



**Universidade  
Atlântica** Oeiras

## **GESTÃO DE SISTEMAS E COMPUTAÇÃO**

### **B**ASE DE **D**ADOS: DESENVOLVIMENTO E IMPLEMENTAÇÃO PARA GESTÃO E DISTRIBUIÇÃO DE ASSINATURAS

*Trabalho Final para Laboratório de GSC Aplicado*

Elaborado por: Filipe Ricardo da Costa Almeida

Aluno N°: 20071032

Orientador: Prof. Miguel Oliveira

Barcarena, Novembro de 2010



# UNIVERSIDADE ATLÂNTICA

## GESTÃO DE SISTEMAS E COMPUTAÇÃO

### BASE DE DADOS: DESENVOLVIMENTO E IMPLEMENTAÇÃO PARA GESTÃO E DISTRIBUIÇÃO DE ASSINATURAS

*Trabalho Final para Laboratório de GSC Aplicado*

Elaborado por: Filipe Ricardo da Costa Almeida

Aluno Nº: 20071032

Orientador: Prof. Miguel Oliveira

Barcarena, Novembro de 2010

Declaro que concedo à Universidade Atlântica uma licença não-exclusiva para arquivar e tornar acessível, o presente trabalho, no todo ou em parte.

Retenho todos os direitos de autor relativos ao presente trabalho, e o direito de o usar futuramente.

O autor é o único responsável pelas ideias expressas neste relatório.



## Agradecimentos

**Meus pais:** por tudo...

**Minha família:** pela compreensão e solidariedade constantes...

**Meu sogro Zé:** que partistes durante a elaboração deste trabalho, Deus te guarde no céu...

**Meus amigos:** pelo companheirismo e amizade...

**Meus colegas de GSC:** um grupo de desconhecidos há três anos, a maior das amizades no dia de hoje...

**Prof. Miguel Oliveira:** pela colaboração para que este trabalho chegasse ao destino...

**Meus professores:** pela partilha de conhecimento...

**Meus colegas de trabalho:** pela compreensão nestes três anos onde a minha ausência foi mais recorrente...

## **Base de Dados: desenvolvimento e implementação para gestão e distribuição de Assinaturas**

### **Resumo**

A criação, desenvolvimento e implementação de uma base de dados é um desafio permanente para um profissional de tecnologias de informação. O primeiro passo de compreender as motivações e as necessidades que o sistema obriga é fundamental para se atingir o sucesso.

A abordagem que envolve a modelação de diagramas utilizando as potencialidades da linguagem UML permitiu a compreensão de um problema que resultou na criação de uma base de dados destinada à gestão e distribuição de Assinaturas. A partir da modelação, foi possível desenvolver-se os princípios pelo qual se iriam gerar os objectos necessários para a construção e implementação num Sistema de Gestão de Base de Dados.

**Palavras-chave:** Base de dados, Modelação, Implementação

### **Abstract**

Building, developing and implementing a database is an ongoing challenge for an IT Professional. The first step to understand the motivations and needs for every new system is vital to achieving success.

The approach that involves modeling diagrams using the potentials of UML language, allowed the understanding of the problem that resulted in the creation of a database intended for management and distribution to Subscribers. From the modeling, it was possible to develop the principles by which it would generate the objects needed for building and implementing on a Database Management System.

**Keywords:** Database, Modeling, Implementation

## Índice

Agradecimentos .....	i
Resumo.....	ii
Índice.....	iii
Acrónimos e Siglas .....	iv
Índice de Figuras.....	v
1. Apresentação do Problema.....	1
2. Problema e Estrutura da Solução .....	3
2.1. Análise da oferta existente .....	4
2.2. Especificidades do problema.....	6
2.3. Desenho da Solução .....	11
2.3.1. Diagrama de Use Cases.....	11
2.3.2. Diagrama de Classes .....	21
2.3.3. Diagrama de Actividades .....	24
2.4. Modelação da Base de Dados .....	27
3. Implementação.....	33
3.1. Selecção do Sistema de Gestão de Base de Dados.....	34
3.1.1. Microsoft SQL Server .....	34
3.1.2. Oracle .....	35
3.1.3. MySQL.....	36
3.2. Implementação .....	39
3.2.1. Criação da base de dados, das tabelas e das relações .....	39
3.2.2. Segurança .....	42
3.2.3. Redundância.....	45
3.2.4. Replicação de dados.....	46
3.3 Considerações sobre o modelo.....	50
4. Conclusões e o futuro.....	51
Bibliografia .....	52

## Acrónimos e Siglas

**DER:** Diagrama de Entidade-Relacionamento. Modelo de dados que representa o nível conceptual do negócio.

**SGBD:** Sistema de Gestão de Base de Dados. Ferramenta de software que gere as bases de dados.

**SQL:** Structured Query Language. Linguagem utilizada para gerir base de dados em modelo relacional.

**UML:** Unified Modeling Language. Linguagem que especifica, constrói, visualiza e documenta sistemas de informação orientados por objectos.

## Índice de Figuras

Figura 1. Estrutura típica das aplicações de gestão de assinaturas.....	7
Figura 2. Estrutura proposta para a aplicação .....	8
Figura 3. Combinação actores-requisitos .....	20
Figura 4. Diagrama de use cases .....	20
Figura 5. Diagrama de classes.....	22
Figura 6. Diagrama de actividades na criação de uma assinatura.....	25
Figura 7. Diagrama de actividades na gestão de reclamação.....	26
Figura 8. Representação gráfica das diferenças das variantes de Chen e Martin.....	28
Figura 9. Diagrama de Entidade-Relacionamento completo.....	31
Figura 10. Diagrama de Entidade-Relacionamento reduzido .....	32
Figura 11. Comparação entre os SGBD's apresentados .....	37
Figura 12. Criação de uma nova base de dados em ambiente GUI no SQL Server 2008.....	40
Figura 13. Criação de tabelas através do <i>Server Management Studio</i> .....	41
Figura 14. Criação de uma relação entre tabelas.....	42
Figura 15. Permissões dos roles nos diversos <i>schemas</i> .....	43
Figura 16. Tabelas e respectivos <i>schemas</i> atribuídos.....	44
Figura 17. Database Mirroring com máquina <i>witness</i> .....	46
Figura 18. Quatro locais para implementação.....	47
Figura 19. Tipo de replicação modelo <i>merge</i> .....	48



## 1. Apresentação do Problema

O trabalho tem em vista a criação e implementação de um sistema de bases de dados, no âmbito de um projecto profissional para um grupo português de media.

O projecto originalmente passa pelo desenvolvimento de um sistema específico para a gestão e distribuição de assinaturas na actividade de negócio Imprensa, mas na sua globalidade, envolve um conjunto de objectivos faseados:

- Implementação do sistema de gestão de assinaturas e distribuição;
- Implementação do sistema de facturação de assinaturas;
- Implementação do sistema de gestão de produtos diversificados (também designados de subprodutos).

No entanto, este trabalho incidirá apenas sobre o primeiro objectivo, visto que as outras fases estão destinadas para serem efectuadas num período de tempo mais alargado e não merecerão grande relevância nesta abordagem.

Irá ser abordada a definição de requisitos e funcionalidades esperadas para o desenvolvimento de um sistema específico para gestão de assinaturas de imprensa, incluindo também a possibilidade de planeamento e distribuição das mesmas. Os requisitos serão apresentados e desenhados tendo em conta os princípios de uma linguagem de modelação: a UML.

Será abordada a implementação dos diagramas resultantes da modelação num Sistema de Gestão de Base de Dados, que será o repositório dos dados. Serão ainda apresentados os modelos considerados necessários para o bom funcionamento do sistema, de modo a minimizar a ocorrência de falhas.

O objectivo passa por conseguir criar um produto que satisfaça várias fases: angariação, gestão de assinaturas, expedição, distribuição.



## 2. Problema e Estrutura da Solução

O negócio de assinaturas de Imprensa (jornais, revistas) tem vindo a mudar gradualmente ao longo dos últimos anos. Aquilo que até há uns anos atrás era visto como a “simples” constituição de um ficheiro com nomes e moradas que seriam os destinatários dos produtos (jornais ou revistas), passou a ser tratado com maior rigor. Informações que antes seriam consideradas como acessórias, hoje em dia têm um tratamento já rigoroso (exemplificando, cada vez mais existe a procura por parte dos editores em conhecer determinado tipo de informações do seu público como sectores de actividade para que possam ser incluídas em futuras campanhas, bem como os contactos, tipo de instituições, gostos, hobbies, etc.). Com o crescimento das assinaturas online e com o crescimento gradual de produtos extras que acompanham o produto principal (seja jornal ou revista), acresceu também a preocupação junto dos editores na obtenção de informações mais fidedignas do seu universo de clientes.

Com este aumento da procura de informações por parte das equipas editoriais, passou a existir uma propagação de softwares que ainda hoje têm como preocupação ajudar a estabelecer alguns parâmetros ao nível da gestão das assinaturas, para contrapor ao que era habitual (como a gestão através de ficheiros de Excel e de Word que apesar de tudo ainda hoje são usuais).

Este capítulo iniciar-se-á com uma síntese da análise feita a várias aplicações já existentes no mercado, evidenciando as principais características de cada uma delas. Depois, apresentar-se-ão as especificidades do nosso problema, incluindo as que as aplicações analisadas não contemplam. Para terminar, será apresentado o desenho da solução proposta.

## 2.1. Análise da oferta existente

Foram analisadas algumas aplicações disponíveis no mercado, com o objectivo de perceber se alguma delas conseguiria satisfazer as necessidades entretanto pensadas para o projecto:

- **Sistema de Gestão Global de Assinaturas** da *Uanetuane* (<http://uanetuane.com>): é o sistema vigente no grupo de media a que se destina este projecto. Contém diversos módulos como o de gestão de assinaturas, de facturação, de venda de produtos diversificados e gestão de reclamações. É um sistema que tem uma orientação “um exemplar → um assinante”. Apresenta um módulo de gestão de assinaturas e de subprodutos satisfatório, apesar de apresentar um contra que consiste no caso de um cliente querer vários exemplares de uma mesma publicação, obriga à abertura do número de fichas iguais ao número de exemplares requeridos. No entanto, durante a consulta, houve informação de que tal estaria a ser corrigido. As fichas de assinante contêm informação variada e pensada para campanhas de marketing. Porém, o sistema não consegue distinguir produtos diferentes de uma mesma publicação, obrigando à abertura de publicações diferentes para definir cada um deles. Em termos de distribuição, é praticamente inexistente.
- **Solução de Gestão de Assinaturas** da *MCComputadores* (<http://www.mccomputadores.com>): sistema vigente no editor *Diário de Notícias da Madeira*, foi construído com o suporte da aplicação *Microsoft Dynamics NAV* (<http://www.microsoft.com/en-us/dynamics/products/nav-overview.aspx>) e tem a parceria do grupo dinamarquês *IT Gruppens AS*. Possibilita a gestão de assinaturas, facturação e distribuição. Tem um sistema de facturação muito completo e que permite ligar várias assinaturas a uma entidade. O preenchimento da ficha de assinatura contempla informações de modo que possam ser integradas em campanhas de marketing. Permite o registo de reclamações, mas não permite efectuar qualquer tipo de gestão sobre as mesmas. Não contempla qualquer possibilidade de gestão de subprodutos, sendo que qualquer gestão deste tipo terá de ser efectuada fora do sistema. Também não

efectua distinção de produtos numa mesma publicação: para existir produtos diferentes no sistema, obriga à criação de novas publicações. Contém um módulo de distribuição, mas o mesmo é muito rígido e incompleto, só estando preparado para fornecer apenas informações de distribuição básicas (como a rota e as condições de entrega) das assinaturas entregues pelo editor, ignorando rotas que sejam efectuadas por entidades externas.

- **Sirius** da *Credimática* (<http://www.credimatica.com>): é apresentado pelos mesmos como um dos que tem maior implementação em Espanha, principalmente na região da Catalunha. Também já se encontra razoavelmente posicionado no mercado da América Latina, principalmente no México. É um sistema que permite a gestão de assinaturas, de facturação, de distribuição, da gestão de subprodutos, de reclamações, de gestão em bancas e com um módulo de atendimento. A aplicação em si tem um modelo de gestão de assinaturas muito satisfatório, sendo que as próprias fichas de assinante permitem o preenchimento de informações variadas e pensadas para campanhas de marketing. Pode-se inclusive, com a compra de um módulo próprio, efectuar a gestão das campanhas de marketing. A gestão de subprodutos é bastante completa e para o utilizador comum torna-se relativamente intuitivo. Permite o registo e a gestão das reclamações. Já prevê a possibilidade de que uma entidade possa patrocinar uma assinatura, mas tal só é possível nos casos em que a entidade seja uma pessoa colectiva. Não efectua a distinção de diferentes produtos numa mesma publicação: nestes casos, obriga à criação de novas publicações. O modelo adoptado para o módulo de distribuição é rígido, e confuso na sua concepção e gestão. A aplicação está disponível apenas em língua espanhola. Das aplicações que analisamos, foi ainda assim a que se considerou como a mais interessante.

