilidade ao utilizador de alterar, em qualquer altura, certos dados da etapa como a data, a entidade e as observações.

- Eliminar etapa

O utilizador poderá também eliminar uma etapa.

- Definir resultado da Proposta

A proposta poderá ter um resultado associado consoante o tipo de proposta. Esse pode estar ou não definido. O utilizador poderá então definir o resultado através desta opção.

- Definir plano associado à proposta

O sistema deve dar a possibilidade de associar a proposta a um plano de pagamentos previamente definido noutra componente do sistema.

- Arquivar documentos

Deverá ser dada a possibilidade ao utilizador de arquivar os documentos seleccionados num directório próprio da proposta em questão.

- Criar documentos

Será possível criar documentos Word, a partir de templates pré-definidos, relativos à proposta seleccionada. Os documentos serão relativos ao tipo de proposta e o utilizador poderá definir quais os documentos que estão associados aos diferentes tipos de propostas.

- Visualizar documentos

Seleccionando um ou vários documentos será possível visualizar de uma forma rápida o documento anteriormente produzido.

4.1.7 Cauções

A definição literal de uma caução é um bem posto à disposição para garantir o pagamento de uma dívida, caso esta entre em incumprimento. No entanto, esse bem tem agregado informações diversas que, como requerido, estão divididas nas seguintes secções:

1. Dados Complementares
2. Constituição Manutenção da Fracção
3. Inscrição Matricial
4. Descrição Predial
5. Registo Transmissão
6. Aquisição e Registo da Hipoteca Escritura

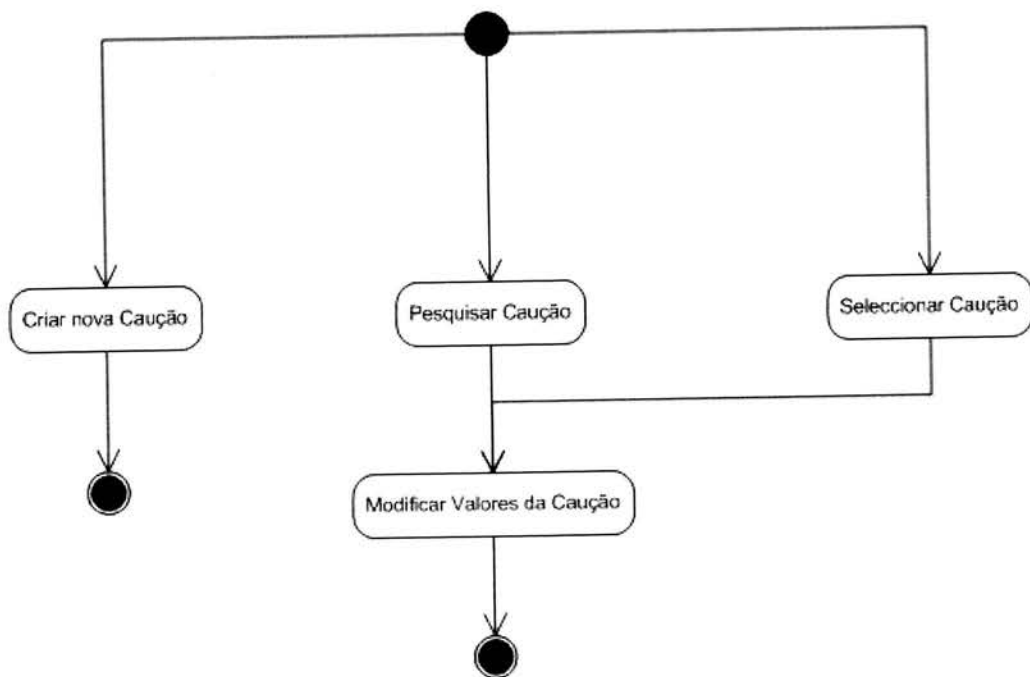


Figura 4-10 Diagrama de actividades da Gestão de Cauções

- Pesquisar cauções

Para um utilizador pesquisar dados sobre uma caução terá que escolher o processo jurídico, seleccionar na listagem de bens um bem que seja uma caução e depois poderá ver e modificar os detalhes dessa caução. Outra forma de pesquisa será pelo numero da caução.

- Modificar Caução

Depois de seleccionada a caução poderemos alterar todos os valores associados a essa mesma caução.

- Definir dados complementares

- Constituição Manutenção da Fracção
- Inscrição Matricial
- Descrição Predial
- Registo da Hipoteca Escritura
- Registo Transmissão/ Aquisição
- Visualizar e adicionar Contratos associados à caução
- Inserir Caução

Outra possibilidade será a de criar uma caução nova

Não é possível aprofundar mais a informação que é mantida no âmbito das cauções por razões de confidencialidade.

O sistema permite visualizar os contratos associados à caução que estamos a consultar. Ainda terá que existir uma funcionalidade que associe outros contratos à mesma caução.

4.1.8 Avaliação de Bens

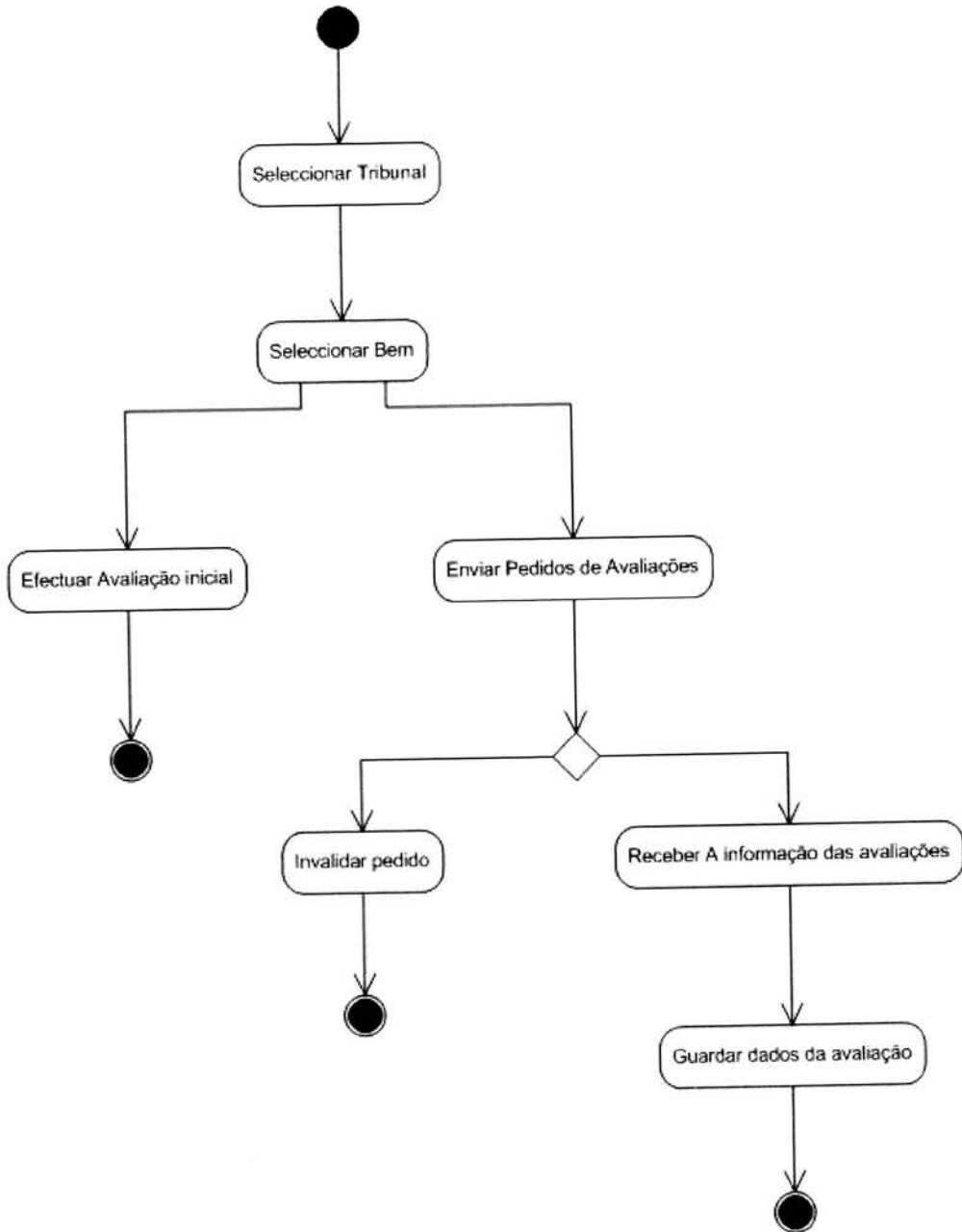


Figura 4-11 Diagrama de actividades da Avaliação de Bens

- Inserir avaliação inicial

O utilizador do sistema poderá inserir uma avaliação inicial do bem seleccionando. Essa avaliação estará sempre associada ao bem como sendo a inicial. Uma avaliação tem agregados diversos valores bem como o engenheiro ou empresa que o avaliou.

