

UML (Unified Modelling Language)

Diagrama de Classes

I	Classes	2
II	Relações	3
II.1	Associações	3
II.2	Generalização	9
III	Exemplos de Modelos	11
III.1	Tabelas de IRS	11
III.2	Exames	13
III.3	Multibanco.....	15
IV	Transposição: Modelo de Classes / Modelo Relacional	17
IV.1	Regras.....	17
IV.2	Exemplos	19



UML (Unified Modelling Language)

Diagrama de Classes

UML é uma linguagem gráfica genérica de modelação, ou seja, permite o desenho de qualquer tipo de modelos. A UML baseia-se no paradigma dos objectos, nomeadamente suporta a definição de objectos e classes. Ao longo deste texto é apenas abordada a modelação estática do UML, nomeadamente os diagramas de classes.

I Classes

Um objecto é qualquer *coisa* relevante do domínio que pretendemos modelar, que possui uma identidade, um estado (os dados associados a esses objecto) e um comportamento (aquilo que podemos fazer com o objecto). Uma classe é uma descrição de um conjunto de objectos que partilham os mesmos atributos, operações, relações e semântica. Por exemplo, o cliente ‘João Silva’ pode ser considerado um objecto relevante num sistema que pretende manipular informação referente aos clientes de uma empresa. O ‘João Silva’, para além do nome, é também caracterizado por outros atributos, nome, morada, número de contribuinte, etc.. O ‘João Silva’ possui uma identidade própria, isto é, para a empresa, ele é distinto de todos os outros clientes. Sobre o cliente ‘João Silva’ podem-se efectuar várias operações, nomeadamente emitir-lhe facturas, efectuar alterações de morada, eliminá-lo do sistema. O ‘João Silva’ relaciona-se com a empresa através, por exemplo, dos produtos ou serviços que adquire. Provavelmente existirão outros clientes na empresa, e todos eles são caracterizados pelo mesmo conjunto de atributos, pelas mesmas operações e relações e todos são distintos um dos outros. Esses diversos clientes podem ser agrupados em uma classe, a classe dos clientes da empresa. Note-se que os objectos não têm necessariamente corresponder a entidades humanas ou, mais genericamente, a entidades com representação física (por exemplo, uma factura). Um conceito abstracto, por exemplo um departamento, pode ser um objecto (caso seja relevante para o domínio em causa).

Graficamente uma classe representa-se tal como é ilustrado na Figura 1. Na secção de topo indica-se o nome da classe (que tem que ser único em um diagrama), na secção intermédia enumeram-se os atributos da classe, e na secção inferior enumeram-se

as operações que são permitidas efectuar sobre os objectos da classe. Ao longo deste texto não serão consideradas as operações (secção inferior) das classes. Como regra geral de bom senso, de modo a não sobrecarregar os diagramas, é aconselhável não enumerar o conjunto básico de operações que usualmente podem ser aplicadas a todos os objectos de todas as classes, nomeadamente, adicionar, remover e alterar. Essas operações apenas devem ser indicadas caso possuam alguma particularidade específica.

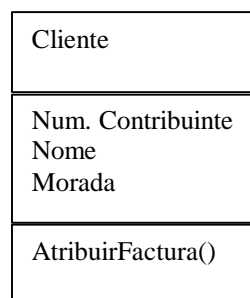


Figura 1 Representação de uma classe de objectos

II Relações

Em qualquer sistema existem objectos que se relacionam entre si. Por exemplo, se considerarmos a classe das facturas da empresa, o cliente 'João Silva' relaciona-se com as facturas a ele emitidas. A relação entre objectos é representada através das relações entre as classes. Existem diferentes tipos de relações, neste texto apenas serão abordadas algumas que se podem classificar em dois tipos, associações e generalizações.