Praticamente todas as aplicações consultadas apresentam o problema de conseguir interligar com satisfação os processos de gestão da assinatura com os processos de distribuição. Normalmente, focam apenas uma das áreas do negócio e como o mercado-alvo são os grupos editoriais, privilegiam a componente da gestão de assinaturas, descurando o processo de distribuição.

## 2.2. Especificidades do problema

Explicando sucintamente como ocorre o processo de uma assinatura, este passa por várias fases:

1. Solicitação de assinatura para uma ou mais publicações por parte de um potencial cliente;
2. Caso seja uma assinatura paga, terá que existir a recepção do pagamento nas modalidades ao qual o editor aceita (como por exemplo cheque, multibanco, nalguns casos em dinheiro) e envio da factura/recibo ao cliente (caso seja uma Venda Especial<sup>1</sup>, a recepção do pagamento é efectuada após a recepção da factura por parte do cliente);
3. Inserção dos dados no sistema utilizado por parte da equipa responsável pela gestão de assinaturas;
4. Verificação do endereço e colocação numa rota por parte da(s) equipa(s) de distribuição;
5. Geração de serviço em cada edição, para impressão de rótulos e/ou etiquetas dos assinantes activos, e caso necessário, a impressão dos mapas de suporte.

Um dos problemas que as aplicações consultadas apresentam é esta falta de articulação, sendo mais visível nos casos em que ocorre a gestão de assinaturas e a distribuição. Isto leva a que acabem por ser adoptadas duas soluções distintas para as duas componentes, o que aumenta o risco de que as ligações entre as duas soluções possam apresentar falhas e acabem por ser dispendiosas.

Pelas razões que foram sendo apresentadas, foi decidido avançar-se com um novo projecto que consiste na criação de um sistema de gestão global, e que permita acompanhar todo o trajecto de uma assinatura sem que nenhuma das etapas seja descurada.

---

<sup>1</sup> É um tipo de assinatura que normalmente os editores gostam de diferenciar, porque a estes clientes é-lhes permitido efectuar o pagamento ao fim de determinado período de tempo, conforme as quantidades que entretanto receberam. É-lhes dada a possibilidade de alterar as quantidades que pretendam receber em cada edição. Normalmente, este tipo de designação só é atribuído pelos editores a entidades que sejam reconhecidamente confiáveis e que tenham celebrado este contrato (como companhias aéreas, hotéis, etc.).

Algumas das aplicações que foram consultadas são baseadas na orientação: “um assinante, uma publicação”. Essa orientação tem deixado de fazer tanto sentido, visto cada vez mais existir pressão dos grupos editoriais junto dos departamentos ou das empresas de distribuição para se fornecer o maior número possível de entidades e exemplares, com o maior leque de produtos possível. A agressividade aplicada no marketing directo tem sido uma aposta cada vez mais recorrente junto do público-alvo. Tem sido mais recorrente entidades receberem não um, mas três, quatro ou cinco produtos diferentes. Aplicando o paradigma que se encontra em algumas das aplicações analisadas, a mesma informação tem de ser repetida três, quatro, cinco vezes, sem que muitas vezes seja possível especificar uma ligação lógica entre elas. O proposto é efectuar essa ligação, apresentando um determinado tipo de informação como o nível máximo e apresentar todas as assinaturas como pertencentes e logicamente ligadas a ele. Basta atentar nas diferenças apresentadas nas figuras seguintes.

MediaCare	João Afonso	MediaCare	MediaCare Cuidados Médicos Lda	Carlos Lopes
<i>Contribuinte:</i> 123456789	<i>Contribuinte:</i> 210987654	<i>Contribuinte:</i> 123456789	<i>Contribuinte:</i> 123456789	<i>Contribuinte:</i> 201654378
Av República	R República da Coreia	Av Marechal Gomes da Costa	Av República	R Ouro
Diário de Notícias	Diário de Notícias	Record	Expresso	Record

Figura 1. Estrutura típica das aplicações de gestão de assinaturas

Na figura acima apresentada (**Figura 1**), em cinco registos, existem três que têm uma ligação entre elas. Olhando para o contribuinte e para o nome, percebe-se que a *MediaCare* está presente em 3 assinaturas. No modelo existente, pode implicar a abertura de três fichas de assinantes, com o preenchimento de informação em triplicado (nome: *MediaCare*; Contribuinte: *123456789*; Actividade: *Serviços em Saúde*, etc.), e constante nas três fichas, mas sem se definir qualquer ligação entre os registos. Acresce desde logo uma probabilidade de erro maior: qualquer alteração de um dos registos, terá

de ser replicada nas restantes fichas, mas como não existe ligação entre elas no sistema de dados, dependerá da sensibilidade e conhecimentos dos dados por parte dos utilizadores humanos, podendo acontecer que a sua detecção não seja verificada.

MediaCare			João Afonso	Carlos Lopes
Contribuinte: 123456789			Contribuinte: 210987654	Contribuinte: 201654378
Av República		Av Marechal Gomes da Costa	R República da Coreia	R Ouro
Diário de Notícias	Expresso	Record	Diário de Notícias	Record

Figura 2. Estrutura proposta para a aplicação

Neste caso, e o que é proposto para o sistema em construção (**Figura 2**), as assinaturas possuem uma ligação entre si, estabelecendo-se uma relação entre elas. A partir do mesmo número de contribuinte, é possível diferenciar publicações e endereços de distribuição diferentes. O proposto é a definição da entidade como nível máximo, e na qual pode ter várias assinaturas associadas. Caso exista alguma alteração da entidade (como o endereço principal, nome, sector de actividade, etc.) basta efectuar-la apenas numa ficha, para que a mesma possa ser replicada nas assinaturas que possui, ao contrário do que foi verificado anteriormente. Tem outra vantagem: torna-se possível saber ao certo o número de assinaturas que cada entidade possui e quais as publicações que recebe, situação que no modelo anterior depende da sensibilidade por parte dos utilizadores relativamente aos dados, como já referido.

Após uma breve explicação sobre a orientação planeada para o sistema (entidade → assinatura), apresenta-se uma lista de funcionalidades esperadas para o sistema:

- Criação de entidades que permita agregar as várias assinaturas podendo ter publicações diferentes (como exemplo: permitir a criação de uma entidade *MediaCare*, que receba as publicações *Diário de Notícias*, *Record*, *Visão*). Só

fará sentido criar uma entidade sempre que a mesma tenha direito a receber pelo menos uma assinatura;

- Consulta e actualização dos dados das entidades registadas, exceptuando a informação existente no número de contribuinte e no ID que lhe é primariamente fornecido pelo sistema (exemplo: na entidade *MediaCare*, pode-se alterar a designação para *MediaCare Cuidados Médicos Lda* ou então alterar o endereço da entidade);
- Criação de uma assinatura a qual terá de estar obrigatoriamente ligada a uma entidade, com preenchimento das informações relativa à mesma e sendo obrigatório definir a gratuidade (exemplo: criar uma assinatura na entidade *MediaCare* definindo-a como oferta ou então criar uma assinatura paga pela entidade *MediaCare* mas a ser destinada ao colaborador *António Santos*);
- Consulta e actualização dos dados de assinaturas activas (como o endereço, o destinatário da mesma, tipo de entrega, rota). Para se poder alterar os dados de uma assinatura suspensa, a mesma terá de ser activada primeiramente;
- Os produtos (jornais ou revistas) têm de estar ligados a uma assinatura, com possibilidade de se alterar as quantidades para cada edição;
- Possibilidade de associar a uma assinatura dois ou mais produtos diferentes (exemplo: *Diário de Notícias* e *Visão*), mas terão que partilhar obrigatoriamente o mesmo tipo de gratuidade, data de início, data de fim e rota.
- Criação de rotas de distribuição, com possibilidade de definir a posição das assinaturas na rota, bem como associar um tipo de entrega possível e a sua prioridade para efeitos de expedição;
- Actualização dos dados das rotas de distribuição, como o nome da rota, prioridade, distribuidor e a posição de entrega das assinaturas;
- Introdução de reclamações de não entrega de produtos numa assinatura;
- Gestão de reclamações que se encontram por fechar (ou seja, aquelas que ainda não se encontram resolvidas), nomeadamente com a possibilidade de inserção de data de resolução, descrição da resolução e outras informações consideradas pertinentes;

- Criação de fecho de serviço (uni-publicação ou multi-publicação) para permitir um histórico de entregas das assinaturas em cada edição;
- Criação de histórico sobre as entidades e assinaturas sempre que se registar alterações nos respectivos dados (como nome, morada, quantidade, etc.) sendo activado de forma automática.

Os intervenientes a quem é destinada esta solução e que podem consultar e manipular os dados são pertencentes ao departamento de assinaturas e ao departamento de distribuição.

## 2.3. Desenho da Solução

O desenho da solução foi feito tendo em conta os princípios enunciados pela *UML*. Apesar de existir um leque de diagramas possíveis de serem efectuados, neste trabalho só serão incluídos alguns deles: o de Use Cases, o de Classes e o de Actividades.

Para ajudar na elaboração dos diagramas da linguagem UML, foi utilizado o software *Visual Paradigm*, disponibilizado em <http://www.visual-paradigm.com>.

### 2.3.1. Diagrama de Use Cases

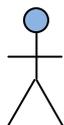
O diagrama de use cases é uma técnica onde se pode representar os requisitos de um sistema. O desenho de um diagrama de use cases pressupõe na sua concepção, a existência de actores do sistema, que são aqueles que irão interagir com o sistema. Pressupõe também a existência de requisitos a serem cumpridos e nos quais os actores terão que ter a capacidade de interagir com o sistema para poderem cumprir com o funcionamento dos requisitos que lhes são destinados.

Mas primeiramente terá de ser abordado o âmbito do sistema a ser desenhado. Com as indicações que entretanto foram sendo prestadas, pode-se definir o âmbito do projecto como o seguinte:

**Âmbito:**

Desenvolvimento de um sistema de informação para gestão e distribuição de assinaturas, de modo que seja passível de utilização por parte dos departamentos de assinaturas e dos departamentos/empresas de distribuição.

Tendo em conta a descrição do problema abordado e já com o âmbito definido, são apresentados os actores que irão estar em contacto com o sistema.

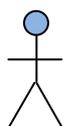


### **Departamento de Assinaturas**

Responsável pela gestão de assinaturas, nomeadamente com a possibilidade de inserir os dados das entidades e assinaturas, bem como modificá-los. Responsável pelo processamento dos fechos das edições das publicações e por inserir as reclamações apresentadas pelos assinantes.

Entenda-se por “Departamento de Assinaturas” todos os elementos que irão assegurar a gestão e o contacto junto dos assinantes. Podem ser elementos internos à organização ou elementos que têm um contrato de prestação de serviços junto dos editores para cumprir essas funções. Cabe assegurar aos elementos que constituem o Departamento de Assinaturas, o atendimento telefónico e/ou presencial de potenciais clientes e assinantes efectivos. Presta também vários serviços de informação como preços, publicações disponíveis e outros tipos mais que os clientes possam necessitar. Pode receber indicações por parte dos assinantes de alterações nas fichas individuais (como o nome ou a morada) e receber reclamações apresentadas pelos mesmos e verificar junto deles, sempre que possível, a correcta resolução das mesmas.

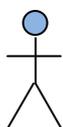
Na sua interacção com o sistema, é da sua responsabilidade registar os dados necessários para a criação de uma entidade/assinatura, bem como efectuar a gestão desses dados sempre que necessário (informação prestada pela entidade/assinante, por exemplo). Tem de inserir as reclamações apresentadas pelos assinantes, para que as mesmas possam ser consultadas pelo Departamento de Distribuição, e terá de validar as resoluções efectuadas através do contacto junto dos assinantes. Em caso da mesma não ter sido cumprida, informará de novo o Departamento de Distribuição. Uma das funções mais importantes na relação com o sistema prende-se com o fecho de serviço: cabe ao Departamento de Assinaturas dar a ordem de para o processamento do fecho das publicações para cada número de edição.



### **Responsável do Departamento de Assinaturas**

Pertencente ao Departamento de Assinaturas, além das funções reservadas a esse actor, é responsável pela gestão dos recursos de informação necessários para o bom funcionamento do sistema (como os países existentes, o tipo de assinaturas disponíveis, etc.)

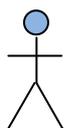
Aconselha-se que este actor seja de facto apenas um indivíduo, preferencialmente o responsável pelo departamento. Além de ter acesso às mesmas funcionalidades que qualquer elemento do Departamento de Assinaturas, é-lhe permitido gerir informações primordiais para o correcto funcionamento do sistema como tais como os países existentes, os tipos de assinaturas disponíveis, as publicações que podem ser disponibilizadas no sistema, entre outras. É o supervisor da qualidade dos dados que podem ser seleccionados, tendo a possibilidade de os criar, alterar ou mesmo suspender.