- Fazer pedidos de avaliação

O sistema permitirá efectuar pedidos de avaliação de um determinado bem. Esse pedido irá ficar num estado pendente, esperando assim pela avaliação.

- Gravar dados da avaliação pedida

Depois de avaliado o bem o utilizador do sistema poderá inserir os dados da avaliação guardando assim no histórico de bens que este foi avaliado com os dados inseridos.

- Invalidar pedido

O pedido pode também ser invalidado ficando registado como pedido invalidado.

- Visualizar historial de pedidos

O sistema terá disponível uma listagem com informação relativa a todos os pedidos já feitos sobre o bem seleccionado.

4.1.9 Créditos Reclamados

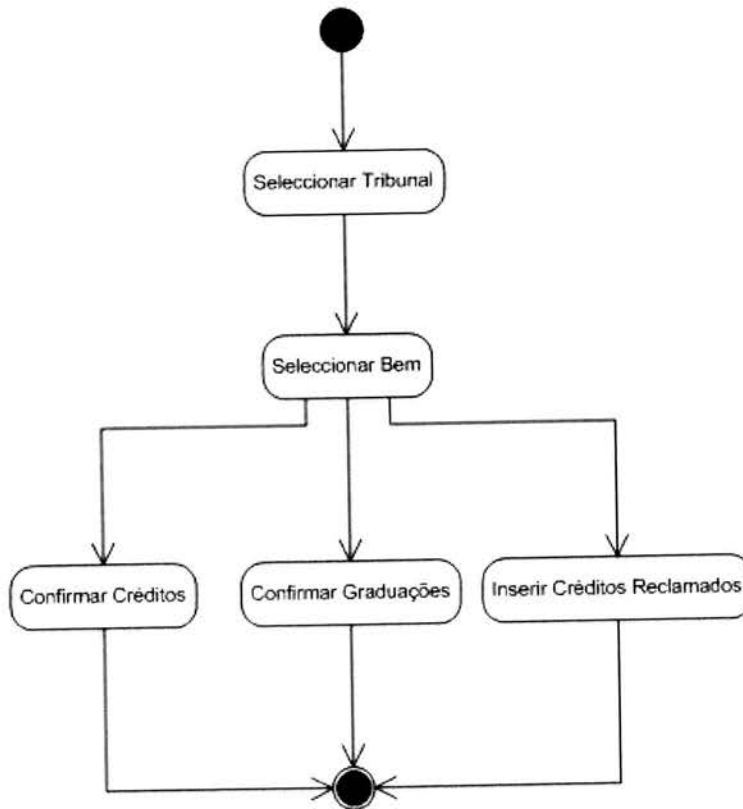


Figura 4-12 Diagrama de actividades dos Créditos Reclamados

- Seleccionar tribunal

O sistema só permite inserir créditos reclamados de um bem no âmbito de um tribunal. Portanto é necessário escolher um tribunal para poder reclamar créditos.

- Seleccionar bem

O sistema só permite inserir créditos reclamados de um bem no âmbito de um tribunal. Portanto é necessário escolher também um bem para poder reclamar créditos.

- Visualizar créditos reclamados do bem no tribunal

Depois de seleccionado o tribunal é possível visualizar todos os créditos reclamados deste bem neste tribunal.

- Inserir Crédito

O sistema permitirá inserir créditos possibilitando ao utilizador a escolha da entidade que reclama o crédito, o valor do crédito e a sua graduação.

- Modificar Crédito

É possível modificar os dados de um crédito reclamado, a menos que os créditos já tenham sido confirmados.

- Confirmar as graduações

A graduação de um crédito indicará a sua prioridade em relação aos outros créditos. Depois de definidas as graduações de todos os créditos, o utilizador poderá confirmar as suas graduações para que se mantenham inalteráveis.

- Confirmar os créditos

Depois de definidos os créditos, o utilizador poderá confirma-los para que se mantenham inalteráveis.

- Alerta para créditos privilegiados

É importante para o utilizador ter identificados os créditos que são privilegiados, ou seja os que tem prioridade sobre os créditos da instituição bancária.

4.1.10 Gestão de Bens

Basicamente na gestão de bens pretende-se manter a informação de um bem. Um bem é uma garantia que o cliente apresenta quando pede um empréstimo a um banco. É importante o sistema permitir a gestão de bens dos clientes dos processos jurídicos.

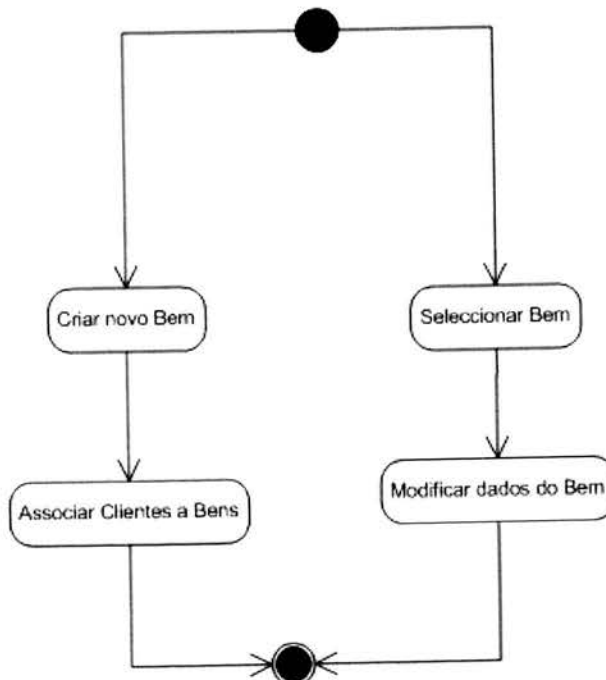


Figura 4-13 Diagrama de actividades da Gestão de Bens

- Criar novo Bens

O sistema tem que permitir ao utilizador inserir novos bens num determinado processo jurídico

- Manter informação genérica

O bem, como já foi referido na pesquisa de bens, tem dados genéricos e específicos. Assim é possível manter a informação relativa aos dados genéricos do bem que previamente foi seleccionado. O mesmo se passa com a informação específica.

- Associar Clientes a bens

É necessário também que o gestor de bens permita associar um proprietário ao bem. Ou seja, um cliente do processo jurídico pode ser o proprietário de diversos bens.

4.2 Requisitos suplementares (Não funcionais)

4.2.1 Usabilidade

Um aspecto emergente e cada vez mais importante é a questão da usabilidade dos sistemas informáticos. Qualquer sistema deve ser fácil, intuitivo e apelativo para quem o utiliza, com ainda maior impacto numa situação em que o trabalho é volumoso e rotineiro.

Quando a informação a introduzir num sistema é numerosa e/ou complexa, este tem que ter uma interface que seja *user-friendly*, esclarecedora e que ajude o utilizador a não cometer erros. Para garantir isso, o sistema foi optimizado no sentido de simplificar a interacção homem-máquina, não descurando o objectivo do sistema que é manter toda a informação útil aos utilizadores. Obviamente, hoje em dia as *interfaces web* apresentam possibilidades infinitas quando comparadas com tecnologias como o *Visual Basic 6*, mas de qualquer forma o aspecto da usabilidade não pode ser descurado e sempre em colaboração directa com o utilizador final deve-se tentar encontrar um equilíbrio entre as necessidades do mesmo e conceitos teóricos de usabilidade.

4.2.2 Desempenho

O desempenho num sistema como este é de extrema importância. Embora o sistema não tenha um volume de dados muito elevado e o número de utilizadores também não seja elevado, é necessário que os diferentes módulos implementados sejam optimizados.

Onde se nota que o desempenho desce, nesta aplicação, é em consultas mais complexas à base de dados. Desta forma é necessário procurar sempre melhores formas de consultar a informação tendo em conta boas práticas em SQL, tentando sempre reduzir ao máximo os custos das consultas, analisando a fundo cada plano de execução e estudando bem os índices. Agregado a este requisito foi imposto na realidade um número, um limite para o desempenho das consultas. Assim foi requerido que só mesmo em situações extremas é que uma consulta poderia demorar mais do que 2 segundos.

4.2.3 Segurança

O sistema garante a fiabilidade em termos de segurança (falando de autenticação), pois a aplicação desde a sua fase número um, que efectua autenticação através do utilizador Windows. Assim ao abrir o sistema informático o utilizador nunca tem que inserir nenhuma password, e os seus privilégios estão definidos de acordo com o seu utilizador Windows. Esta aplicação não permite qualquer tipo de acesso em máquinas fora da rede do banco.

4.2.4 Fiabilidade

O sistema apresentado deve revelar um grau elevado de fiabilidade, ou seja, as pessoas que o usam devem sentir-se confortáveis e confiantes na utilização do sistema para realizar o seu trabalho diário. Os requisitos funcionais descritos devem estar embutidos de tal maneira no sistema, que a aplicação, na prática dos processos que derivam destes, seja uma realidade e possa contribuir efectivamente como um acrescento de valor para a Instituição bancária.

4.2.5 Entrega

A colocação dos novos componentes, assim como de todo o modelo de dados, em ambiente de produção está planeada para o dia 15 de Novembro.