II.1 Associações

Uma associação é uma relação que permite especificar que objectos de uma dada classe se relacionam com objectos de outra classe. Por exemplo, um cliente (ou seja, qualquer cliente) pode se relacionar com várias facturas, sendo que uma factura também se relaciona com clientes (neste caso particular, uma factura apenas se poder relacionar com um cliente). As associações são representadas através de uma semi-recta a unir as duas classes, tal como consta na Figura 2. Os intervalos de valores indicados nos extremos da semi-recta indicam a multiplicidade da associação. O intervalo $\{0 \dots *\}$ significa o intervalo entre 0 e infinito, o intervalo $\{1 \dots 1\}$ (que usualmente é abreviado para $\{1\}$),

isto é, quando o limite inferior é igual ao superior apenas se indica um dos limites) corresponde ao intervalo que apenas contém o valor 1. O exemplo da figura indica que: um cliente pode estar associado a várias (infinitas) facturas, ou a nenhuma, e uma factura necessariamente tem que estar associada a um cliente e a não mais do que um.

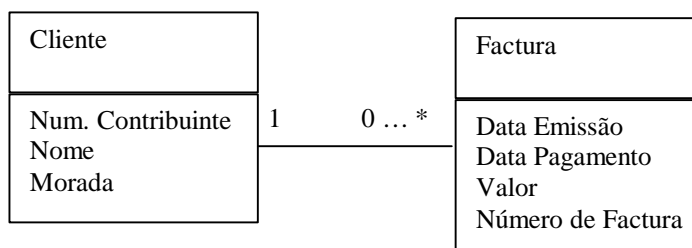


Figura 2 Representação de uma associação entre classes

Uma associação pode ser sempre lida nos dois *sentidos*. A multiplicidade diz sempre respeito ao número de objectos da classe mais próxima que podem estar associados a um objecto da classe do extremo oposto. Existem infinitas tipos de associações (todas as combinações possíveis entre os números inteiros), no entanto é usual agrupar-se as associações em três grandes grupos: “muitos para muitos”, “um para muitos” e “um para um”.

Na Figura 3 apresentam-se três exemplos que ilustram os tipos de associações. No primeiro exemplo, em um departamento podem trabalhar diversos funcionários (e admite-se a possibilidade de existirem departamentos que de momento não possuam funcionários), mas um funcionário tem que estar afecto a um e apenas um departamento. No segundo exemplo (que representa o modelo de uma outra empresa), considera-se que os funcionários podem estar afectos a vários departamentos. As associações também ter atributos, nessas situações (Classes Associativas) cria-se uma classe (unida à associação por uma semi-recta a tracejado) onde se colocam os atributos respectivos. Existem situações onde, mesmo que a associação não possua atributos próprios, é necessário criar a Classe Associativa (nomeadamente quando se pretende representar uma associação entre uma associação e uma classe. No terceiro exemplo, um recibo tem que estar obrigatoriamente associado a uma e só uma factura, enquanto que uma factura, pode ou não estar associada a um recibo (paga), mas não mais do que a um (ou seja, o sistema em causa não permite o registo de pagamentos a *prestações* da factura (vários recibos para a mesma factura).

Note-se que a obrigatoriedade de uma associação (limite inferior superior a zero) é uma restrição extremamente forte. Por exemplo, nos dois primeiros exemplos é explicitamente referido que um funcionário tem que trabalhar em um departamento. Ou seja, em termos formais e práticos, apenas pode ser *adicionado* um funcionário à classe dos funcionários se, no momento em que ele é criado, lhe for logo atribuído um departamento. Se, por exemplo, numa empresa em particular, o procedimento de registo de admissão de novos funcionários puder ser efectuado antes de eles serem atribuídos ao departamento respectivo, teria que ser criada uma outra classe para agrupar os recém contratados. Isto é, estes funcionários não partilham juntamente com os outros o mesmo tipo de associação.