<b>Departamento de Distribuição</b>
Responsável pela distribuição de assinaturas, tem o dever de organizar as rotas de distribuição e colocar as assinaturas devidamente ordenadas. Responsável pela execução das resoluções e respostas às reclamações.

Podendo ser um departamento interno na organização ou então uma empresa exterior especializada, cabe-lhe assegurar a distribuição das assinaturas activas. Tem a responsabilidade de organizar as rotas necessárias, de arranjar distribuidores e de preparar todo o plano de distribuição. Tem também de responder às reclamações apresentadas e colocar em prática as resoluções acordadas com os assinantes.

Na interacção com o sistema, é da responsabilidade do Departamento de Distribuição a criação de rotas, bem como associar um tipo de entrega para cada rota (correios, distribuição nocturna, distribuição diurna, distribuição interna). Tem de gerir as rotas no sistema, colocando as assinaturas nas rotas respectivas e proceder à ordenação das mesmas, para que no processamento do serviço e na extracção de relatórios de apoio à distribuição, surjam devidamente ordenadas de modo a não criar situações dúbias junto dos distribuidores. Uma outra interacção é verificar as reclamações existentes e que se encontram em aberto (ou seja, que ainda estão por responder) e efectuar o preenchimento dos dados relativos à resolução das mesmas, aquando da sua concretização.



**Sistema**

Representativo do sistema que está a ser criado.

É sobre este actor que os outros actores irão cumprir as funcionalidades previstas. No entanto, o mesmo foi aqui colocado para que possam ser visualizados um par de requisitos considerados importantes, como o facto de automaticamente registar alterações nas entidades e assinaturas colocando-as no histórico e também registar automaticamente informações de distribuição. É o próprio sistema que tem de levar a cabo o cumprimento destes dois requisitos.

Após definição dos actores que irão interagir com o sistema, o próximo passo passa pela definição das funcionalidades esperadas. Estes requisitos são considerados essenciais para o bom funcionamento do sistema e na interacção com os vários actores. Basicamente, é um serviço que deve ser colocado à disposição dos utilizadores. No problema enunciado, destacam-se os seguintes requisitos:

<b>Use Case:</b>	<b>Introduzir Entidade</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e a entidade não pode estar registada no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma opção de 'Nova Entidade';</li><li>2. São inseridos os dados da entidade (Número de Contribuinte, Nome, Morada);</li><li>3. Após os dados terem sido preenchidos, os dados são confirmados.</li></ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação, seleccionando a respectiva opção.
<b>Pós-condição:</b>	A entidade fica criada e disponibilizada no sistema.

<b>Use Case:</b>	<b>Introduzir Assinatura</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e a entidade existente no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma entidade;</li><li>2. Se a entidade não estiver registada:<ol style="list-style-type: none"><li>a. <b>Extend: Introduzir Entidade</b></li></ol></li><li>3. Selecciona-se uma opção de ‘Criar Assinatura’;</li><li>4. Insere-se os dados necessários (destinatário, endereço, publicação, quantidade, tipo de entrega, data de início, tipo de assinatura, etc.);</li><li>5. Após o preenchimento dos dados, os mesmos são confirmados.</li></ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação, seleccionando a respectiva opção.
<b>Pós-condição:</b>	A assinatura fica criada e disponibilizada no sistema, ficando interligada a uma entidade.

<b>Use Case:</b>	<b>Actualizar Assinatura</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e a entidade existente no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma entidade;</li><li>2. Selecciona-se uma assinatura que esteja ligada à entidade;</li><li>3. Podem ser alterados dados da assinatura (destinatário, endereço, quantidade, tipo de entrega, tipo de assinatura).</li></ol>
<b>Pós-condição:</b>	A assinatura fica disponível no sistema actualizada.

<b>Use Case:</b>	<b>Registar Reclamação</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema, a entidade e a assinatura têm de existir no sistema e a edição válida no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma entidade e uma assinatura que esteja dependente desta</li><li>2. É escolhida uma edição sobre a qual já exista distribuição;</li><li>3. Selecciona-se uma opção de 'Inserir Reclamação';</li><li>4. Preenche-se os dados necessários (tipo de reclamação, causa provável, tipo de resolução solicitada, quantidade);</li><li>5. Após os dados terem sido preenchidos, os mesmos são confirmados.</li></ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação, seleccionando a respectiva opção.
<b>Pós-condição:</b>	A reclamação fica criada e disponibilizada no sistema ligada a uma assinatura.

<b>Use Case:</b>	<b>Gerir Reclamação</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema, a entidade e a assinatura têm de existir no sistema e a reclamação existente no sistema mas não como 'Concluída'.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma entidade e uma assinatura que tenha indicação de reclamação pendente à mesma;</li><li>2. Selecciona-se a opção 'Reclamações';</li><li>3. Podem ser actualizados os dados da reclamação (causa da reclamação, resolução efectuada, data de resolução, observações, estado).</li></ol>
<b>Pós-condição:</b>	A reclamação fica registada com as novas informações. Se ficar actualizada no seu estado como 'Concluída', deixa de ser possível de se efectuar mais alterações, permitindo apenas a sua visualização.

<b>Use Case:</b>	<b>Registrar Rota de Distribuição</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e a rota não pode estar registada no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma opção 'Criar Rota';</li><li>2. Preenche-se os dados necessários (código-sigla de rota, nome da rota, tipo de entrega, localização primária, responsável);</li><li>3. Após os dados terem sido preenchidos, os mesmos são confirmados.</li></ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação, seleccionando a respectiva opção.
<b>Pós-condição:</b>	A rota de entrega fica criada e disponibilizada no sistema.

<b>Use Case:</b>	<b>Actualizar Rota de Distribuição</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e a rota têm de existir no sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma rota;</li><li>2. Podem ser alterados os dados da rota (código-sigla, nome, localização primária, responsável);</li><li>3. Podem ser geridas as assinaturas na rota (inserir, retirar, alterar posição).</li></ol>
<b>Pós-condição:</b>	A rota fica disponível no sistema actualizada.

<b>Use Case:</b>	<b>Processar Fecho de Edição</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma opção 'Processar Fecho de Edição';</li><li>2. É escolhida a data de edição e a(s) publicação(ões);</li><li>3. É seleccionada a opção 'Correr Fecho' para dar início ao processamento do fecho de edição.</li></ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação, seleccionando a respectiva opção.
<b>Pós-condição:</b>	O histórico de entregas de assinaturas fica preenchido.

<b>Use Case:</b>	<b>Criar Histórico de Entregas</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema, a edição e respectiva publicação não existentes no histórico.
<b>Descrição:</b>	<ol style="list-style-type: none"> <li>1. O sistema verifica se a data de edição e a publicação escolhidas na sua globalidade já estão disponibilizadas no histórico;</li> <li>2. Em caso negativo, processa os dados para o histórico;</li> <li>3. Em caso positivo, dá informação de 'Existente em histórico'.</li> </ol>
<b>Caminhos Alternativos:</b>	Cancelando a operação de “Processar Fecho de Edição” origina a que o carregamento do histórico seja automaticamente cancelado.
<b>Pós-condição:</b>	Fica disponibilizado no histórico de entregas as informações respeitantes aos assinantes sobre a edição.

<b>Use Case:</b>	<b>Gerir Dados de Apoio</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"> <li>1. O use case começa quando é seleccionada uma opção ‘Gerir Informações de Apoio’;</li> <li>2. É escolhida qual a informação que se pretende gerir mediante uma lista pré-definida (países, endereços, publicações);</li> <li>3. Após os dados serem preenchidos, os mesmos ficam confirmados.</li> </ol>
<b>Pós-condição:</b>	As informações ficam disponíveis no sistema actualizadas.

<b>Use Case:</b>	<b>Registar Tipo de Entrega</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema e o tipo de entrega não pode estar registado pelo sistema
<b>Descrição:</b>	<ol style="list-style-type: none"> <li>1. O use case começa quando é seleccionada uma opção ‘Registar Tipo de Entrega’;</li> <li>2. São inseridas as informações relacionadas com o tipo de entrega;</li> <li>3. Após os dados serem preenchidos, os mesmos ficam confirmados.</li> </ol>
<b>Caminhos Alternativos:</b>	A qualquer momento, é possível cancelar a operação.
<b>Pós-condição:</b>	O tipo de entrega fica registado e disponibilizado no sistema.

<b>Use Case:</b>	<b>Consultar Relatórios de Edição</b>
<b>Pré-condição:</b>	O utilizador tem de estar validado pelo sistema.
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é seleccionada uma opção de 'Consultar edição';</li><li>2. É escolhida a data de edição e se necessário, publicação (por defeito, estará seleccionada "Todas as Publicações");</li><li>3. Caso já esteja disponibilizada a edição, aparece uma informação para se escolher 'Quantidade por rota', 'Quantidade por Publicação', "Quantidade por tipo de entrega", "Roteiros de Distribuição", "Guias de aceitação".</li></ol>
<b>Pós-condição:</b>	-

<b>Use Case:</b>	<b>Registar Alterações de Assinaturas</b>
<b>Pré-condição:</b>	-
<b>Descrição:</b>	<ol style="list-style-type: none"><li>1. O use case começa quando é efectuada uma actualização nos dados de uma entidade ou assinatura, sendo disparado automaticamente um procedimento;</li><li>2. Dá origem a que seja carregadas informações no histórico de alterações como o campo actualizado, informação anterior, data de alteração, utilizador.</li></ol>
<b>Pós-condição:</b>	O histórico de alterações fica com o registo sobre as informações antes de depois de actualização.

Com a apresentação dos actores e a definição dos requisitos efectuada, procede-se à ligação entre eles, definindo-se quem deve poder efectuar o quê (**Figura 3**):

<b>Departamento de Assinaturas</b>	<ul style="list-style-type: none"> <li>➤ Introduzir Entidade</li> <li>➤ Introduzir Assinatura</li> <li>➤ Actualizar Assinatura</li> <li>➤ Registar Reclamação</li> <li>➤ Gerir Reclamação</li> <li>➤ Processar Fecho de Serviço</li> </ul>
<b>Responsável de Departamento</b>	<ul style="list-style-type: none"> <li>➤ Introduzir Entidade</li> <li>➤ Introduzir Assinatura</li> <li>➤ Actualizar Assinatura</li> <li>➤ Registar Reclamação</li> <li>➤ Gerir Reclamação</li> <li>➤ Processar Fecho de Serviço</li> <li>➤ Gerir Informação de Apoio</li> </ul>
<b>Departamento de Distribuição</b>	<ul style="list-style-type: none"> <li>➤ Registar Rota de Distribuição</li> <li>➤ Actualizar Rota de Distribuição</li> <li>➤ Registar Tipo de Entrega</li> <li>➤ Consultar Relatórios de Edição</li> <li>➤ Gerir Reclamação</li> </ul>
<b>Sistema</b>	<ul style="list-style-type: none"> <li>➤ Registar Alterações de Assinaturas</li> <li>➤ Criar Histórico de Entregas</li> </ul>

Figura 3. Combinação actores-requisitos

Juntando os requisitos, os actores e seus relacionamentos na execução de um diagrama de Use Cases, obtêm-se o seguinte desenho (**Figura 4**):

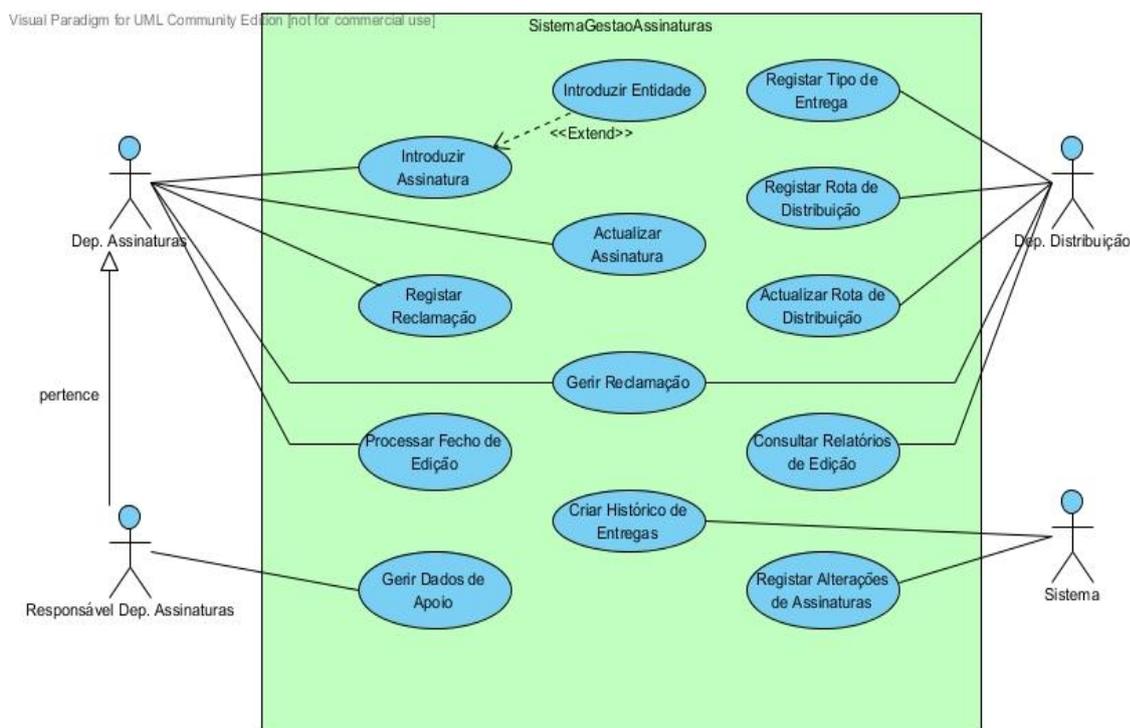


Figura 4. Diagrama de use cases

Como se verifica na imagem representativa do diagrama, os quatro actores estão presentes, estando ligados aos requisitos definidos.