5 Desenvolvimento do Projecto

Neste capítulo é pretendido focar os principais aspectos do desenvolvimento do projecto. Foi sempre tido em conta o nível de confidencialidade dos dados e por isso muita informação não foi passível de ser colocada no presente relatório.

Assim, fazendo um pequeno resumo, o estagiário ocupou uma posição de programador na equipa de desenvolvimento, embora tenha tido sempre a intenção de não se conformar com as suas tarefas de programador e auxiliar também o desenho do sistema e a gestão do projecto. Esse trabalho envolveu toda a parte de lógica de negócio e serviço de dados de cada uma das componentes, sendo as regras de negócio por vezes bastante complexas e de desenvolvimento moroso.

Neste capítulo é descrita a arquitectura das componentes implementadas pelo estagiário, assim como as principais dependências do sistema. Não foi possível por questões de confidencialidade descrever os modelos de dados das componentes implementadas assim como propostas de alteração dos modelos. No fim do capítulo são identificados problemas e soluções encontradas ao longo do projecto.

5.1 Arquitectura do sistema

O desenvolvimento do projecto foi conduzido segundo a metodologia definida anteriormente.

A título introdutório, pode-se dizer que a estruturação de aplicações tem vindo a evoluir ao longo dos tempos. Se antigamente eram pensados dois módulos, o cliente e o servidor, hoje em dia as coisas são bastante diferentes, tendo em vista as novas tecnologias, como os *webservices*. Nesta primeira fase a interface gráfica (ou a linha de comandos) ficava do lado do cliente, enquanto que o servidor tratava de toda a lógica de negócio e do acesso à base de dados. Esta arquitectura era tida como eficaz, mas com o desenvolvimento rápido e quase sufocante da computação dos dias de hoje, a introdução de novas plataformas para desenvolvimento usando programação por objectos, permitiram introduzir uma maior abstracção entre os componentes das aplicações.

Nesta segunda fase separou-se a lógica de negócio da base de dados, deixando essa de estar dependente do SGBD¹ usado. Assim a lógica de negócio pode ser implementada numa linguagem com muito mais flexibilidade do que as linguagens que estão disponíveis nos SGBD's, estas muito mais direccionadas para as consultas e outros tipos de operações sobre a base de dados. O sistema informático (na sua versão anterior), em questão neste projecto, encontrava-se sobre esta arquitectura, existindo três camadas conceptuais correspondendo a interface gráfica do *Visual Basic*, a lógica de negócio sendo o código por trás da interface e a base de dados sendo o SGBD da aplicação.

Para esta fase 3 a arquitectura lógica do sistema foi dividida conceptualmente em 4 partes, modificando um pouco o modelo de 3 camadas *three-tier*. Assim temos, segundo a figura seguinte, o modelo adoptado.

¹ Sistema de Gestão de Bases de Dados



Figura 5-1 Arquitectura lógica

Apesar de fisicamente ser impossível separar as 3 camadas superiores, tendo em conta as tecnologias utilizadas, esta divisão conceptual foi efectuada tendo em vista a interoperabilidade. A nível tecnológico o *Visual Basic* não é uma linguagem orientada a objectos, mas sim a eventos. Tendo em conta esse facto, a separação dos componentes foi efectuada como mostra a figura seguinte. Qualquer módulo criado funcionará como neste diagrama genérico.

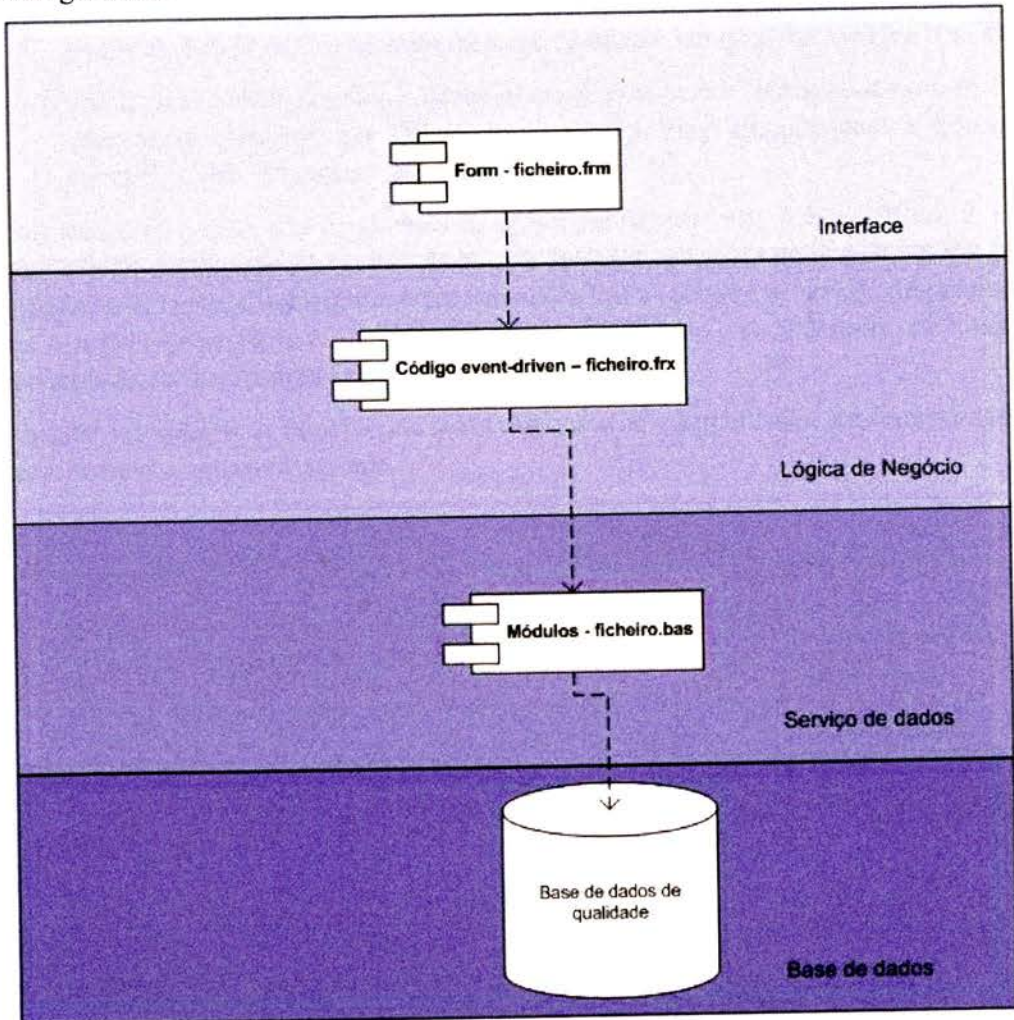


Figura 5-2 Diagrama genérico de Componentes

A base de dados, onde se armazena a informação persistente, existe sob a forma do SGBD SQL Server 2000.

O serviço de dados, que tem como principal objectivo encapsular a estrutura da base de dados (concentrando as consultas à base de dados nesta camada), estabelecendo a ligação com a lógica de negócio. Essa camada existe sobre a forma de ficheiros módulos do *Visual Basic* e o acesso à base de dados utiliza procedimentos (*Stored Procedures*), vistas (*Views*), funções (*Functions*) ou selecção de informação através de *Queries*. Desta forma poderemos manter a informação coerente e centralizar o serviço de acesso a dados.

A lógica de negócio, tem por objectivo processar a informação vinda da camada de interface, que, no caso desta tecnologia, é o controlo dos eventos vindos da interface, aplicando assim as devidas regras de negócio à informação a tratar. Tem também como função tratar a informação vinda da camada de Serviço de dados.

Tendo em conta a tecnologia utilizada esta foi a solução que nos oferecia maior modularidade para futuras migrações da aplicação. Assim, num futuro próximo, quando a aplicação for migrada para a tecnologia .Net (projecto que se encontra em estudo), a informação do acesso ao SGBD já se encontra especificada e separada das regras do negócio.

Temos assim que os objectivos para esta arquitectura são os seguintes:

- permitir alterações em qualquer uma das camadas sem que isso interfira nas outras;
- retirar da lógica de negócio a necessidade de conhecer a estrutura da base de dados, oferecendo métodos, que quando invocados acedem directamente à informação contida na base de dados

De frisar que nem todos os componentes da aplicação seguem esta mesma filosofia, pois o controlo de toda a aplicação não era só da equipa da Indra cpc mas também da equipa interna do banco. Assim, todas as soluções apresentadas pela Indra cpc para as novas componentes do sistema a implementar seguiram esta arquitectura, assim como as alterações efectuadas às componentes existentes da aplicação.

Para ilustrar em detalhe a arquitectura adoptada para as componentes implementadas pelo estagiário temos o seguinte diagrama.