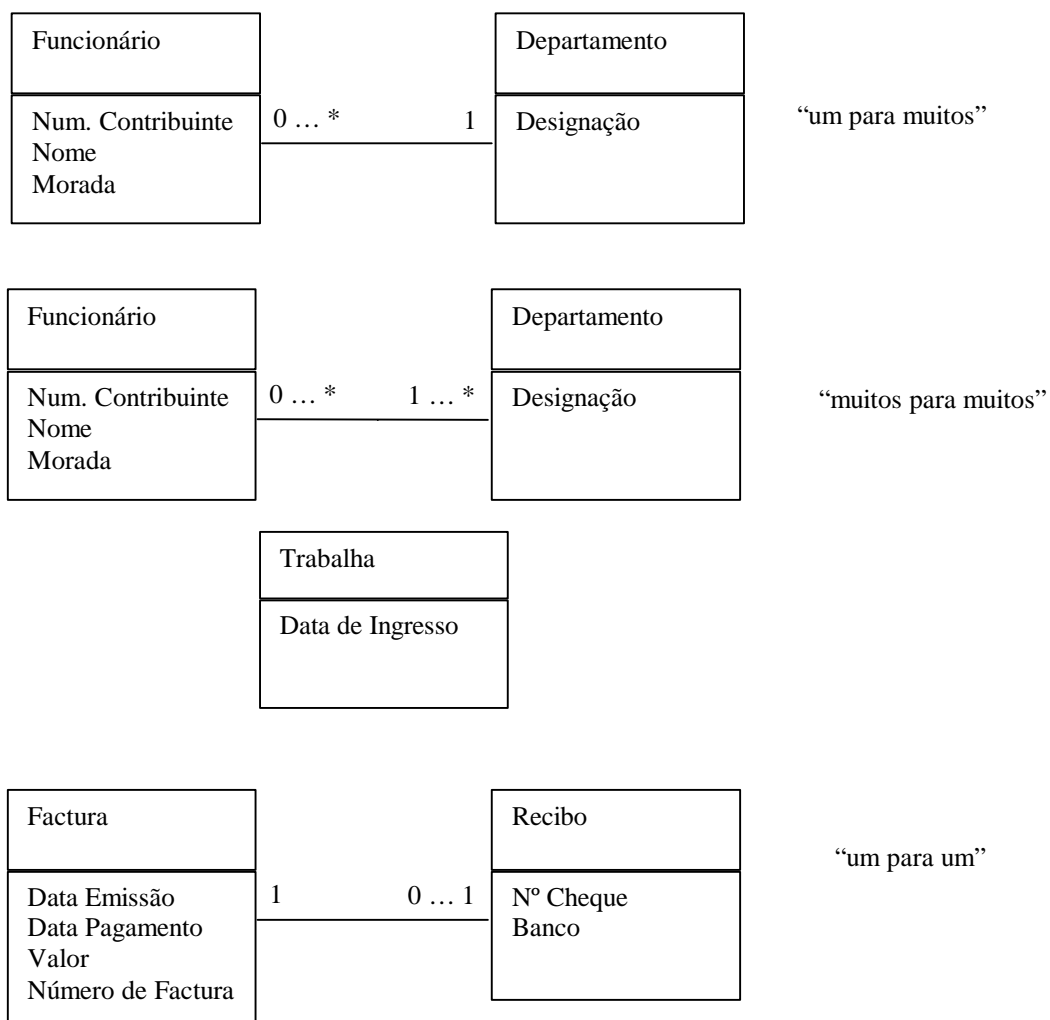


Figura 3 Exemplo de multiplicidade das associações

Na linguagem UML prevê-se a possibilidade de atribuir nomes às associações. Esses nomes (que terão que ser únicos) em um diagrama são colocados a meio da semi-recta. Por exemplo, nos dois primeiros exemplos da Figura 3, a associação poder-se-ia designar ‘Trabalham’.