Há dois aspectos que merecem destaque. Nota-se a existência de uma relação de generalização (representada por uma seta) que parte do Responsável do Dep. Assinaturas para o actor Dep. Assinaturas.

Quando existe uma representação de generalização, significa que existe um relacionamento entre um elemento mais geral (que denominaremos como *pai*) com um outro mais específico (que denominaremos de *filho*), sejam eles actores ou use cases, e no qual o *filho* herda as propriedades do *pai*. Esta representação é utilizada em casos de existência de um comportamento adicional por parte do *filho*. Neste caso específico, o actor *filho* Responsável Dep. Assinaturas e o actor *pai* Dep. Assinaturas partilham muitas das tarefas e funcionalidades, mas no entanto cabe ao *filho* preencher um requisito extra. Deste modo, é criada uma generalização para o actor Dep. Assinaturas (visto que já se verificou que partilha das mesmas funcionalidades) e liga-se directamente ao actor *filho* apenas os requisitos adicionais.

Outra particularidade a destacar no diagrama é a existência de uma relação de *extend*. Deve ser utilizada quando existe um comportamento normalmente opcional e o qual deve ser incluído num use case. No caso apresentado, quando o utilizador cria uma assinatura, deve verificar se existe a entidade com que deve estar relacionada. Caso não exista, terá de criá-la. Este procedimento reflecte-se no diagrama, com a colocação do requisito “*Introduzir Entidade*” que está ligado através de um ponto de extensão ao requisito “*Introduzir Assinatura*”. E na explicação do requisito respectivo, deve constar uma situação de *extend*, como se verifica no quadro de “*Introduzir Assinatura*” (conforme se verifica no quadro respectivo na página 15).

### **2.3.2. Diagrama de Classes**

Fazendo ainda parte das especificações UML, foi elaborado também o diagrama de classes, que é uma descrição formal da estrutura de objectos (identidade,

relacionamento com outros objectos, atributos, operações) num sistema. Um diagrama de Classes mostra as diferentes classes que compõem um sistema e como é que elas se relacionam umas com as outras.

Neste caso, optou-se por simplificar o processo, e tendo em conta os requisitos já definidos, foi desenhado um modelo que resultou da análise do diagrama de use cases. Como tal, o desenho a apresentar (**Figura 5**), só faz referência às tabelas de produção do sistema em construção.

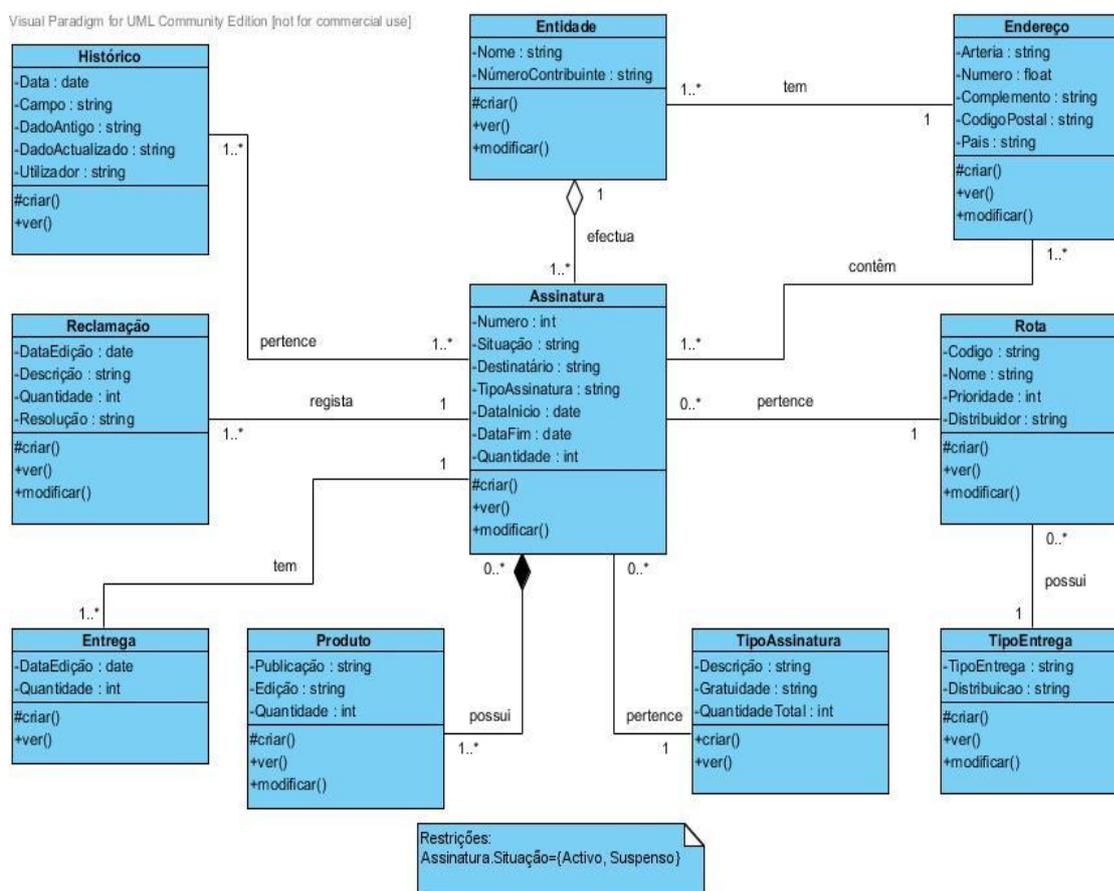


Figura 5. Diagrama de classes

Neste modelo, foram incluídas duas relações: uma de agregação e uma outra referente a uma composição. A opção sobre estes dois tipos serve para indicar a existência de uma relação mais forte entre as classes.

Apesar de não ser fácil distinguir-se um do outro<sup>2</sup>, optou-se por uma relação de agregação entre as classes Entidade e Assinatura de modo a indicar que ambas fazem parte de um todo. Apesar de ser possível a criação de uma Entidade sem colocar uma Assinatura dependente, tal não faz sentido porque só estaria a ocupar espaço na base de dados. Como tal, para referir que existe uma relação forte entre ambos, optou-se por indicar uma relação de agregação.

Coloca-se a questão do porquê não se ter optado pelo mesmo tipo de relação entre as classes Assinatura e Produto, mas sim por uma relação de composição. Esta opção foi escolhida com o intuito de mostrar que esta relação tem um significado ainda mais forte, indicando uma dependência quase total entre as duas classes. Não faz sentido existir uma Assinatura sem ter um produto associado para o assinante receber. Tal como não faz sentido o contrário, existir um produto mas sem ninguém para o receber. Com esta relação de composição, pretende-se indicar que a relação é tão forte que no caso de uma parte não existir, a outra parte perde o seu sentido e deixa de existir também.

O diagrama de classes contém também uma nota que apresenta uma restrição, que indica quais as possibilidades que devem existir em determinado atributo. No caso, indicou-se que a situação de uma assinatura só pode ter duas hipóteses:

- Activo, sinal de que o destinatário da assinatura tem direito a receber os produtos que estão contemplados na sua ficha;
- Suspenso, que indica que o mesmo não se encontra disponível para receber qualquer produto, por uma razão específica (como falta de interesse, não pagamento, data de término ultrapassada).

A indicação da restrição torna-se bastante útil pois notifica desde logo os desenhadores que irão construir o sistema, quais as opções que terão de disponibilizar naquele atributo específico.

---

<sup>2</sup> Nunes, M.; O'Neill, H., Fundamental de UML, Lisboa: FCA, 2004, pp. 51

### 2.3.3. Diagrama de Actividades

Um outro modelo utilizado na linguagem de modelação é o diagrama de actividades que se torna útil na descrição de fluxos de trabalho. Permite entender quais as actividades que são necessárias para se atingir determinados objectivos, bem como definir os actores que irão participar na execução desses mesmos objectivos.

São apresentados dois diagramas referentes a situações específicas. O primeiro (**Figura 6**) diz respeito ao fluxo utilizado durante a criação de uma assinatura onde os intervenientes são o Dep. Assinaturas e o Dep. Distribuição. Pelo fluxo, entende-se que uma boa parte da execução do esquema está relacionada directamente com o primeiro interveniente. No entanto, aparece uma série de questões ao qual terão de existir respostas, como por exemplo logo após a recepção do pedido em que o elemento terá de discernir o tipo de assinatura, porque dependendo desta, terá opções diferentes.

Anota-se também a hipótese de ser uma assinatura paga e se a mesma não tiver sido liquidada, o pedido é cancelado e o processo dá-se por terminado.

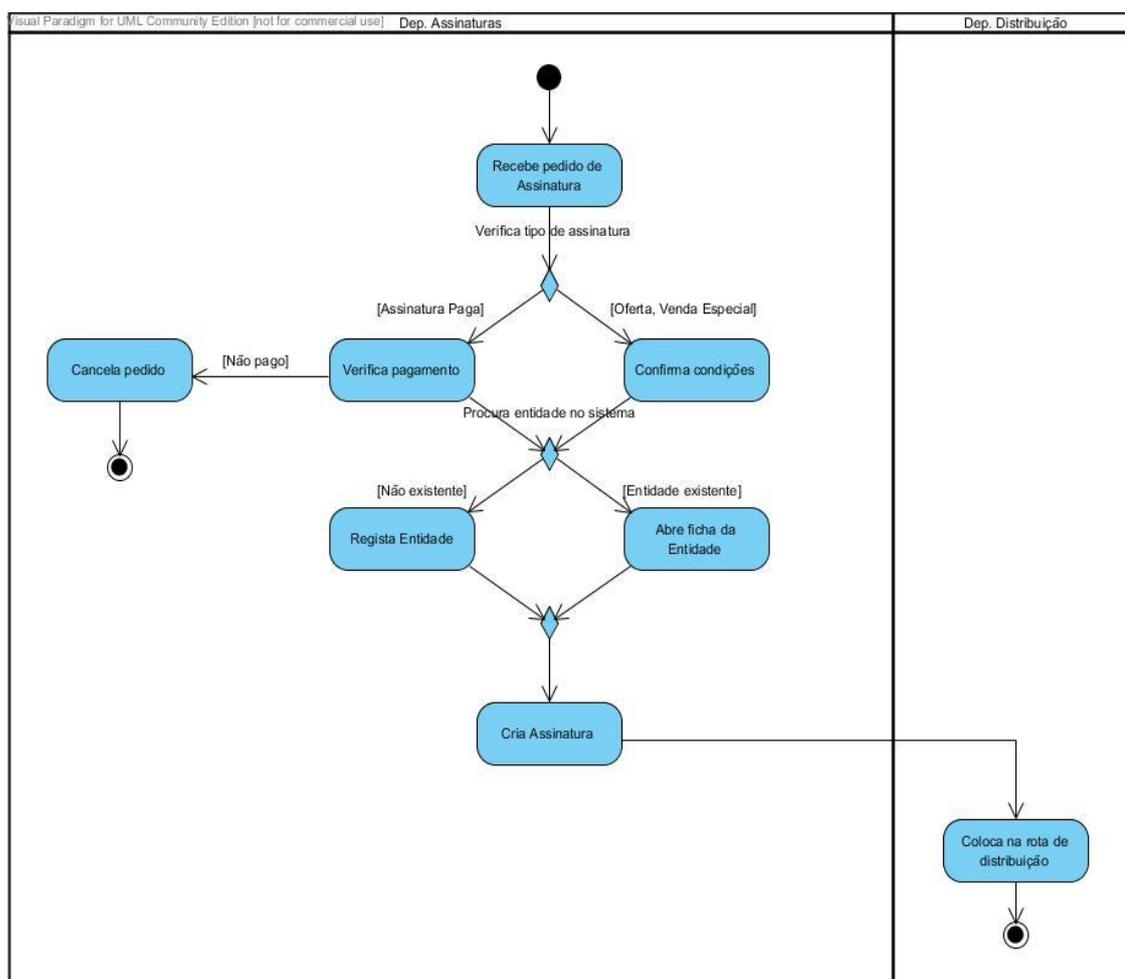


Figura 6. Diagrama de actividades na criação de uma assinatura

Um outro exemplo de um fluxo de actividades é o processo de reclamação e respectiva resolução. Neste caso (**Figura 7**), são três os intervenientes, com a existência de uma situação de decisão, sendo necessária a confirmação por parte do cliente sobre a resolução efectiva da reclamação. Em caso da resposta ser negativa, o Dep. Assinaturas fornece essa informação ao Dep. Distribuição, que por sua vez, voltará a proceder a mais uma tentativa de resolução conforme acordada com o cliente. Enquanto este passo não for completado, o processo entrará num ciclo.