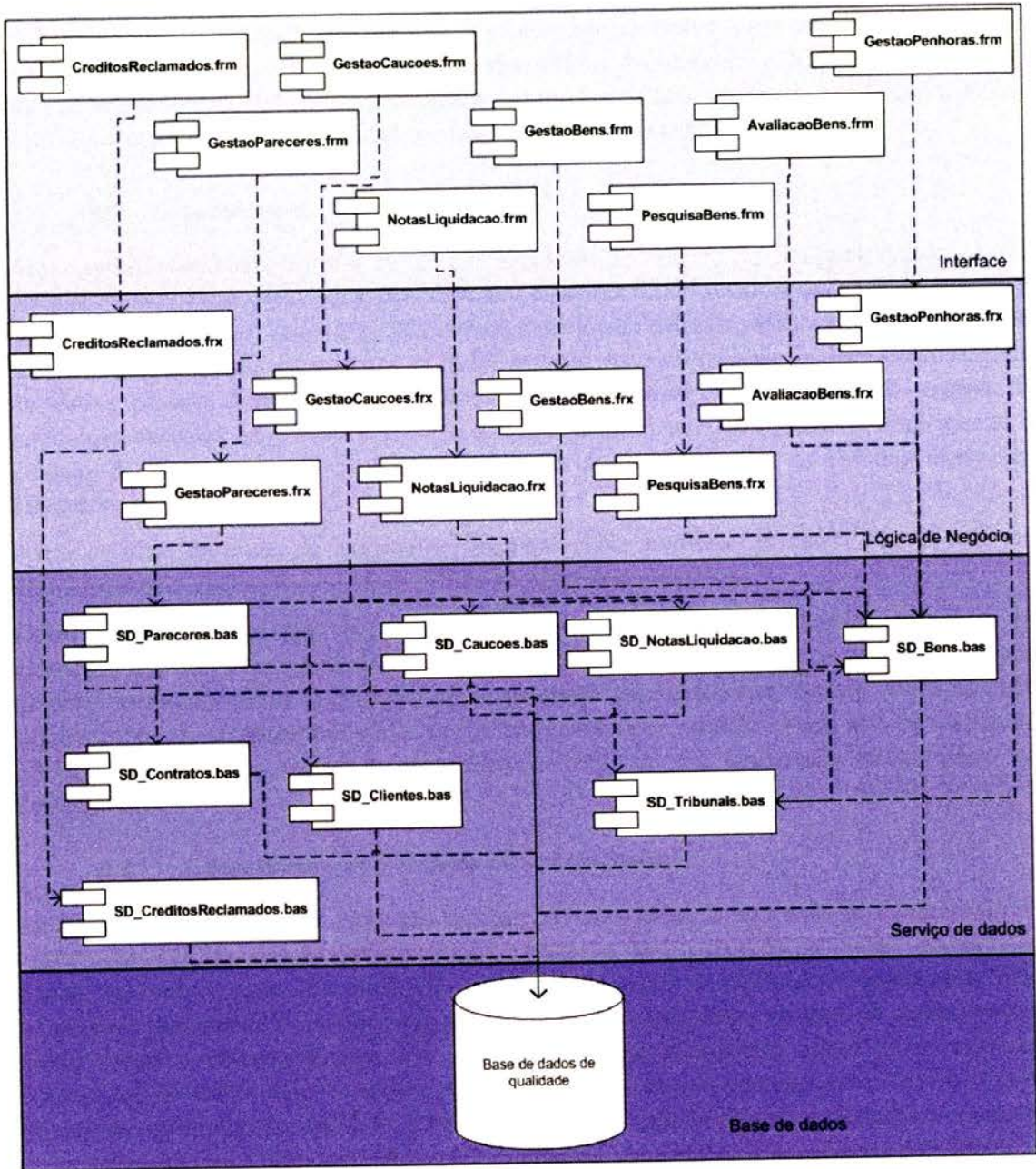


Figura 5-3 Diagrama detalhado de Componentes

5.2 Tecnologias utilizadas

Esta fase do projecto utiliza as mesmas tecnologias que foram utilizadas desde a sua primeira fase. As opções a nível tecnológico não eram da responsabilidade do grupo de trabalho da Indra cpc. Por estas razões não é efectuada neste capítulo uma exaustiva introdução teórica às tecnologias.

No entanto, fazendo um pequeno resumo, o sistema informático em questão é uma aplicação implementada em Microsoft *Visual Basic* v6.0 com ligação ao Sistema de Gestão de Bases de Dados SQL Server 2000. Para implementar as componentes de geração de relatórios (denominados mapas) foi utilizada a ferramenta *Crystal Reports* 8.5.

Existiram também componentes em que eram gerados documentos pré formatados, baseados em *templates*. Para implementar essas funcionalidades foi utilizado o Microsoft Word 2000 com acesso a *bookmarks*. Foi também utilizada uma ferramenta interna da instituição bancária com o objectivo de definir transacções com a base de dados central

5.3 Dependências

Assim, nesta fase 3 do projecto, que já tem seis anos de vida, existe um pressuposto técnico do qual tínhamos que partir: o modelo de dados estava definido assim como a tecnologia. Para as novas componentes foram definidos novos modelos de dados e para as alterações à versão anterior foram alterados os modelos de dados antigos. As decisões sobre o desenho do modelo de dados passava sempre pela apreciação do responsável pelo projecto da equipa da instituição bancária. Esse aspecto levou a que muitas vezes não fosse possível implementar o modelo de dados como o estagiário o desenhava mas sim com as devidas alterações efectuadas pelo cliente.

Ainda na fase de testes foi necessário trabalhar sobre modelos de dados que já estavam definidos e que não era da responsabilidade do estagiário alterá-los.

Outro pressuposto era que, como a metodologia do modelo evolucionário o indica, os requisitos não são bem conhecidos pelo utilizador e isso fez com que o ciclo fosse iterado diversas vezes de forma a acompanhar a evolução dos requisitos. Assim, foi necessário implementar as componentes de uma forma genérica e modular para que se existirem alterações de alguma gravidade, estas não se reflectam em demasiado tempo gasto no desenvolvimento.

5.3.1 Transacções com o sistema central do Banco

Existem diversos dados que estão não na base de dados local da aplicação de recuperação de crédito mas sim na base de dados central do banco, a qual é acedida por todas sucursais do banco, em todo o país. Por exemplo, num banco, quando é efectuada uma consulta pelo empregado que atende o público, este acede e mantém esses dados na base de dados central. Desta forma é necessário que haja sincronização da informação assim como executar transacções de dados com o central. A aplicação já efectuava comunicações com o central através de um protocolo, definido ainda nas primeiras fases do projecto. Nesta fase competia ao estagiário definir transacções utilizando esse protocolo e um *software* interno do cliente.

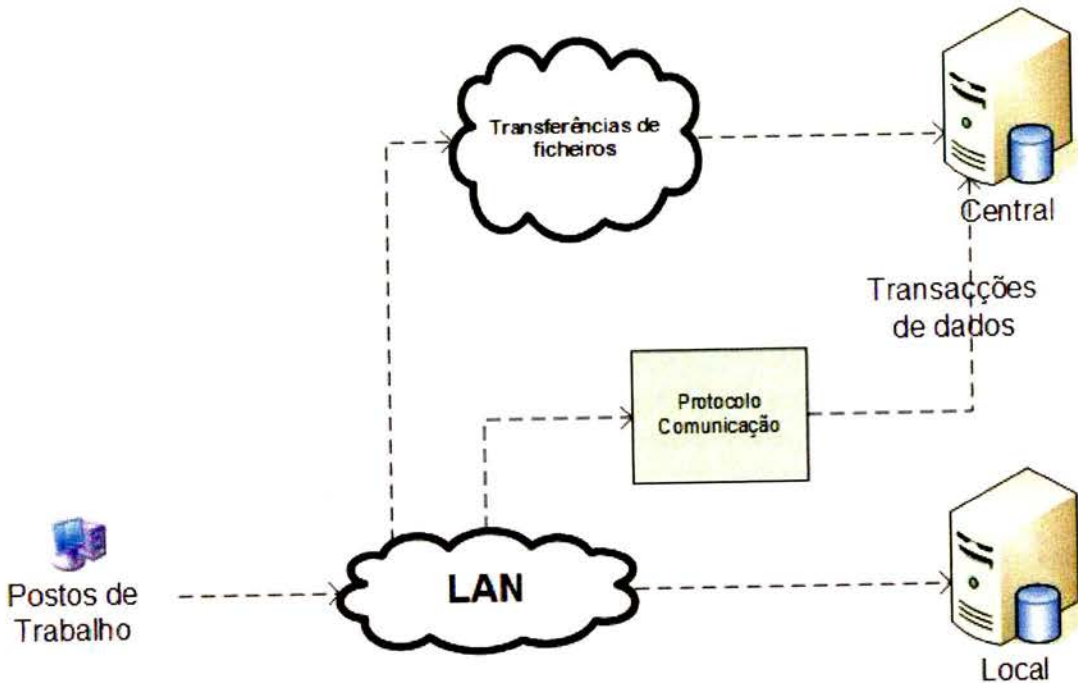


Figura 5-4 Diagrama Informal do sistema

Este diagrama representa informalmente a arquitectura do sistema. Desta forma a aplicação comunica através de transacções com os dados centrais e assim não compromete a coerência dos dados fornecidos ao utilizador.