Agregações

As agregações são um caso especial das associações. Elas representam a noção de composição e apenas têm sentido em associações “um para muitos” e, mais raramente, em associações “muitos para muitos”. Enquanto que nas associações anteriormente referidas não existem classes mais *importantes*, nas agregações existe uma classe (*supra classe*) que representa a agregação dos objectos da outra classe (*sub classes*). A título de exemplo, considere-se uma os itens (linhas) de uma factura. Usualmente uma factura é (ou pode ser) composta por vários itens, cada um deles diz respeito a um produto facturado. Os itens têm atributos próprios (quantidade, preço unitário, descrição do produto, etc.) e entidade própria (é possível distinguir um item de outro na mesma factura). Uma possível representação deste domínio é o diagrama da Figura 4. Uma factura possui vários itens, mas um item apenas diz respeito a uma factura.

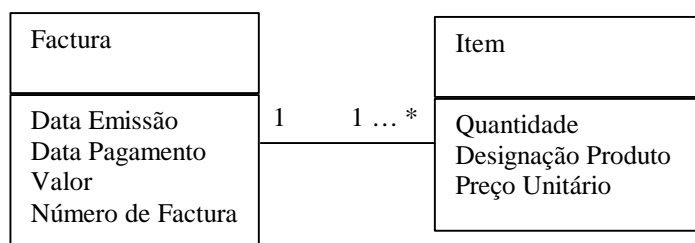
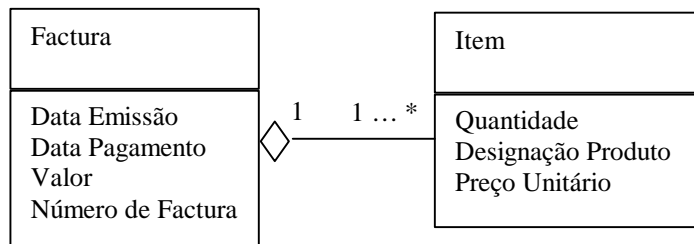
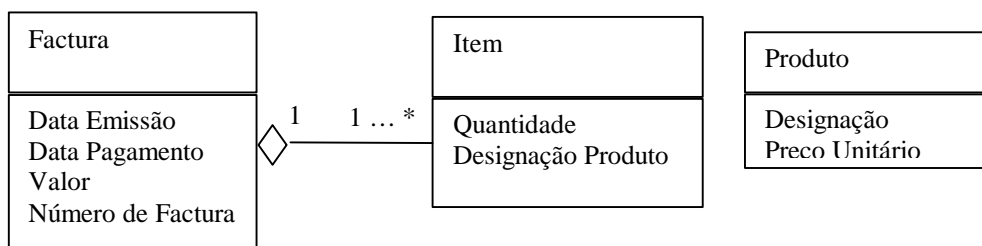
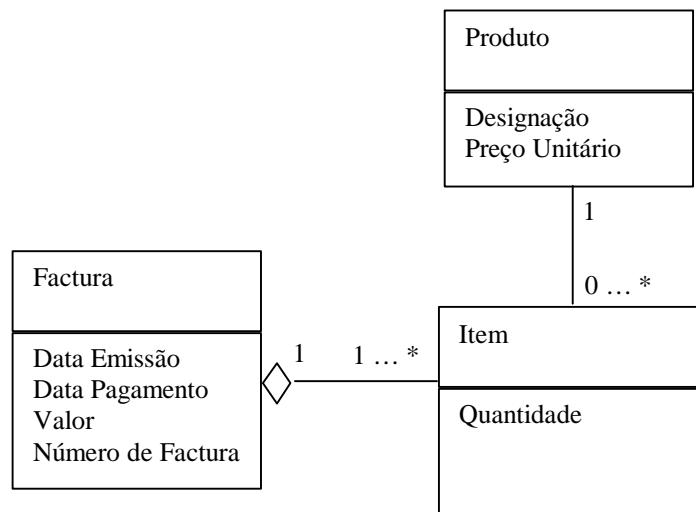


Figura 4 Representação de uma factura sem agregação

A representação da Figura 4 *esconde* uma realidade que pode ser considerada relevante: os itens apenas existem enquanto existir a factura da qual fazem parte. Dito de outra forma, uma factura é uma agregação de itens. Caso opte pela representação da Figura 5 (onde se introduz o símbolo gráfico que representa a agregação), fica explicitamente descrito que, por exemplo, se remover-mos uma factura, os itens dessa factura também serão removidos.