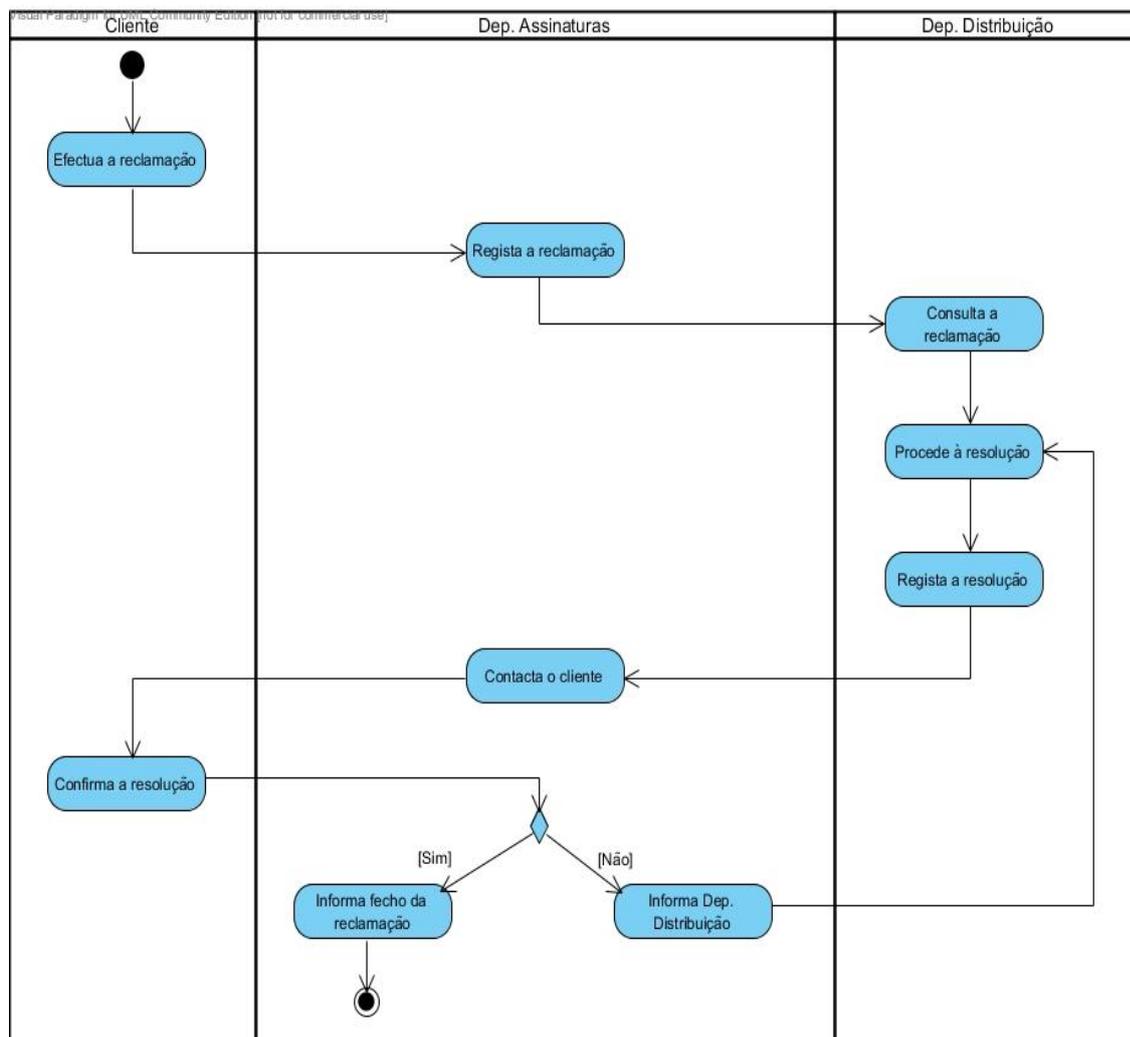


Figura 7. Diagrama de actividades na gestão de reclamação

## 2.4. Modelação da Base de Dados

Após concluída a elaboração de requisitos e a definição do funcionamento esperada do sistema através da linguagem UML, o passo seguinte ao qual ponto aborda é a modelação da base de dados.

O modelo escolhido é o Diagrama de Entidade-Relacionamento (também conhecido por DER), que foi desenvolvido para permitir a representação das estruturas de dados de uma forma mais natural e aproximada do mundo real de negócios. Este modelo propõe a visualização da perspectiva em estudo sobre três pontos de vista: entidade, atributo e relacionamento.

Uma entidade representa um objecto que pode ser identificado de uma forma unívoca em relação a todos os outros objectos, podendo ser uma coisa concreta e física (como por exemplo um produto jornal) ou uma coisa mais abstracta (como uma assinatura).

Um atributo é uma propriedade que descreve uma entidade, podendo ser definido como simples (isto é, não pode ser dividido em partes, como por exemplo uma localidade), ou como composto (isto é, pode ser dividido em atributos menores como acontece num endereço em que se pode subdividir em artéria, número de polícia e andar).

O relacionamento é aquilo que liga as várias entidades. Existem vários tipos de relacionamento possíveis entre duas entidades:

- Um para Um (representado por 1:1): onde uma ocorrência de uma entidade está associada apenas a uma e só uma ocorrência de uma outra entidade. É um tipo de relação de rara existência na realidade do mundo dos negócios;
- Um para Muitos (representado por 1:M ou M:1): onde uma ocorrência de uma entidade está associada a várias ocorrências de uma outra entidade. É um tipo de relação comum no mundo dos negócios;
- Muitos para Muitos (representado por N:M): onde várias ocorrências de uma entidade estão associadas a várias ocorrências de uma entidade. Esta relação deve ser transformada, através da criação de uma entidade intermediária

chamada entidade associativa, em duas relações de tipo Um para Muitos entre cada entidade original e a entidade intermediária.

A variante representada neste trabalho é conhecida como Martin (em homenagem em seu precursor o britânico James Martin. Pode ser reconhecida como *Crow's Foot* devido à representação de um pé de corvo no relacionamento de Muitos) que normalmente é mais utilizada no Reino Unido, enquanto a versão mais reconhecida nos Estados Unidos da América é versão proposta pelo Prof. Peter Chen. Na **Figura 8**, são representadas as diferentes variantes apresentadas nas situações possíveis de 1:1, M:1 e N:M.



Figura 8. Representação gráfica das diferenças das variantes de Chen e Martin

O diagrama apresentado (**Figura 9**) já contém todas as entidades, atributos e relações consideradas indispensáveis para o funcionamento do sistema.

Há vários aspectos apresentados no diagrama que merecem relevância.

O primeiro diz respeito à existência de uma entidade *CodigoPostal*. Esta entidade representa os códigos postais e artérias nacionais, que são disponibilizados pelos Correios de Portugal ([www.ctt.pt](http://www.ctt.pt)). Deverá servir o seguinte propósito:

- No processo de preenchimento de endereço para uma entidade/assinatura, será solicitado o País (estes estão codificados conforme a norma internacional ISO-3166) e no caso de ser escolhida a opção 'Portugal', será solicitada ao utilizador a escolha de uma artéria. Após o utilizador efectuar a escolha, alguns campos do endereço da entidade/assinatura serão automaticamente preenchidos (como

Artéria, Localidade e Código Postal). Um dos maiores problemas que as bases de dados enfrentam, se não mesmo o maior, é a descrição de dados que os utilizadores colocam na mesma. No registo de um endereço, é possível uma artéria ser escrita de forma diferente cada vez que é referida. Dessa maneira, é possível que estes dados possam induzir em erro outros utilizadores, seja na atribuição de rotas, seja na consulta de dados estatísticos. Como os Correios de Portugal detêm a base de dados mais completa em termos de artérias nacionais, e para prevenir a multi-digitação de uma artéria que se pretende que seja preenchida de maneira única e correcta, optou-se por este tipo de solução.

Outro aspecto a ser notado é que os requisitos referidos anteriormente como responsabilidade do sistema (Alterações de Assinaturas e Histórico de Entregas) foram constituídos como entidades (*HistoricoEntidade*, *HistoricoAssinaturas* e *Distribuicao*) que serão registados automaticamente, mediante as condições propostas e explicadas anteriormente aquando da constituição dos requisitos.

Uma terceira situação a apontar e que se encontra no modelo, é a existência de *Produtos* e de *Publicacoes*. A impressão de uma publicação, seja jornal ou revista, pode não ser efectuada em apenas um local. Pode ocorrer também o caso de uma publicação ter edições diferentes na mesma data ficando cada uma delas a cobrir zonas específicas. Pegando num caso típico como é o *Jornal de Notícias*, o mesmo tem três edições diferenciadas diariamente. Apesar das notícias em todas elas serem as mesmas, diferem na posição e nos destaques, chegando mesmo a existir capas diferentes entre as várias edições. No caso referido, existe também diferenças nas páginas de classificados. No entanto, e apesar das diferenças, não deixam de ser reconhecidas todas elas como uma mesma publicação: *Jornal de Notícias*. Mas exemplificando, como seria possível reconhecer que um assinante de Lisboa possa querer receber a edição que sai no Norte? Ou então como fazer caso um assinante de Braga queira a edição Sul? Para dar uma resposta a este tipo de solicitações, porque elas realmente existem e fazem parte do dia-a-dia neste negócio, faz sentido a criação de uma entidade *Produto* que por sua vez responde a uma entidade *Publicação*. Deste modo, e pegando no caso referido, será possível classificar uma publicação *Jornal de Notícias* que por sua vez contempla três

tipos de produtos diferentes: *Jornal de Notícias Norte*, *Jornal de Notícias Porto* e *Jornal de Notícias Sul*. Numa assinatura, serão os produtos que terão de ser definidos, para permitir visualizar se um assinante quer receber uma edição Norte e/ou uma edição Porto e/ou outra edição Sul.

Esta é uma situação típica existente na Imprensa nacional, sendo mais visível na gestão de produtos diários (onde para se dar resposta eficaz, normalmente opta-se por imprimir em um par de gráficas) e que na amostra das aplicações de mercado consultadas, a resposta que fornecem a este tipo de situações é normalmente pouca esclarecida como a criação de publicações diferentes.