5.4 Modelo de dados

Para modelação dos dados foram utilizados modelos entidade relação. Estes são representações lógicas dos dados. Consistem em entidades, relações entre as entidades e os atributos das entidades. Uma entidade pode ser definida como um objecto, um conceito do qual se pretende fazer a manutenção de dados. Os atributos são certas propriedades desses conceitos que se desejam manter. As relações são associações que existem entre uma ou mais instâncias das entidades. Existe ainda outro conceito importante nestes diagramas que é o conceito de *cardinality* da relação, que representa, por exemplo entre a entidade A e a entidade B, o número de instâncias da entidade B que poderão ser associadas com cada instância da entidade A. Temos então que os principais objectivos para a modelação de dados utilizando o modelo entidade relação são:

- Fornecer um elevado grau de independência dos dados.
- Proporcionar uma definição comum dos dados de simplicidade espartana, de forma que uma vasta gama de utilizadores numa empresa possa com ela lidar.
- Simplificar o potencialmente gigantesco trabalho do administrador da Base de dados.
- Introduzir uma fundamentação teórica na gestão de Bases de dados.

- Elevar a programação de aplicações de bases de dados para um novo nível, em que conjuntos sejam tratados como operandos, em vez de serem tratados elemento a elemento.

Associado a esta metodologia de modelação de dados temos o conceito de normalização de dados.

Existem três formas normais que indicam o nível de normalização das relações entre as entidades. Assim uma relação está na 1ª forma normal (doravante no presente relatório, identificada como FN) se, e somente se, todos os seus atributos contêm apenas valores atômicos (simples, indivisíveis). Para uma relação estar na 2ª FN necessita em primeiro lugar de estar na 1ª FN e que cada atributo não chave dependa funcionalmente da chave inteira e não só de uma parte da chave composta. A 3ª FN diz que é necessário a relação estar na 2ª FN e não existir nenhum atributo não chave dependendo funcionalmente de outro atributo não chave.

Partindo destes pressupostos teóricos, sempre tendo em conta os requisitos de performance do sistema e por isso tentando sempre arranjar um equilíbrio entre os diversos requisitos, foram modelados os dados para as componentes implementadas.

5.4.1 Desnormalizações

Foram efectuadas em diversos casos “desnormalizações” do modelo de dados de forma a obter melhores resultados de performance. As desnormalizações só fazem sentido nesse contexto e deve-se sempre primeiro normalizar e só depois desnormalizar se for realmente necessário. Focando alguns exemplos de desnormalizações:

- Duplicação de colunas não chave

Este é o caso mais básico de desnormalização. Poderemos ter a necessidade de duplicar colunas não chave para reduzirmos o acesso a menos uma tabela. Normalmente faz-se isto quando temos uma tabela do tipo *lookup*, ou seja que só tem código e descrição, e isso obriga a aceder a mais uma tabela para irmos buscar a descrição. É necessário, nestes casos, ter em conta a sincronização da informação, para não termos valores diferentes da tabela de referencias.

- Separação horizontal de tabelas

Quando temos tabelas que já se encontram com um número elevado de registos podemos solucionar o problema de performance dividindo essa tabela em diversas tabelas com a mesma estrutura, mas que guardam diferentes gamas de valores da chave primária. Isto reduz o tamanho e a densidade dos índices usados para retornar os dados, melhorando assim a performance. Por outro lado é necessário ter em conta que se vai ter que aceder a dados em várias tabelas em vez de uma só, e é necessário usar um procedimento lógico mais complexo para manter as tabelas.

- Separação Vertical de tabelas

Quando tabelas com muitas colunas são sujeitas a uma grande actividade de escrita e leitura, é possível separar essa estrutura em diversas tabelas com diferentes conjuntos de colunas. Cada uma dessas tabelas usa a mesma chave primária. Como no caso anterior é necessário ter cuidado com a manutenção das tabelas.

- Campos compostos

Outro caso interessante de desnormalização são os campos compostos por outros campos. Imaginemos que temos o campo primeiro nome e o campo último nome. Se concatenarmos os 2 campos num terceiro e quisermos ordenar por nome completo, será muito mais rápido ordenar só por esse terceiro do que pelo primeiro e segundo campos (isto com os devidos índices).

5.5 Problemas e Soluções

Dado aos pressupostos já explicados (o facto de a equipa de desenvolvimento ter trabalhado sobre uma tecnologia que não é orientada a objectos mas sim a eventos, e de o controlo sobre o modelo de dados não ser inteiramente da equipa da Indra cpc) os problemas que surgiram foram variados:

5.5.1 Migração de dados

Numa projecto em que se faz *upgrade* de um sistema informático existem dois ambientes (se assim se podem chamar) da aplicação: o ambiente de produção, ou seja a base de dados que está em produção neste momento, e o ambiente de qualidade, ou seja a base de dados que está a ser usada para o desenvolvimento (neste caso da fase 3) do projecto.

Quando se iniciou este projecto uma cópia da estrutura e dos dados da base de dados de produção foi colocada numa outra máquina a qual se passou a identificar como a base de dados de qualidade. A figura seguinte mostra essa primeira migração de dados e estrutura.

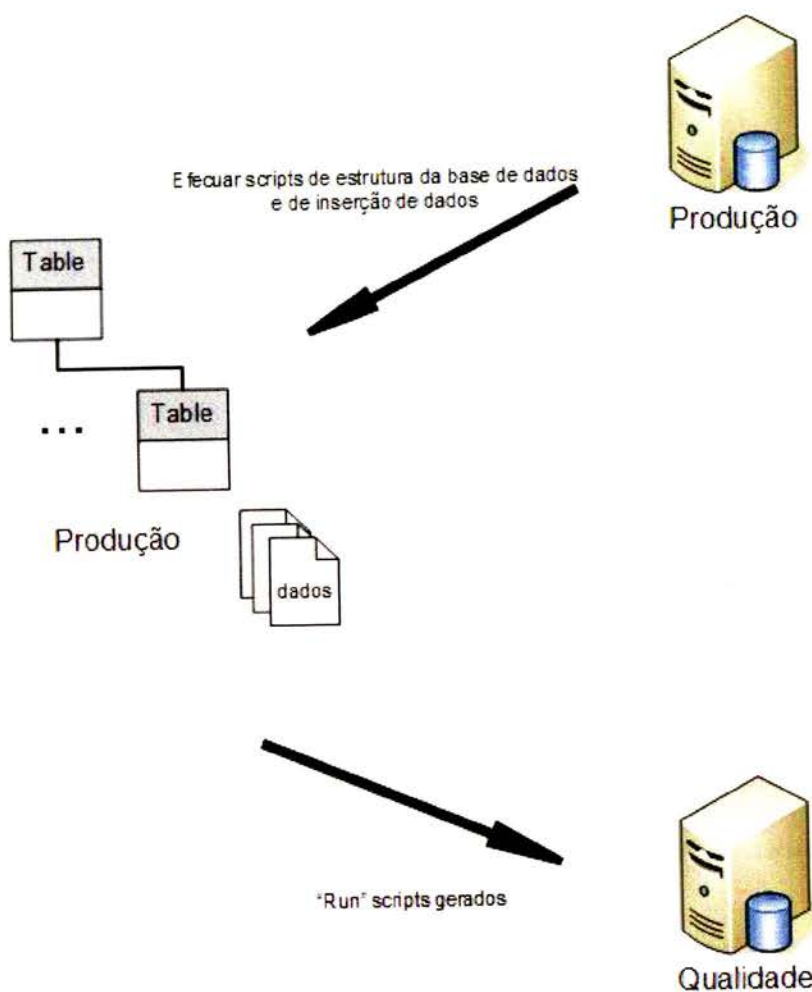


Figura 5-5 Diagrama informal da migração de dados no início do projecto

Com o desenvolvimento das componentes novas e com as alterações às componentes antigas existiram obviamente inúmeras entidades e relações que foram alteradas e criadas no modelo de dados de qualidade. Numa passagem do software para o ambiente de produção é necessário garantir que não se perdem os dados que existiam em produção e que a estrutura de qualidade é criada em produção.

Esta migração de dados era da responsabilidade da equipa da instituição bancária, mas como a nossa equipa interagiu directamente com esta, é necessário explicar como se procedeu à migração dos dados, os erros que poderiam ter sido evitados e as melhorias que se poderiam efectuar.

Assim, antes dos testes dos utilizadores finais começarem foi efectuado um teste de migração de dados para um ambiente de pré-produção. Ou seja foi feita uma simulação do que seria passar a aplicação para produção. Para termos valores indicativos da dimensão do problema, poderemos indicar que a base de dados ocupa cerca de 8 GB, em cerca de 500 tabelas, 2000 procedimentos, 300 vistas e 50 funções. Esta estrutura toda era necessária migrar de uma forma coesa e eficiente, para no máximo em 2 dias (correspondente ao tempo de paragem do fim de semana) a migração estar 100 % efectuada. A figura seguinte mostra sequência lógica de procedimentos que fazem parte da migração de dados.