**Figura 5** Representação de uma factura com agregação

Com regra geral, um atributo nunca deve estar repetido num diagrama. Quando se refere atributo, não nos referimos a uma designação (em ambas as entidades Cliente e Fornecedor existirá o atributo Nome) mas a um domínio. Por exemplo, o desenho da Figura 6 é incorrecto porque o atributo Designação Produto corresponde (em termos semânticos) ao atributo Designação da Classe Produto. A relação entre o item e o produto deverá ser feita através de uma associação, tal como é apresentado na Figura 7.

**Figura 6** Um exemplo de dois atributos com o mesmo domínio**Figura 7** Correção ao diagrama da Figura 6

Na Figura 7 exemplificou-se a situação mais habitual nas agregações, uma associação do tipo “um para muitos” (um objecto é composto por vários objectos, e um objecto apenas faz parte de um objecto), no entanto as agregações podem existir em associações do tipo “muitos para muitos”. Na Figura 8 exemplifica-se essa situação: uma escola é composta por alunos, sendo que cada escola distingue os seus alunos de uma forma independente das outras. Isto é, um mesmo aluno (aqui entendido como uma pessoa com existência física), caso esteja inscrito em duas escolas, é identificado nas escolas de uma forma independente, por exemplo, pode corresponder ao aluno número 2345 da uma escola e ao aluno 456 da outra escola. Caso a primeira escola seja removida deste sistema, o objecto que representa o aluno 2345 é removido, mantendo-se o objecto que representa o aluno 456 da segunda escola.

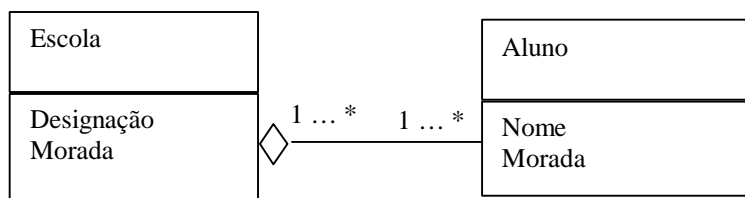


Figura 8 Exemplo de uma agregação “muitos” para “muitos”

Na Figura 9 apresenta-se um diagrama alternativo ao da Figura 8. Esse diagrama, apesar de aparentemente mais complexo, é provavelmente de leitura mais clara. A classe Aluno engloba os alunos entendidos como pessoas com existência física. A classe Aluno Inscrito engloba os alunos entendidos como as inscrições dos alunos numa escola. Note-se que dado que um aluno têm que estar obrigatoriamente associado a um Aluno Inscrito, à semelhança do que é descrito na Figura 5, caso todas as escolas onde um aluno esteja inscrito forem removidas, o aluno também será removido (ele só existe se existir uma inscrição a ele associado).

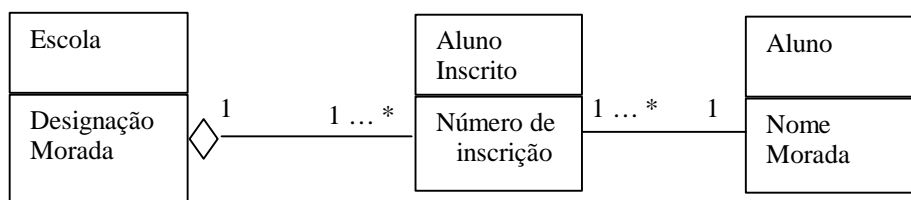


Figura 9 Alternativa ao diagrama da Figura 8

II.2 Generalização

A generalização é uma relação que permite representar a noção de pertença ou especificidade. Uma classe representa um conjunto de objectos que partilham, entre outras coisas, um conjunto comum de atributos. Mas isso não significa necessariamente que partilhem todos os seus atributos. Alguns objectos podem possuir um conjunto de atributos que outros não possuam. Por exemplo, todos os funcionários de uma empresa partilham os atributos nome, morada, número de contribuinte e salário base. No entanto, uma classe especial de funcionários – os vendedores – possuem atributos específicos, por exemplo, zona geográfica a que estão afectos e taxa de comissão. Os vendedores pertencem à classe dos funcionários, mas eles por si formam uma outra classe, a dos vendedores. Graficamente a generalização representa-se tal como consta na Figura 10. O termo herança é habitualmente associada às generalizações. Efectivamente, o vendedor *herda* todos os atributos (e operações) de funcionário.