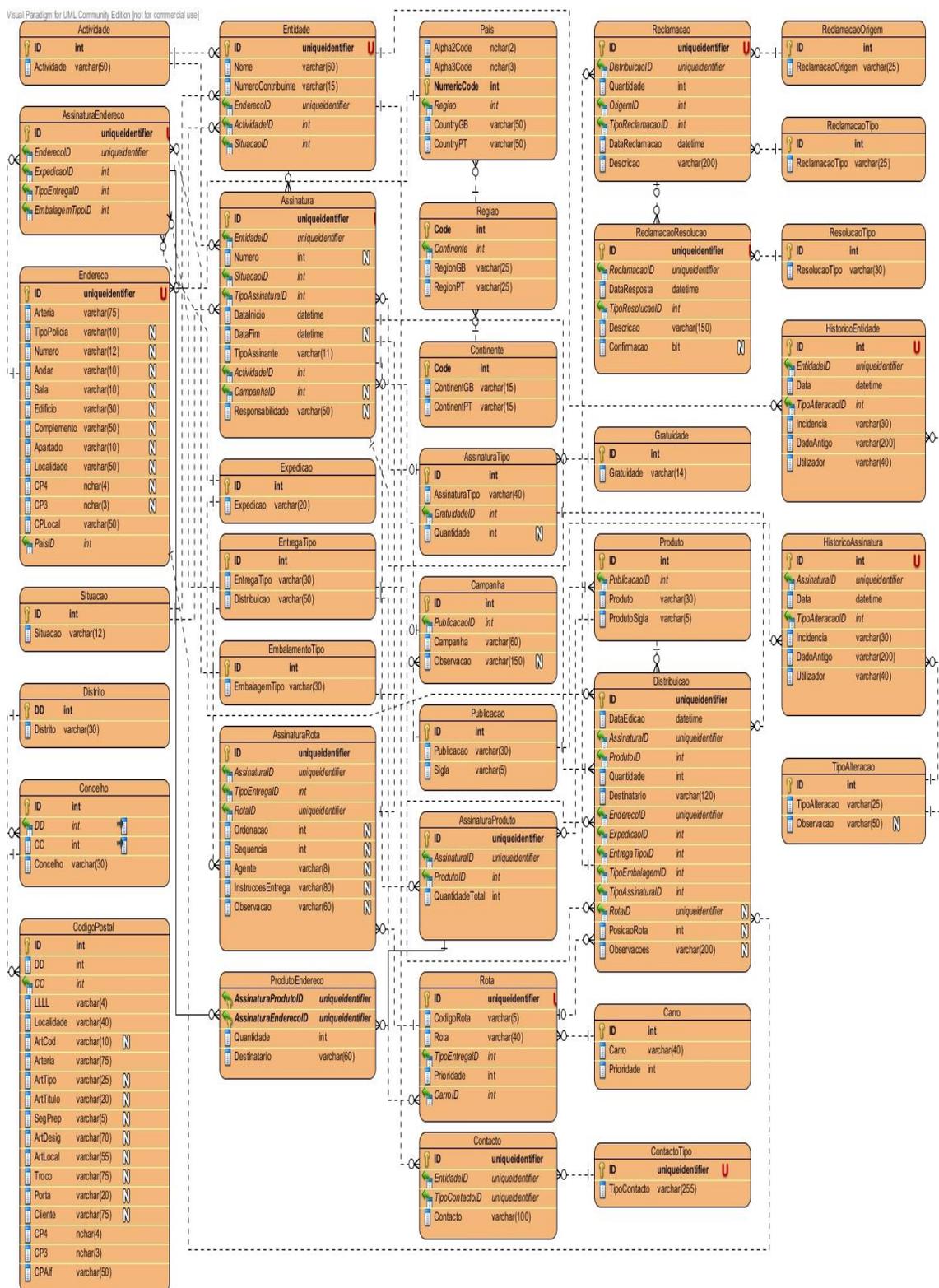


Figura 9. Diagrama de Entidade-Relacionamento completo



### 3. Implementação

Terminado a fase de planeamento e desenho da base de dados, segue-se a implementação num sistema que permita armazenar, gerir e manter os dados de um modo estruturado. Esse sistema é uma ferramenta de software conhecida como Sistema de Gestão de Base de Dados (conhecido pelo conjunto de siglas SGBD).

O primeiro passo é a escolha do SGBD, já que terá implicações na forma como o processo de implementação se irá desenrolar.

### 3.1. Seleção do Sistema de Gestão de Base de Dados

Um SGBD “é uma colecção de programas que permitem ao utilizador definir, construir e manipular uma ou mais bases de dados para as mais diversas finalidades” (Novelli, 2006). Pode-se definir também como um serviço que faz a ligação entre o utilizador e os dados, sendo a única entidade que tem a capacidade de aceder e de manipular os dados directamente. Tem como objectivo o fornecimento de um ambiente onde se pode aceder e armazenar informações na base de dados de modo fiável e eficiente.

Um SGBD deve ter as seguintes características elementares:

- ser eficiente (capacidade de acesso, processamento e alterar volumes de dados eficientemente);
- ser robusto (capacidade de manutenção dos dados de forma consistente, mesmo perante situações de existência de falhas de hardware e/ou software);
- ser persistente (capacidade de manutenção dos dados durante um período longo de tempo, independentemente do tipo e número de aplicações que a eles podem aceder);
- ser controlador (capacidade de controlo de acesso dos utilizadores aos dados, validando apenas aqueles que se encontram autorizados).

Existem no mercado vários SGBD disponíveis e que têm a possibilidade de suportar um volume de dados elevados. Apresenta-se de seguida uma breve análise dos mais conhecidos.

#### 3.1.2. Microsoft SQL Server

O *SQL Server*<sup>3</sup> é um SGBD relacional propriedade da Microsoft. Inicialmente, foi produzido em parceria com a empresa *Sybase*, mas após o término dessa parceria, a Microsoft não deixou de aperfeiçoar o produto. Suporta o tipo de linguagem SQL (através do *Transact SQL* ou T-SQL) e trabalha apenas sobre a plataforma Windows.

---

<sup>3</sup> Disponível em <http://www.microsoft.com/sqlserver/en/us/default.aspx>

Algumas das funcionalidades que o *SQL Server* detém são o de possuir um motor de base de dados relacional com alta fiabilidade e a possibilidade de replicar os dados, facilitando este processo em aplicações distribuídas, em serviços de análise e em relatórios.

A versão 2008 tem algumas características muito interessantes no que diz respeito à administração de dados, como a possibilidade de se efectuar um *mirroring* de uma base de dados, a possibilidade de serem efectuadas recuperações mais rápidas, de criar uma vista estática (designada por *snapshot*) de uma determinada base de dados, de efectuar replicações entre tabelas em pontos diferentes e de ter umas ferramentas de gestão próprias bastante interessantes e intuitivas para o Administrador de Base de Dados.

### 3.1.2. Oracle

*Oracle*<sup>4</sup> é um SGBD relacional pertencente à *Oracle Corporation*. Começou a ser produzido no final dos anos 70, por Larry Ellison, Bob Miner e Ed Coates, que perceberam a potencialidade de criação de um produto de base de dados relacional e que pudesse ser comercializado. Também tem suporte para a linguagem SQL (*PL-SQL* que é considerada mais forte do que a *T-SQL* utilizada pela Microsoft) e é compatível com uma série de plataformas (Windows, MAC OS, Linux).

A *Oracle* possui um motor de base de dados relacional com alta fiabilidade e com controlo absoluto ao nível transaccional (ou seja, nenhuma transacção será permanente até se dar ordem para tal).

A versão 11g tem algumas características muito interessantes no que diz respeito à administração de dados, como a possibilidade de serem efectuadas recuperações rápidas de uma base de dados, o facto de avisar automaticamente o Administrador da Base de Dados de erros considerados críticos, de criar uma vista estática (designada por *snapshot*) de uma determinada base de dados, de efectuar a gestão automática da

---

<sup>4</sup> Disponível em <http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html>

memória e comportar uma série de ferramentas de gestão próprias que auxiliam o Administrador de Base de Dados.

### 3.1.3. MySQL

*MySQL*<sup>5</sup> é um SGBD relacional e é o mais popular software livre disponível, sendo neste momento parte integrante do catálogo da *Sun Microsystems*. Foi criado nos anos 90 por David Axmark, Allan Larsson e Michael Widenius. É um SGBD em crescendo, onde cada vez mais instituições utilizam este produto, como são os casos da NASA, Sony, Nokia, Lufthansa, Google, Cisco Systems, entre outros. Utiliza a linguagem SQL e pode ser utilizado em várias plataformas (Windows, Mac OS, Linux).

Algumas das funcionalidades que o *MySQL* possui são o de deter um motor de base de dados relacional com alta fiabilidade, a possibilidade de ter motores de bases de dados independentes, permitindo escolher qual deles pode ser mais efectivo junto de determinada tabela e a possibilidade de juntar múltiplas transacções de múltiplas conexões para se poder aumentar o número de *commits*<sup>6</sup> por segundo.

A versão 5.1 tem características muito interessantes no que diz respeito à administração de dados, como a possibilidade de efectuar replicações de uma base de dados, a possibilidade de serem efectuadas recuperações rápidas e de poder comportar uma série de ferramentas de gestão próprias que serve de auxílio para o Administrador de Base de Dados.

Na **Figura 11**, são apresentadas as principais semelhanças e diferenças entre os três produtos:

---

<sup>5</sup> Disponível em <http://www.mysql.com>

<sup>6</sup> Designa-se por *commit* o processo torna permanentes um conjunto de mudanças que eram consideradas experimentais. O contrário de *commit* é *rollback*.

	Licença	Sistemas Operativos	Transacção ACID	Integridade Referencial	Transacções	Unicode
SQL Server	Proprietário	Windows	Sim	Sim	Sim	Sim
Oracle	Proprietário	Windows, MAC OS, Linux, UNIX, IBM z/OS	Sim	Sim	Sim	Sim
MySQL	GNU (é proprietário na versão Enterprise)	Windows, MAC OS, Linux, UNIX, AmigaOS, Symbian, IBM z/OS	Sim	Sim	Sim	Parcialmente

	Tamanho Máximo da Base de Dados	Tamanho Máxima de Tabela	Tamanho Máximo de Linha	Tamano Máximo de Colunas	Tamanho Máximo de CHAR	Tamanho Máximo de NUMBER	Tamanho Máximo de Nome de Campo
SQL Server	524,258 TB	524,258 TB	Ilimitado	30000	2 GB	126 bits	128
Oracle	Ilimitado	4 GB	8 Kb	1000	4000 B	126 bits	30
MySQL	Ilimitado	256 TB ou 64 TB	64 Kb	4096	64 Kb	64 bits	64

	Tabelas Temporárias	UNION	INTERSECT	EXCEPT	INNER/OUTER JOIN	MERGE JOIN
SQL Server	Sim	Sim	Sim	Sim	Sim	Sim
Oracle	Sim	Sim	Sim	Sim	Sim	Sim
MySQL	Sim	Sim	Não	Não	Sim	Sim

	Cursor Mechanism	Trigger	Function	Procedure	Password Rules	Certificação de Segurança
SQL Server	Sim	Sim	Sim	Sim	Sim	Sim
Oracle	Sim	Sim	Sim	Sim	Sim	Sim
MySQL	Sim	Sim	Sim	Sim	Não	Não

Figura 11. Comparação entre os SGBD's apresentados

Em termos de suporte, qualquer dos três oferece respostas a possíveis dúvidas e pedidos de ajuda (mesmo que sejam personalizados).

Os custos de licenças, são variados entre eles. Não foi possível obter um valor exacto da versão *MySQL Enterprise*, mas em relação aos produtos apresentados pela *Microsoft* e pela *Oracle*, tendencialmente esta última apresenta custos mais elevados em relação à sua concorrente.

No processo de escolha do SGBD, foi dada atenção às funcionalidades que cada solução dispõe, ao suporte que cada empresa oferece e a todos os custos estimados.

Após ponderação, e considerando que as três opções disponibilizam as funcionalidades necessárias e que o suporte prestado pelas empresas representantes seria o adequado às necessidades, a escolha recaiu sobre o produto da *Microsoft*, atendendo a dois factores principais:

- Uma vez que já foram adquiridas anteriormente algumas licenças do produto, uma parte significativa do investimento está feita;
- Os utilizadores estão familiarizados com as funcionalidades do *Microsoft SQL Server*, o que permitirá poupar nos custos de formação.

## 3.2. Implementação

Escolhido o SGBD, será necessário passar à fase de implementação propriamente dita: criar a base de dados; criar as tabelas e as relações entre elas; criar outros objectos de códigos, definir o modelo de segurança.

### 3.2.1. Criação da base de dados, das tabelas e das relações

A criação da base de dados no *SQL Server* bem como das tabelas definidas no Diagrama Entidade-Relacionamento pode ser realizada de diferentes maneiras:

- Através da aplicação de apoio *Server Management Studio*, que permite criar visualmente e de maneira simplificada uma base de dados e criar as tabelas respectivas;
- Através da execução de instruções de código SQL;
- Através de software para o efeito (como exemplo, o *Visual Paradigm* na versão *Enterprise* tem uma opção que permite gerar o código SQL tendo em conta as especificações processadas no desenho do Diagrama de Entidade-Relacionamento).

Nesta abordagem, a primeira opção é a escolhida. É possível criar a base de dados através da opção ‘*New Database*’ e preencher as informações conforme exemplificado na figura seguinte (**Figura 12**).