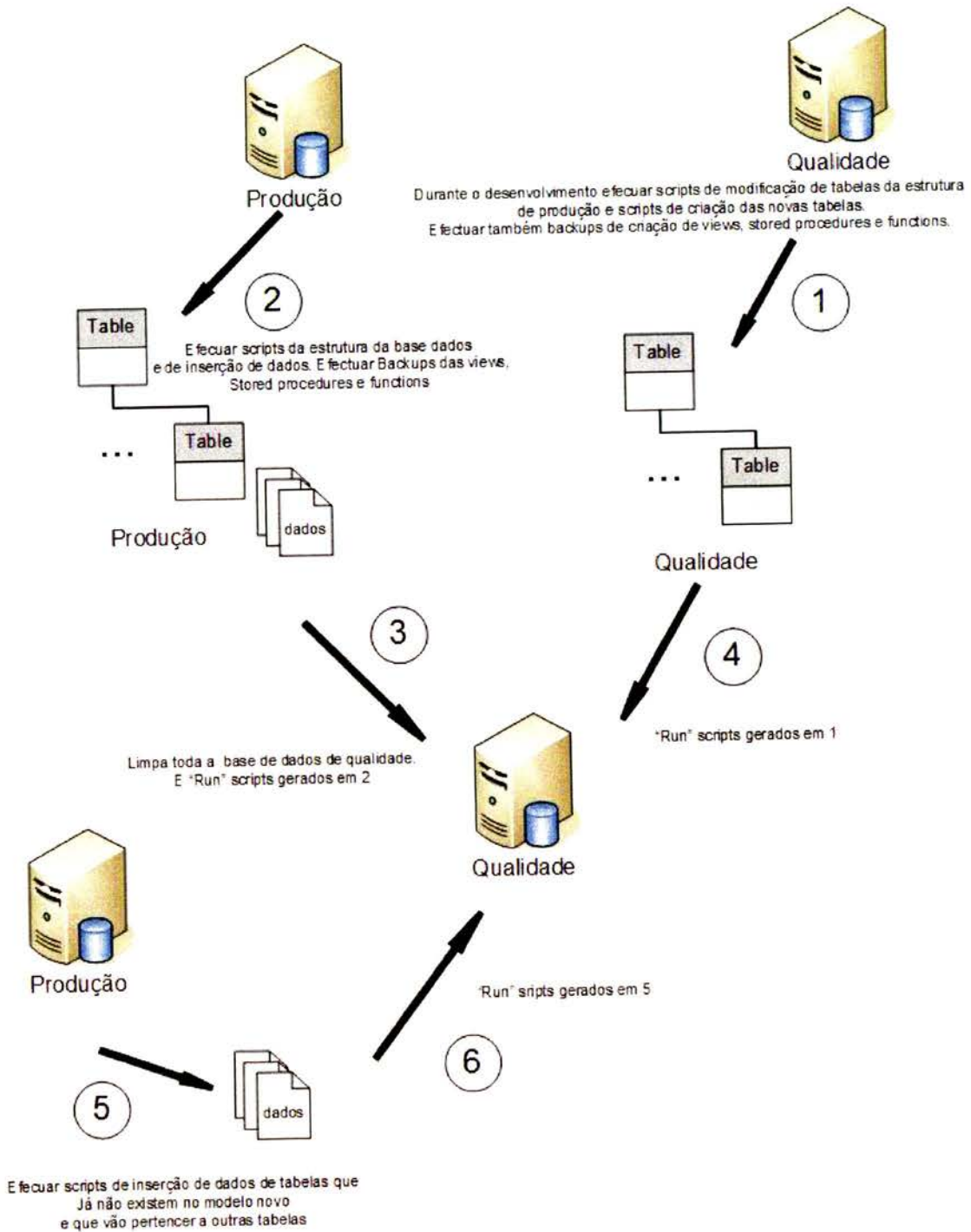


Figura 5-6 Diagrama informal da migração de dados de simulação final

Esta simulação de entrada do modelo de dados em produção tem uma metodologia que pretende alcançar todos os objectivos descritos anteriormente. E no fundo a ideia de passar os dados a produção desta forma é correcta. O problema é quando não se tem o controlo completo sobre o modelo de dados e o passo número 4 e 6 ocorrem com deficiências. O que acontece nesse caso é que a máquina que tinha a base de dados de desenvolvimento, sobre a qual as duas equipas estavam a desenvolver, fica com uma estrutura e dados com deficiências

devido aos erros ocorridos. Foi um dos erros que aconteceu durante este projecto, demorando este procedimento um tempo inexplicavelmente longo.

A modificação que proponho neste caso é que se tivesse criada outra base de dados numa outra máquina, denominada pré-produção, na qual se faria este teste para ver se a passagem a produção poderia correr com normalidade. Nunca fazer este procedimento sobre a base de dados de qualidade pensando que tudo poderia correr sem erros. Desta forma as equipas de desenvolvimento tiveram que estagnar, perdendo também algum do trabalho efectuado anteriormente. Mas como já foi dito este processo era da responsabilidade da equipa do cliente. A única responsabilidade que tinha a nossa equipa era que ocorressem os passos 1 e 4 eficientemente, procedimento que na realidade ocorreu sem quaisquer problemas.

5.5.2 Performance nas consultas

Existia, como já foi referido um requisito de performance, que tinha como valor indicativo o tempo em segundos de consultas à base de dados. Esse valor era de 2 segundos e assim surgiram diversos casos em que depois de realizadas as consultas era necessário alterar as próprias consultas, os índices das tabelas, etc. Assim era necessário conhecer bem os tipos de índices e as inúmeras potencialidades do SGBD de forma a resolver este problema que surgiu e acompanhou-nos ao longo do desenvolvimento do sistema.

De seguida são abordados dois tópicos que fizeram parte desse estudo e que em muito melhoraram certas consultas mais complexas: índices e vistas indexadas.

O SGBD usado, como já foi referido, foi o Microsoft SQL Server 2000 e este suporta dois tipos de índices:

- Índices *Clustered*

Estes tipos de índices definem fisicamente a ordenação das linhas da tabela. Por esta razão cada tabela só pode ter um único índice *clustered*. De qualquer forma esse índice pode ser composto por múltiplas colunas, denominado índice composto. Quando se cria uma tabela e se define uma chave primária, por defeito é criado um índice deste tipo nessa chave.

São extremamente úteis em colunas onde são efectuadas pesquisas onde é definido um filtro que seja uma gama de valores. O que acontece é que, se a tabela está ordenada desta forma, temos a garantia que as linhas adjacentes, à primeira encontrada, são importantes e não interessa pesquisar fora desse alcance. Se imaginarmos o caso em que é pesquisado numa gama entre duas datas. O que acontece é que só é necessário encontrar a primeira ocorrência da data menor e a última ocorrência da data maior. As linhas que se encontram entre essas duas linhas encontradas fazem parte garantidamente do conjunto de dados que queríamos seleccionar. Assim a performance pode melhorar bastante para este tipo de consultas.

Tendo estes conceitos entendidos poderemos resumir isto em algumas *best-practices* de SQL que foram tomadas em conta durante o desenvolvimento e testes de performance das consultas efectuadas.

Deve ser criado índices clustered nos seguintes casos:

- Colunas que contem um número elevado de valores distintos.
- Consultas que retornem uma gama de valores usando operadores como BETWEEN, >, >=, < e <=.
- Colunas que sejam acedidas sequencialmente

- Consultas que retornem grandes conjuntos de dados
- Colunas que são frequentemente acedidas em cláusulas GROUP BY

E poderemos considerar que este tipo de índices não serão uma boa escolha nos seguintes casos:

- Colunas que poderão estar sempre a ser alteradas. Isto é uma importante consideração pois se o índice guarda a tabela fisicamente ordenada. Se para pesquisar é mais rápido, para inserir e modificar valores a performance baixa muito. Se tivermos casos em que a informação tem tendência a ser volátil e existe um volume de transacções de dados elevado, então não será uma boa escolha este tipo de índices.

O outro tipo de índices que é suportado é denominado:

- Índices *NonClustered*

Um índice deste tipo é análogo ao índice de um livro. Os dados são guardados num sítio e o índice noutra local, com apontadores para a localização física dos dados. Os itens no índice são guardados pela ordem dos valores chave do índice, mas a informação na tabela é guardada numa ordem diferente, que por sua vez pode ser indicada por um índice *clustered*.

Estes índices são uma óptima escolha em situações de pesquisa exacta, ou seja o SGBD pesquisa no índice a localização exacta de um determinado valor e depois retorna os dados directamente dessa localização na tabela. Tal como fazemos num índice de um livro.

Como os livros que poderão ter mais do que um índice, poderão ser criados múltiplos índices do tipo *NonClustered*. São mais apropriados de usar nas seguintes situações:

- Colunas que contem um número elevado de valores distintos. Se existirem colunas com poucos valores distintos, como 0 ou 1, geralmente um *table scan* é mais eficiente e o índice não será usado.
- Consultas que não retornem grandes volumes de dados
- Colunas frequentemente envolvidas em cláusulas de pesquisa exactas.
- Colunas que são frequentemente usadas em *joins* de tabelas.

Para além dos índice foi ainda analisada outra funcionalidade nova no SQL Server 2000 denominada Vistas indexadas.