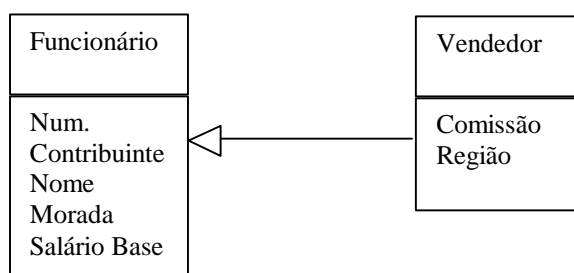


Figura 10 Representação gráfica de uma Generalização

No diagrama da Figura 10 não se encontra representada a multiplicidade da associação. Uma generalização é sempre uma associação do tipo “um para um”, um vendedor apenas pode corresponder a um funcionário, e vive versa. Um funcionário não tem necessariamente que ser vendedor, mas um vendedor é sempre um funcionário. Por outro lado, os funcionários não têm que ser todos vendedores, podem existir outros subconjuntos de funcionários que não estão explícitos no modelo. Na Figura 11 apresenta-se um outro exemplo mais completo. Os produtos não têm que ser todos matérias primas ou produtos acabados, podem existir produtos intermédios (produtos em via de fabrico). Um produto pode também ser simultaneamente produto final e matéria

prima para outros produtos finais. Estas diferentes possibilidades (a intersecção entre as classes ser ou não nula e a união das classes ser igual à *supra* classe) não é representada graficamente no modelo. Por fim, um produto ser ou não tóxico é independente (transversal) do facto de ser produto final ou matéria prima.