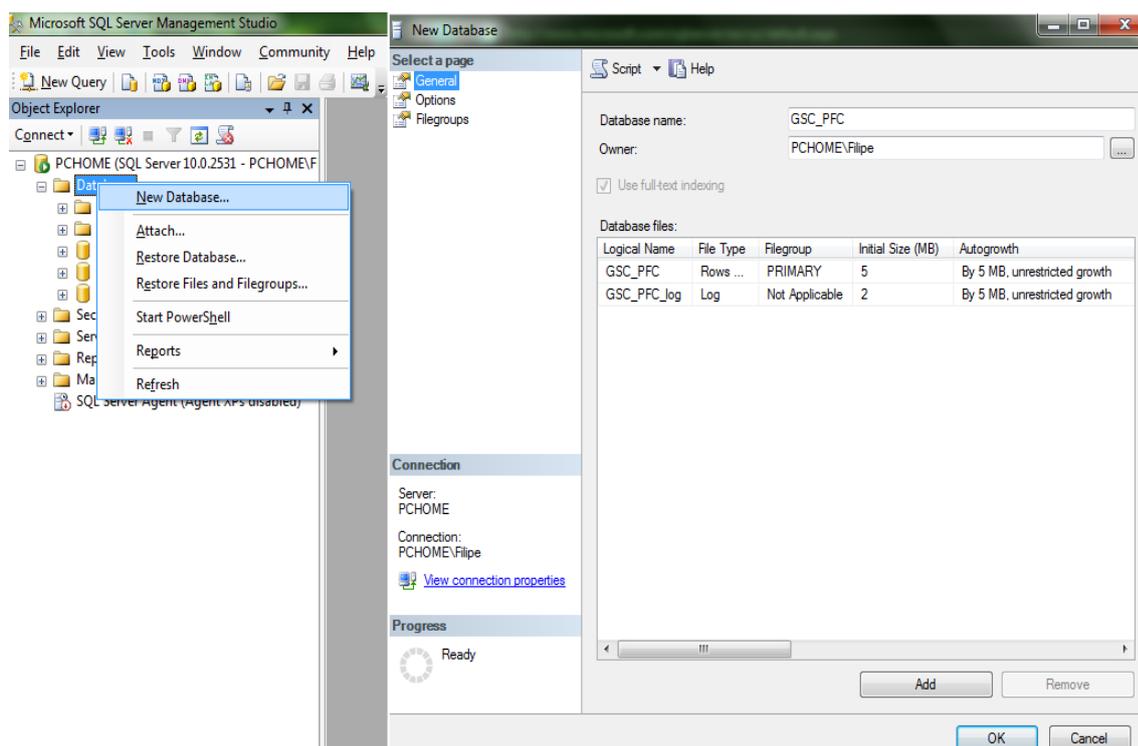


Figura 12. Criação de uma nova base de dados em ambiente GUI no SQL Server 2008

Uma opção que se recomenda é que no final do processo e antes de se carregar na opção OK, seja activada a opção ‘*Script Action To File*’, que guarda num ficheiro as instruções das opções escolhidas para linguagem de script<sup>7</sup>, o que pode ser útil para futuras utilizações.

O passo seguinte passa pela criação das tabelas necessárias. No caso, vamos dar um exemplo da criação de duas tabelas (*Publicacao* e *Produto*), como exemplificado na **Figura 13**.

<sup>7</sup> Linguagem de script é um termo utilizado para designar uma sequência de comandos e tarefas a serem executadas no interior de um programa ou noutra linguagem de programação, mas pode não se restringir a esse ambiente específico.

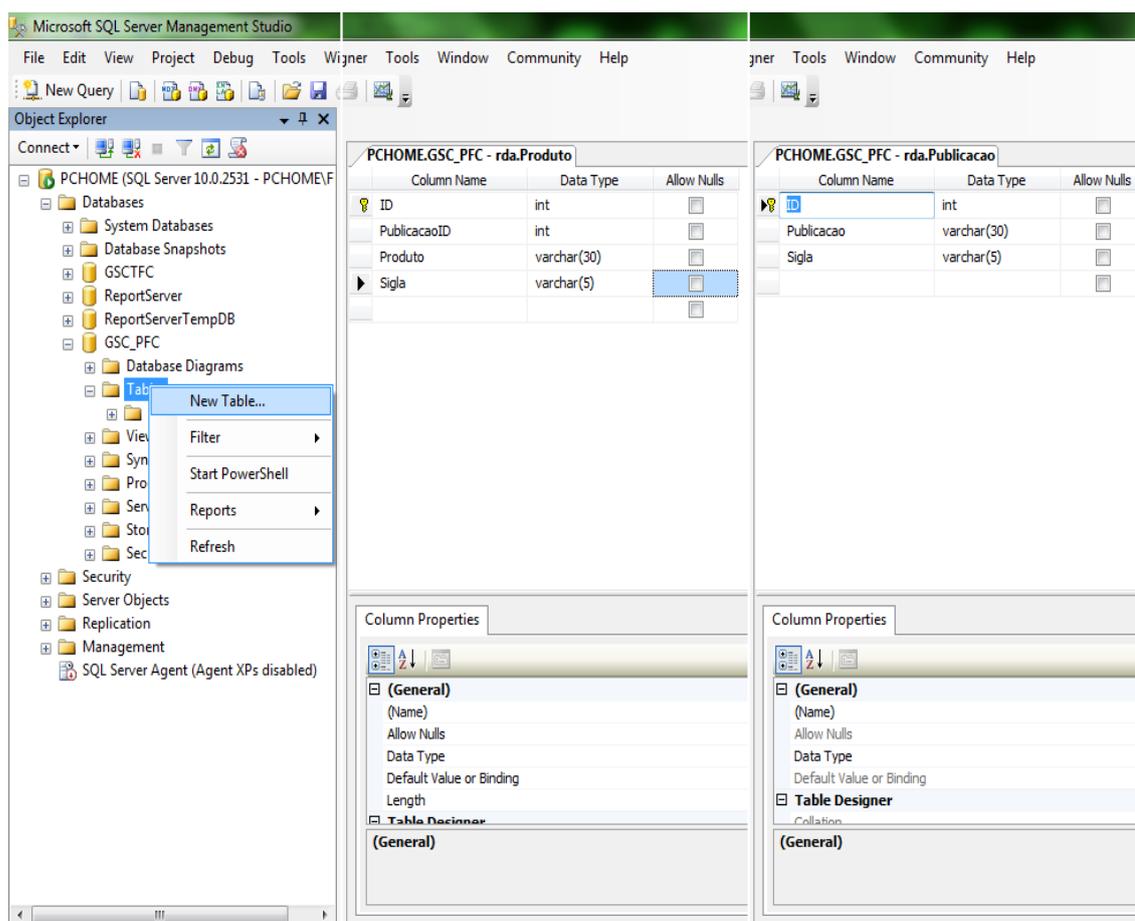


Figura 13. Criação de tabelas através do *Server Management Studio*

E pegando no exemplo das duas tabelas, vamos criar a relação entre as duas tabelas entre os atributos **Publicação.ID** e **Produto.PublicacaoID**.

Selecciona-se a tabela onde irá existir a chave estrangeira (neste caso, a tabela *Produto*) e no modo de desenho, escolhe-se a opção '*Relationships*'. Selecciona-se quais os dois atributos que se irão ligar entre si. Desta maneira, é criada a chave estrangeira no atributo **Produto.PublicacaoID** (Figura 14).

Após serem criadas todas as tabelas e respectivas relações entre elas, a base de dados fica praticamente criada.

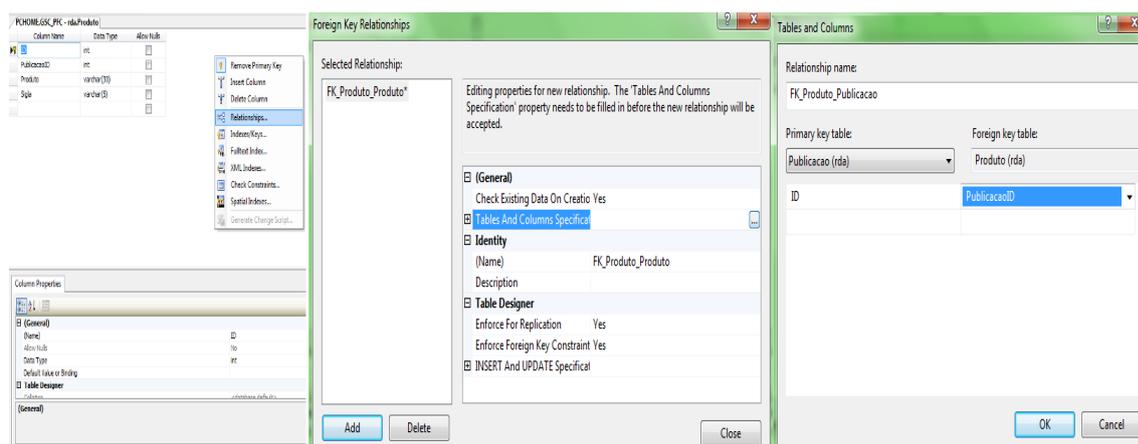


Figura 14. Criação de uma relação entre tabelas

Para alimentar automaticamente as tabelas de histórico (*HistoricoAssinatura* e *HistoricoEntidade*), haverá a necessidade de se criar *triggers*. Um *trigger* não é mais do que um procedimento que é executado automaticamente quando existe uma tentativa de modificação dos dados de um objecto, através dos comandos INSERT, DELETE ou UPDATE. Um *trigger* é tratado com uma única transacção, podendo ser revertido em caso de necessidade.

### 3.2.2. Segurança

Ao nível da segurança, foram criados *database roles* para simplificar o processo de fornecer ou retirar permissões a grupos de utilizadores.

No caso em estudo, foram criados os seguintes *database roles*:

- Distribuiçao: pertencente aos elementos que compõem o Departamento de Distribuição;
- GesAssin: pertencente aos indivíduos que compõem o Departamento de Assinaturas;
- RespAssin: pertencente ao Responsável do Departamento de Assinaturas.

Adicionalmente, foram definidos novos *schemas*. Um *schema* é uma espécie de contentor que pode abrigar vários objectos da base de dados. Ao atribuir a propriedade de um *schema* a um objecto como por exemplo uma tabela, esta automaticamente passa a deter as propriedades que o *schema* em questão possui. A vantagem é que se eliminarmos por exemplo, um utilizador, não será necessário alterar a propriedade dos objectos, porque estes estão primeiramente ligados a um *schema*. Quando uma base de dados é criada, todos os objectos ficam registados sobre um *schema* *dbo*, que existe por defeito.

Tendo em conta o problema, foram criados alguns *schemas* e dadas permissões GRANT sobre estes, tendo em conta os *database roles* criados, conforme **Figura 15**:

<i>Role / Schema</i>	<i>dbo</i>	<i>gda</i>	<i>rda</i>	<i>gdd</i>
<b>Role Distribuicao</b>	INSERT SELECT UPDATE	SELECT	SELECT	INSERT SELECT UPDATE
<b>Role GesAssin</b>	INSERT SELECT UPDATE	INSERT SELECT UPDATE	SELECT	SELECT
<b>Role RespAssin</b>	INSERT SELECT UPDATE	INSERT SELECT UPDATE	INSERT SELECT UPDATE	SELECT

**Figura 15. Permissões dos roles nos diversos *schemas***

Fez-se proveito do *schema* *dbo*, dando sobre este as permissões de inserir e actualizar registos, bem como a opção de se seleccionar dados sobre os objectos ligados ao *schema*. Os restantes foram criados para diferenciar as permissões sobre a distribuição (nomeadamente, tudo o que seja ao nível da distribuição como as rotas e entregas) e sobre as assinaturas. Previne-se deste modo que os utilizadores da distribuição não possam interferir (mas sim, apenas visualizar) nos dados respeitantes à gestão de assinaturas e vice-versa. Com os *schemas* criados e com as permissões definidas para cada um deles, escolhe-se sobre quais tabelas estes irão afectar (**Figura 16**).

Schema dbo	Schema dga	Schema rda	Schema gdd
Distribuicao	Assinatura	Actividade	AssinaturaRota
HistoricoAlteracao	AssinaturaEndereco	AssinaturaTipo	Carro
HistoricoAssinatura	AssinaturaProduto	Campanha	EntregaTipo
HistoricoEntidade	AssinaturaQuantidade	CodigoPostal	Rota
ResolucaoReclamacao	Contacto	Concelho	
	Entidade	ContactoTipo	
	Reclamacao	Continente	
		Distrito	
		EmbalagemTipo	
		Embalamento	
		Expedicao	
		Gratuidade	
		Pais	
		Produto	
		Publicacao	
		ReclamacaoOrigem	
		ReclamacaoTipo	
		ResolucaoTipo	
		Regiao	
		Situacao	

Figura 16. Tabelas e respectivos *schemas* atribuídos

O modo de autenticação no acesso ao servidor pode ser efectuado de duas maneiras: autenticação Windows e autenticação mista.

A primeira indica que a responsabilidade de verificar a segurança do login no acesso ao servidor é do *Windows Server*. Ou seja, cada vez que um utilizador faz login e é validado pelo *Windows Server*, o *SQL Server* recolhe esta informação e passa a permitir-lhe o acesso também. Tem como condição essencial que cada utilizador tenha uma conta criada no sistema operativo *Windows*.

Com o modo de autenticação misto, é possível aceder ao servidor através do *Windows* ou com autenticação no próprio *SQL*. Neste modo, basta que um utilizador tenha um login registado no *SQL Server* que ser-lhe-á fornecido o acesso. Este método é menos seguro que o anterior, mas facilita o acesso a possíveis utilizadores que não pertençam ao domínio *Windows*.

Neste caso de estudo, a opção recai sobre a autenticação Windows, visto que todos os utilizadores têm de estar registados no *Windows Server* para poderem aceder à base de dados.

### 3.2.3. Redundância

O sistema desenvolvido tem de cumprir ainda um requisito adicional no seu funcionamento: garantir a continuidade do negócio.

Para que este requisito seja cumprido eficazmente, foi decidido avançar-se com a implementação de *Database Mirroring*. A configuração adoptada consiste em dois servidores com o *SQL Server* implementado sobre o sistema operativo *Windows Server* onde um será a máquina principal e a outra máquina se comportará como auxiliar.