A titulo introdutório, o SQL Server 2000 suporta uma funcionalidade, denominada Vistas, que consiste em criar tabelas virtuais de uma determinada visão que se quer ter de entidades e relações da base de dados. As vistas fornecem um mecanismo útil para restringir os dados a certos subconjuntos de dados e permitir aos utilizadores visualizar aspectos lógicos dos dados. Normalmente contêm diversas junções entre tabelas e complexas agregações ou retornar grandes volumes de dados, portanto, sem a ajuda de um índice, as vistas sofrem frequentemente de um desempenho pobre.

No SQL Server 2000, temos a possibilidade de melhorar essa performance pois existe uma nova funcionalidade na área das vistas, que são as vistas indexadas.

Uma vista indexada é então qualquer vista que tenha um índice único do tipo *cluster*. Quando se cria um índice dentro de uma vista o resultado é materializado e guardado fisicamente na base de dados com a mesma estrutura que uma tabela quando tem um índice. Esta

materialização é feita sempre em *runtime* quando a vista não tem o índice, logo os custos são menores e a performance aumenta numa vista indexada. Para além disto certas consultas e vistas, que poderão não aceder directamente à vista indexada, poderão beneficiar desta tabela da vista indexada sem ter que se efectuar nenhuma alteração às mesmas. Os dados que são modificados nas tabelas que essa vista utiliza, são alterados também na outra tabela (que foi o resultado da materialização da vista indexada), da mesma forma que aconteceria numa tabela que tenha um índice. Foi sem dúvida um solução que muitos benefícios trouxe.

5.5.3 Desnormalizações com resultados negativos

Um tipo de desnormalização já referido é a duplicação de colunas não chave. Neste modelo de dados já existiam muitas tabelas em que era utilizado uma metodologia para os casos em que existem tabelas de referência e queremos obter a descrição da referência a partir de um código.

Assim, passando a explicar com um exemplo prático, para este tipo de tabelas estar normalizado deverá efectuar-se a modelação de dados da figura seguinte.

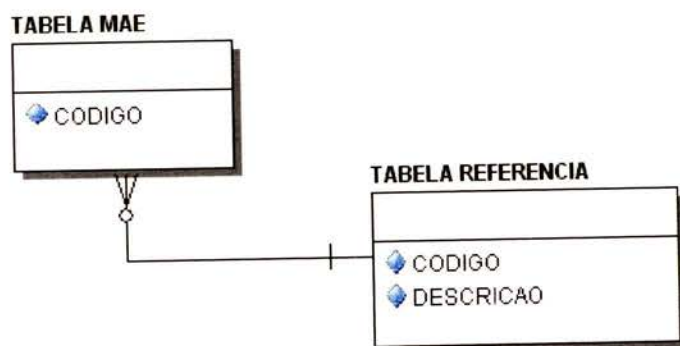


Figura 5-7 Modelo Entidade Relação de tabelas referência

O que acontece é que, muitas vezes, em diversas tabelas existem vários campos que acedem às tabelas de referência e por vezes quer-se reduzir os custos desses *joins* que vão buscar a informação da descrição.

Para se poupar esses custos o modelo de dados foi desnormalizado da seguinte forma: passou a existir uma só tabela com todas as referências e em cada tabela que necessitava de referências existia o campo código e o campo descrição, este ultimo sendo uma função que ia buscar a descrição baseado no código desse registo. Essa tabela de múltiplas referências tem o código da referência, o código definido no âmbito dessa referência e a sua descrição. Clarificando temos a figura seguinte:



Figura 5-8 Modelo entidade relação de tabelas de referência desnormalizado

Neste modelo apesar de não necessitarmos de fazer um *join* para obtermos o valor da descrição do código do registo, temos uma função no próprio campo de descrição o que no fundo não vai ajudar a termos uma execução melhorada da consulta. O que se ganha em ter todas as referências na mesma tabela é que se torna de manutenção mais fácil. Mas, no fundo, o facto de as referências estarem todas na mesma tabela tornará mais lento o acesso a esses dados.

O que se propõe é desnormalizar no verdadeiro sentido da palavra. Ou seja não termos o código na tabela mãe, mas sim só a descrição. Assim garantidamente não temos nunca que aceder à tabela de referências quando estamos a ler dados da tabela mãe. Em relação às tabelas de referência considero uma boa opção para este tipo de desnormalização que proponho, devido, em primeiro lugar, ao facto de serem de manutenção facilitada. Em segundo lugar porque, como nunca temos que aceder a esta tabela de referências para visualizarmos os dados dos registos das tabelas mãe, não existe um problema de performance pois todos os *joins* são evitados.

Assim, só teremos que manter um lógica de procedimentos para garantirmos que os dados das descrições nas tabelas mãe estão sempre concordantes com as tabelas de referência. Embora possa ir contra toda a teoria do que foi ensinado ao longo do nosso curso, poderá ser uma boa política pouparmos esforços adicionais desta forma rudimentar.



Figura 5-9 Proposta para a modelação das tabelas de referência

6 Conclusões e perspectivas de trabalho futuro

O estágio completado tinha como principal objectivo implementar a solução pretendida pelo cliente e, em traços gerais, poder-se-á dizer que o estágio foi concluído com sucesso. Como foi descrito o projecto foi implementar um *upgrade* de uma versão antiga do sistema. Posto isto, e mesmo com as limitações tecnológicas existentes, foi pensada uma arquitectura que foi de encontro ao conhecimentos adquiridos ao longo da Licenciatura em Informática e Computação na área de Engenharia de *Software*.

O facto de o projecto ser extremamente importante para a empresa cliente e de os seus resultados influenciarem directamente a evolução da recuperação de crédito da instituição bancária cliente foi extremamente aliciante para o desenvolvimento do projecto. Assim, o factor “cliente” trouxe novos desafios e, sem dúvida, uma perspectiva diferente do mundo de trabalho.

O facto de estar em contacto directo com o mundo real, e não estar a produzir um protótipo que poderia ficar na “prateleira”, teve os seus pontos fortes e pontos fracos. Se, por um lado, não foi um projecto de investigação, e por isso não marcar tão significativamente a nível de desenvolvimento enquanto engenheiro de software inovador, por outro lado, o facto de estar em contacto com as metodologias, o negócio e toda uma infra estrutura real e que engloba diversas vertentes, obrigou-me a rever toda uma teoria ensinada ao longo de anos e repensar todo um estatuto, do que na realidade, é ser um engenheiro informático. Ser confrontado com realidades de desenvolvimento de software tão distintas das que a faculdade nos apresenta fez-me elevar a um nível bastante alto o meu sentido crítico e pensar sempre em melhorar tudo o que me era apresentado. Esse sentido crítico que nos ajudam a desenvolver na faculdade é sem dúvida um dos grandes valores que levo deste estágio. Numa empresa como a Indra cpc não tive qualquer problema em relacionar-me com os meus colegas de trabalho e de sempre expor as minhas ideias e de ser apoiado e/ou criticado noutras. O facto de ter estado a trabalhar directamente com o cliente, levou a que muitas vezes essas ideias não pudessem ser postas em prática por vontade do mesmo. O que critico nesta conclusão é que muitas vezes essas decisões não tinham por base fundamentos teóricos, simplesmente porque existe uma forma de trabalhar que está entranhada de tal forma que se torna muito complicado um recém chegado do mundo académico modificar métodos de trabalho que já têm anos. Apesar do apoio da minha gestora de projecto nessas ideias compreendi que o poder da decisão do cliente é muito mais forte, independentemente de ter ou não ter razão para tais decisões. E isto tudo porque, infelizmente, o dinheiro fala mais alto e é necessário compreender que a empresa não se pode dar ao luxo de perder clientes só porque não se concorda com a sua maneira de trabalhar, ainda por cima estando a trabalhar em colaboração tão directa como neste estágio.

De qualquer forma, como me apercebi, aprende-se muito com estas experiências. E que, embora chegue a um cargo de gestor de projecto e esteja junto do cliente, terá que existir sempre um equilíbrio entre o que o cliente pretende e o que a empresa pretende enquanto *software house* de qualidade. Não vou, por isso, deixar de criticar e de propor novos conceitos, novas metodologias e formas diferentes de pensar. Só desta forma poderemos evoluir. Melhorando cada vez mais o que fizemos ontem.