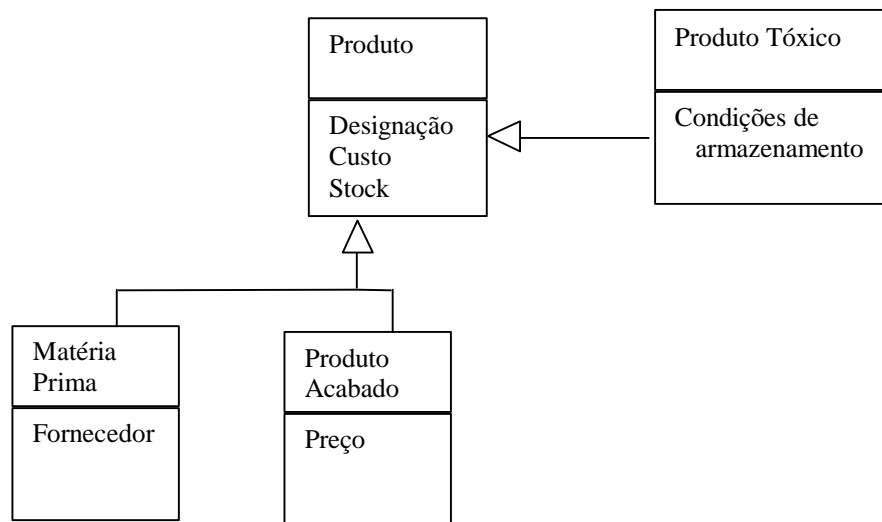


Figura 11 Generalização múltipla

Um modelo deve ser considerado correcto ou incorrecto em função da sua sintaxe (isto é, se obedece ou não aos requisitos sintácticos da linguagem) e em função da sua adequação ao domínio que se pretende representar. Os modelos podem ser utilizados meramente como instrumentos descritivos de um determinado domínio, sem que isso implique necessariamente que venham a ser implementados em sistemas informáticos. Quando um modelo pretende ser descritivo não é *correcto* esconder informação considerada relevante. Por exemplo, no esquema da Figura 11 poder-se-ia ter optado por não criar as classes Matéria Prima e Produto Acabado (os atributos Preço e Fornecedor pertenceriam à classe Produto). Essa situação pode ser considerada incorrecta porque se *esconde* a especificidade dos atributos Fornecedor e Preço. Decidir aquilo que é relevante ou, pelo contrário, apenas um mero detalhe, é muitas vezes uma questão de bom senso.

III Exemplos de Modelos

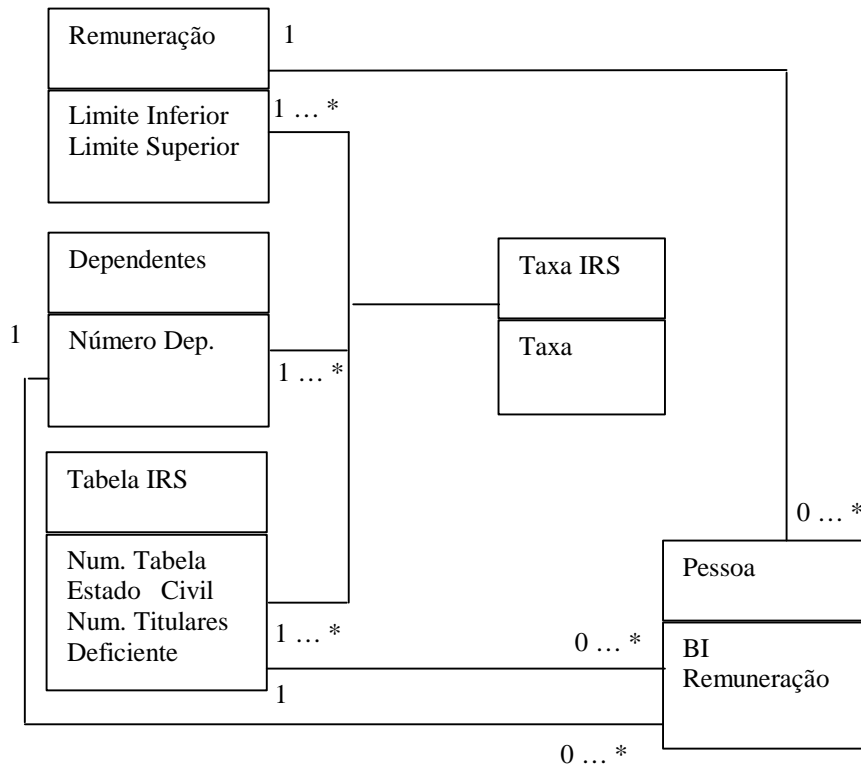
III.1 Tabelas de IRS

Considere a seguinte tabela de taxas de IRS. Trata-se de uma tabela para pessoas não casadas, e as taxas (%) são determinadas em função do rendimento da pessoas (remuneração mensal) e do número de dependentes a cargo da pessoa. O ministério das finanças numera as tabelas através de números inteiros (1, neste caso). Existem outras tabelas, todas com a mesma estrutura só que para situações diferentes; são elas, tabela 2 - casado, único titular; tabela 3 - casado, dois titulares; tabela 4 - não casado, deficiente; tabela 5 - casado, único titular, deficiente; tabela 6 - casado, dois titulares, deficiente.