O *Database Mirroring* é uma funcionalidade que permite efectuar a cópia da base de dados, através da transferência dos registos de log do servidor principal para o servidor secundário, sendo utilizado para se minimizar a perda de dados em caso de falha ocorrida no servidor principal, visto que em condições normais, todas as transacções ocorridas, incluindo a última ocorrida, estarão espelhadas no servidor secundário. Apenas o servidor principal está disponível e acessível aos utilizadores, mas com a implementação do *mirroring* da base de dados, em caso de falha do servidor principal, o servidor secundário pode assumir essa função, visto ter os dados replicados.

Numa primeira fase, e enquanto a aplicação na sua totalidade não estiver terminada, não existirá uma máquina *witness*<sup>8</sup> (máquina que se encontra entre os dois extremos da operação de *mirroring*) que possa acompanhar a comunicação entre os dois servidores (**Figura 17**). Tal implica que em caso de falha do servidor principal, a alteração para que o servidor secundário assuma a posição primária, denominada como operação de *role switching*, tem de ser efectuada manualmente pelo Administrador de Base de Dados. No entanto, a possível existência no futuro de uma máquina *witness* não está colocado de parte.

---

<sup>8</sup> Esta máquina *witness* pode estar implementada com uma versão Express do SQL Server 2008.



Figura 17. Database Mirroring com máquina witness

(Fonte: [http://media.techtarget.com/digitalguide/images/Misc/Database\\_Mirroring.jpg](http://media.techtarget.com/digitalguide/images/Misc/Database_Mirroring.jpg))

### 3.2.4. Replicação de dados

Um das questões mais sensíveis a ser abordadas e discutidas diz respeito à replicação de dados. Como já foi referido, todo este projecto diz respeito à agregação numa só aplicação de várias funcionalidades que normalmente estariam espalhadas por várias aplicações. Na primeira etapa deste processo e à qual este trabalho se restringe, o foco assenta na gestão de assinaturas e na distribuição das mesmas.

Fisicamente, o sistema será implementado em:

- Quatro locais distanciados (**Figura 18**):
  - Lisboa (com serviço de gestão de assinaturas de publicações) representado pela letra **A**;
  - Sintra (com serviço de distribuição), representado pela letra **B**;

- Porto (com serviço de gestão de assinaturas de publicações e serviço de distribuição), representado pela letra **C**;
- Fundão (com serviço de gestão de assinaturas de publicações), representado pela letra **D**.



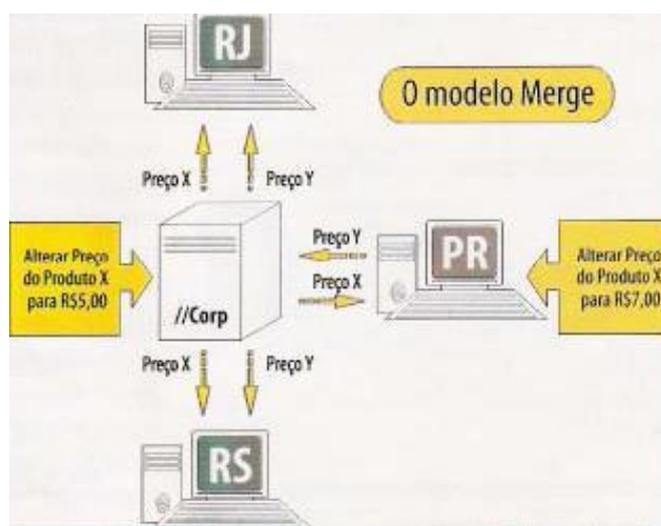
**Figura 18. Quatro locais para implementação**

Os servidores dos quatro locais assinalados terão de estar interligados sobre a mesma base de dados, com acesso e possibilidade de consulta e alteração dos dados. Para evitar que exista uma sobrecarga na rede de acesso entre os vários pontos ao servidor principal e evitar também que em caso de falhas no acesso ao ponto onde o servidor está instalado (como por exemplo, falhas de electricidade, falhas de hardware, etc.), os restantes locais deixem de ter acesso aos dados, foi tomada a decisão de ser criada replicação entre os vários pontos. Isto permite que exista uma distribuição e cópia de dados em modo automático sem que exista intervenção dos utilizadores.

Como acima referido, os quatro locais terão possibilidade de efectuar actualizações nos dados e visto que todos eles têm de ter directa intervenção no processamento dos dados das assinaturas, o modelo que se considera como o mais adequado é aquele que possa permitir que os vários pontos de replicação entrem em contacto uns com os outros, informando as actualizações que ocorram sobre cada um deles localmente. O *SQL Server* permite que isto seja possível, através da funcionalidade de replicação por *merge* e pela topologia de servidores Publicadores/Subscritores.

A topologia Publicador/Subscritor não é mais do que um modelo utilizado quando se deseja manter vários intervenientes actualizados uns com os outros. Significa que uma máquina pode publicar alterações efectuadas localmente e propagá-las através da rede aos subscritores, mas como também é subscritora, a mesma máquina recebe as alterações que foram efectuadas localmente por outras e entretanto propagadas na rede.

O tipo de replicação por *merge* (**Figura 19**) permite actualizar as bases de dados locais, replicando as alterações enviadas pela máquina publicadora. Um dos problemas mais usuais neste modelo é a existência de conflitos quando se dá o *merge* das alterações, mas estes podem ser definidos automaticamente no agente com a configuração de um modelo de resolução de conflitos.



**Figura 19. Tipo de replicação modelo *merge***

(Fonte: <http://www.devmedia.com.br/Imagens/gold/SQL/01/artigo10SQL01/image004.jpg>)

Salienta-se que o próprio *SQL Server* tem uma ferramenta que ajuda a verificar se o processo de replicação está a correr efectivamente como planeado e desejado. É a *Replication Monitor*, que como o nome indica, analisa o estado da replicação, podendo inclusivamente efectuar ajustes no processo de publicação e subscrição.

### 3.3 Considerações sobre o modelo

Independentemente do trabalho abordar uma área de negócio específica (a gestão e distribuição de assinaturas), os conceitos propostos podem ser aplicados também noutras áreas, com algumas alterações.

O facto de se ter definido algumas entidades como tendo um maior peso, como é o caso da *Entidade* que se sobrepõe à *Assinaturas* ou da *Publicações* que se sobrepõem à *Produtos*, permite obter informações encadeadas.

Exemplificando, passa a ser possível saber quantas assinaturas determinada *Entidade* possui. Num outro tipo de informação, pode ser possível saber a quantidade total entregue de um determinado *Produto* num período, sendo possível efectuar a subdivisão por tipos de entrega, ou ainda com maior detalhe, ao nível de rotas.

Permite saber a quantidade distribuída de uma determinada publicação numa determinada data (ou num conjunto de datas), podendo ser dividida ou não pelas várias edições que possa eventualmente ter.

A existência da *Entidade* como atributo maior sobre outras como a *Assinatura*, permite que na ligação entre elas possa ser possível obter-se informações adicionais. Passa a ser possível correr análises sobre o comportamento de determinada entidade em termos de assinaturas, reclamações, distribuição e outros mais.

Transpondo para outro tipo de negócio, e salvo as diferenças que a própria arquitectura de negócios oferece, pode ser possível a uma locadora de automóveis utilizar o conceito. Onde por exemplo, a publicação poderia ser alterado para marcas de automóveis e o produto para modelos. Em termos estatísticos, seria possível identificar a quantidade utilizada de determinada marca e/ou modelo específico.

Os conceitos utilizados na implementação para um SGBD são passíveis de serem utilizados em qualquer área de negócio, sendo um modelo proposto para optimização do funcionamento, acesso e protecção dos dados.

## 4. Conclusões e o futuro

O projecto do qual o trabalho apresentado faz parte, na sua globalidade vai ser extenso. Existe a noção por parte do autor deste trabalho que pode haver melhoramentos que tenham de ser efectuados, sempre na procura de uma solução mais equilibrada. Poderão ocorrer alterações futuras em alguns dos modelos propostos, mas as ideias presentes na concepção deste trabalho são as defendidas pelo seu autor. E em como qualquer projecto que envolva uma base de dados, seja no desenho ou na implementação, é natural que existam outras visões de se encarar o problema e outras possibilidades de se efectuar este trabalho.

Poderão existir alterações futuras em alguns atributos que compõem as tabelas criadas ou mesmo a concepção de novas tabelas de modo a obter-se cada vez mais informação, como é do interesse dos grupos editoriais.

Os objectivos primários neste trabalho foram alcançados, desde a concepção de requisitos e a modelação através da linguagem UML, passando pela concepção e implementação num Sistema de Gestão de Base de Dados entretanto escolhido (neste caso, *Microsoft SQL Server 2008*) bem como a definição neste de algumas funcionalidades que se consideram importantes para o bom funcionamento da base de dados.

Considero que um dos aspectos fulcrais neste trabalho foi precisamente a abordagem que teve de ser feita ao nível do levantamento de requisitos e o seu desenho nos diagramas propostos utilizando as concepções fornecidas pela UML. Foi precisamente esta abordagem que permitiu que algumas ideias se tornassem mais claras, contribuindo desta forma para atingir os objectivos propostos.

Todo o caminho percorrido no levantamento de requisitos e sua transposição para UML permitiu que a concepção da base de dados e a implementação no SGBD se tornassem uma sequência natural ao longo deste trabalho.

## Bibliografia

- Caldeira, C. P. (2010). *Introdução ao Modelo de Dados Relacional*. [Internet] Disponível em <[http://www.portalwebmarketing.com/Tecnologia/mdr\\_introducao\\_modelo\\_dados\\_relacional\\_indice/tabid/621/Default.aspx](http://www.portalwebmarketing.com/Tecnologia/mdr_introducao_modelo_dados_relacional_indice/tabid/621/Default.aspx)> [Consult. 16 de Novembro 2010]
- Damas, L. (2005). *SQL Structured Query Language*, 12<sup>a</sup> ed. FCA: Lisboa.
- Fisher, P. (2007). *Modelo Entidade Relacionamento*. [Internet] Disponível em <[http://chasqueweb.ufrgs.br/~paul.fisher/apostilas/basdad/bd\\_mod\\_er.htm](http://chasqueweb.ufrgs.br/~paul.fisher/apostilas/basdad/bd_mod_er.htm)> [Consult. 12 de Novembro 2010].
- Harkins, S. (2004). *Understanding roles in SQL Server security*. [Internet] Disponível em <[http://articles.techrepublic.com.com/5100-10878\\_11-1061781.html](http://articles.techrepublic.com.com/5100-10878_11-1061781.html)> [Consult. 10 de Novembro 2010].
- Macoratti, J. C. (2007). *UML – Conceitos básicos II*. [Internet] Disponível em <[http://www.macoratti.net/vb\\_uml2.htm](http://www.macoratti.net/vb_uml2.htm)> [Consult. 16 de Novembro 2010].
- Magalhães, A. (2009). *SQL Server 2008 Curso completo*, FCA: Lisboa.
- Novelli, M. (2006). *SGBD – Sistema Gerenciador de Base de Dados*. [Internet] Disponível em <<http://www.plugmasters.com.br/sys/materias/108/1/SGBD---Sistema-Gerenciador-de-Banco-de-Dados>> [Consult. 16 de Novembro 2010].
- Nunes, M.; O’Neill, H. (2004). *Fundamental de UML*, 6<sup>a</sup> ed. FCA: Lisboa.
- Ramos, P.N. (2007). *Desenhar bases de dados com UML*, 2<sup>a</sup> ed. Edições Sílabo, Lda: Lisboa.
- Ribeiro, P. (2004). *Os melhores recursos para replicação*. *Revista SQL Magazine* (1) Março, pp. 44-47
- Ruest, D.; Ruest N. (2009). *Protect virtual databases through SQL Server database mirroring*. [Internet] Disponível em <<http://searchsqlserver.techtarget.com/tip/Protect->

[virtual-databases-through-SQL-Server-database-mirroring](#)> [Consult. 16 de Novembro 2010]

Shammas, G. I. J. (2005). *Modelo Entidade-Relacionamento*. [Internet] Disponível em <[www.shammas.eng.br/acad/materiais/mer.pdf](http://www.shammas.eng.br/acad/materiais/mer.pdf)> [Consult. 12 de Novembro 2010].

Wikipedia (2010). *Microsoft SQL Server*. [Internet] Disponível em <[http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)> [Consult. 12 de Novembro 2010].

Wikipedia (2010). *MySQL*. [Internet] Disponível em <<http://en.wikipedia.org/wiki/Mysql>> [Consult. 12 de Novembro de 2010].

Wikipedia (2010). *Oracle Database*. [Internet] Disponível em <[http://en.wikipedia.org/wiki/Oracle\\_Database](http://en.wikipedia.org/wiki/Oracle_Database)> [Consult. 12 de Novembro 2010].

Wikipedia (2010). *Unified Modeling Language*. [Internet] Disponível em <[http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)> [Consult. 10 de Novembro 2010].