Concluindo, apesar de hoje em dia o *Visual Basic* ser uma linguagem ultrapassada e com conceitos que deixam muito a desejar quando se pensa numa arquitectura modular, o estágio

foi extremamente abundante em experiências, não só do ponto de vista técnico, debruçando-se sobre várias tecnologias, mas também humano (pelo contacto estabelecido com equipas internas da instituição bancária e colegas na empresa), abordando uma série de aspectos extremamente actuais na área da Banca, dando uma perspectiva bastante abrangente do negócio

Referências e Bibliografia

Estratégia para a elaboração de uma tese, António Dias de Figueiredo,

<http://eden.dei.uc.pt/~ctp/teses.htm>

Introduction to Visual Basic 6,

http://www.free2code.net/tutorials/programming/vb/14/vb6_intro.php

Normalization Is a Nice Theory, Davida Adams e Dan Beckett

<http://www.island-data.com/downloads/papers/normalization.html>

SQL Server Books online, Microsoft, 1998-2000

MSDN Library Visual Studio 6.0, Microsoft, 1998

Microsoft Visual Basic Developer Center, Microsoft,

<http://msdn.microsoft.com/vbasic/>

Professional SQL Server DTS (Data Transformation Server), Mark Chaffin, Brian Knight e
Tood Robibson

“Indra cpc”, Sitio Web Institucional, <http://www.indracpc.pt>

ANEXO A: Tecnologias utilizadas

Visual basic 6

O *Visual Basic* (denominado VB) é uma linguagem de programação concebida a pensar nos sistemas operativos Windows. Dadas as características destes sistemas, nomeadamente no que se refere às suas potencialidades gráficas, as aplicações que funcionam nestes sistemas são estruturalmente diferentes das desenvolvidas "tradicionalmente" (por exemplo, para DOS).

Ao contrário das aplicações "procedimentais" (por exemplo, desenvolvidas usando *Pascal* tradicional), em que o fluxo da aplicação segue por um caminho definido, executando procedimentos de acordo com as necessidades, as aplicações orientadas a eventos (aplicações do *Visual Basic*), executam porções de código de acordo com eventos produzidos pelo utilizador ou pelo sistema.

Consequentemente, a ordem de execução do código depende dos eventos produzidos, que, por sua vez, dependem das acções do utilizador e/ou eventos do sistema.

Como as acções do utilizador não são previsíveis, o código deverá considerar as diversas opções que se põem ao utilizador face a um dado *interface*. Quando não o fizer, deverá pelo menos indisponibilizar as operações susceptíveis de colidirem com a normal execução do programa. Por exemplo, não permitir que o utilizador entre numa determinada opção do menu ou pressione determinado botão, quando a execução dessas tarefas não faz sentido ou provoca erro.

Uma aplicação com alguma complexidade em *Visual Basic* utiliza várias janelas interdependentes. Em consequência, existem implicações ao nível do desenvolvimento da estrutura da aplicação e posterior controlo da execução da mesma.

A construção do *interface* de uma aplicação *Visual Basic* consiste em incorporar os controlos gráficos disponibilizados e configurá-los de acordo com as nossas necessidades. Logo, a forma de construção do *interface* com o utilizador é facilitada aumentando a qualidade e produtividade no desenvolvimento da aplicação.

Para além das facilidades de desenho, as aplicações *Visual Basic* apresentam outras vantagens, como sejam:

- utilizam os recursos do ambiente (impressoras, tipos de letra, etc.);
- incluem potencialidades de interligação a SGBDs (Sistemas Gestão de Bases de Dados) permitindo o acesso a dados em formatos específicos
- permitem a troca de dados entre aplicações.

Quando se termina uma aplicação implementada no *Visual Studio* esta será um ficheiro *exe* que usa a *Visual Basic Virtual Machine*.

Sql Server 2000

O *SQL Server 2000*, é um produto *Microsoft*, que engloba um conjunto de componentes que tem o objectivo de, quando utilizados em conjunto, irem de encontro às necessidades de armazenamento e análise de dados de sistemas que processem quantidades enormes de dados.

Performance, escalabilidade e confiabilidade são factores essenciais, e o prazo de desenvolvimento de produtos baseados em bases de dados é crítico. Além de possuir essas

qualidades básicas, necessárias para as actuais corporações, o *SQL Server 2000* proporciona agilidade na gestão e análise de seus dados, permitindo que as empresas se adaptem de forma rápida e competente a novos ambientes em permanente transformação, conferindo-lhes uma sensível vantagem competitiva perante o seu mercado. Do ponto de vista da gestão de dados e análise, transformar dados brutos em inteligência corporativa e tirar o máximo proveito dos recursos da Internet tornaram-se factores críticos para o sucesso dos negócios. Como solução completa de armazenamento e análise de dados, o *SQL Server 2000* abre um novo leque de possibilidades para o desenvolvimento rápido e ágil de uma nova geração de aplicações de negócios que poderão colocar a sua empresa em posição de vanguarda na vantagem competitiva. O *SQL Server 2000*, ganhador de alguns dos maiores destaques mundiais em *benchmarks* de velocidade e escalabilidade é um produto de banco de dados totalmente desenhado para a Web, com total suporte à linguagem XML (*Extensible Markup Language*).

O *SQL Server 2000* traz amplos recursos de programação de bases de dados baseados em padrões da Web. Suporta de forma nativa a linguagem XML e a Internet possibilitando a fácil armazenagem e recuperação de dados em formato XML com *stored procedures* incorporadas no produto. O usuário ou programador pode recorrer aos *updategrams* (diagramas de actualização) para inserir, actualizar ou apagar dados de forma muito simples.

Com o *SQL Server 2000* pode-se atingir gratificantes níveis de escalabilidade e confiabilidade. Os seus recursos de *scale-up* e *scale-out* respondem aos mais exigentes requisitos do ebusiness e desenvolvimento de aplicações corporativas.

- ***Scale-up***

O *SQL Server 2000* utiliza todo o potencial dos sistemas de multiprocessamento simétrico (SMP). O *SQL Server Enterprise Edition* pode utilizar até 32 processadores e 64 GB de RAM.

- ***Scale-out***

O recurso de *scale-out* distribui a base de dados e a carga dos dados através dos diferentes servidores.

- **Disponibilidade**

O *SQL Server 2000* adquire sua disponibilidade máxima através de um avançado *clustering* à prova de falhas, distribuição do log (*log shipping*) e através de novas estratégias de *back-up*.

O *SQL Server 2000* inclui todas as ferramentas para acelerar o desenvolvimento, desde a fase conceptual até a distribuição final.

- **Serviços de análise amplos e integrados**

Com o *SQL Server 2000*, o usuário poderá desenvolver soluções de análise de grande envergadura empresarial, com ferramentas integradas que conferem valor efetivo aos dados. Com isso, pode-se criar novos processos empresariais que se baseiem nos resultados da análise de dados e reunir, com grande flexibilidade, conjuntos personalizados de resultados, a partir dos cálculos mais complexos.

- **Alta velocidade no desenvolvimento, depuração e transformação de dados**

O SQL Server 2000 traz sofisticados recursos para filtrar ou depurar consultas de forma interactiva, mover e transformar, rapidamente, dados a partir de qualquer fonte, além de definir e utilizar funções como se elas estivessem incorporadas ao *Transact-SQL*. O programador poderá construir e codificar aplicações da base de dados a partir de qualquer ferramenta do *Visual Studio*.

- **Gestão e tuning da base de dados simplificados**

Com o *SQL Server*, pode-se gerir uma base de dados de forma centralizada através de todos os recursos de uma organização. Mover e copiar bases de dados entre computadores ou através de diferentes actividades corporativas pode ser realizado on-line.

Agregado à base de dados pode-se ainda ter colecções de instruções SQL, pré compiladas denominadas *Stored Procedures*. Processados como uma só unidade podem ser chamados pela aplicação com uma só chamada e permite parâmetros definidos pelos utilizadores. Permite também controlo de fluxo e execução condicional.

DTS (Data Transformation Services)

A sigla DTS significa *Data Transformation Services*, e é uma ferramenta que pertence ao conjunto de ferramentas disponibilizadas pelo *SQL Server 2000*. Esta foi outra ferramenta usado no decorrer do projecto, pois existia um elevado número de dados em formatos diferentes de pequenas aplicações que os utilizadores usavam como folhas de cálculo *Excel* e bases de dados *Access*. Assim, estes dados foram migrados de uma forma eficiente, prática e segura.

O objectivo desta ferramenta é disponibilizar uma forma simples de transferir dados entre uma variedade enorme de fontes e destinos de uma forma visual. Programa-se assim em termos de objectos virtuais e fluxos de dados desenhados no ecrã. Assim como transformar os dados de acordo com os requerimentos do que se pretende fazer. DTS podem ser usados em aplicações *data-driven* fornecendo a transferência dinâmica dos dados entre aplicações na Internet e Intranet.

Crystal reports 8.5

A ferramenta *Crystal Reports*, versão 8.5 foi desenhada com o propósito de se trabalhar com a base de dados de forma a ajudar a analisar e interpretar informação importante contida na mesma. *Crystal Reports* cria relatórios e contém ferramentas para se criar relatórios de grau de complexidade elevado.

Este software foi desenhado para se produzir relatórios de qualquer fonte. Incluindo formulas, sub relatórios, e formatação condicional ajuda a dar sentido aos dados e descobrir importantes relações que poderiam estar escondidas até então. Gráficos e mapas colocam a informação visualmente relevante quando, por vezes, palavras e números não são suficientes.

Mas a flexibilidade dos *Crystal Reports* não termina só na criação de relatórios, estes podem ser transformados e publicados numa variedade de formatos, incluindo *Microsoft Word* e *Excel*, email e até em sites. A incorporação entre a aplicação em *Visual Basic* e o *Crystal Reports* pode ser efectuado de uma forma simples e eficaz através do *Crystal Reports ActiveX Control*.





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000081524

EN