Tabela 1 - Não casado

Remuneração mensal (escudos)	Número de dependentes				
	0	1	2	3	4 ou mais
até 47 500	0%	0%	0%	0%	0%
até 48 000	4%	2%	0%	0%	0%
até 52 000	7%	4%	2.5%	1%	0%
...					
superior a 800 500	32%	32%	31%	30%	27%

Pretende-se um modelo que, para além de permitir armazenar correctamente as tabelas de IRS, contemple uma classe *Pessoa*, e a informação necessária para, dada uma determinada pessoa (bi), poder obter-se automaticamente o montante de desconto para IRS dessa pessoa

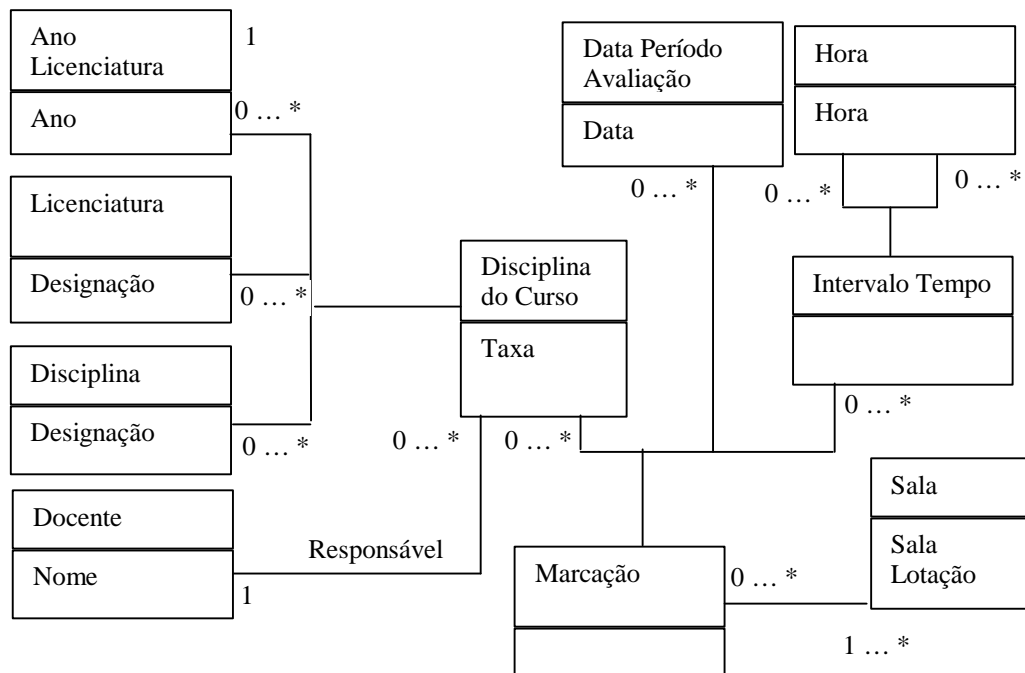
Resolução

III.2 Exames

Considere-se a seguinte folha de requisição de salas de exame. A parte superior é preenchida pelo docente que pretende reservar a sala, sendo a parte inferior preenchida manualmente pelo Concelho Directivo. Pretende-se que o C. Directivo utilize um sistema informático que preencha automaticamente a parte inferior da folha.

Instituto Superior de Ciências do Trabalho e da Empresa Marcação de Salas para Avaliações	
Disciplina: _____ Licenciatura: _____ Ano: _____ Data: _____ Hora Inicio: _____ Hora Fim: _____ Número de alunos previstos: _____ Número de docentes disponíveis: _____ Docente Responsável: _____	
..... (a preencher pelo conselho directivo)	
Sala Atribuida _____ _____ _____	lotação _____ _____ _____

Note-se que um sistema que implemente a tarefa de marcação de salas deverá validar a existência da disciplina no ano da licenciatura, bem como a identificação do docente como responsável. O sistema face ao número de alunos e ao número de docentes disponíveis para acompanhar a avaliação, deverá sugerir as melhores salas para avaliação (em função também da lotação de cada sala). É necessário ter em conta as salas disponíveis no momento da avaliação.

Resolução

III.3 Multibanco

Considere-se o seguinte extracto de multibanco gerado automaticamente por uma máquina ATM (relativo a uma operação de levantamento de dinheiro):

** Multibanco **

Número Caixa: 035/0159/1

Número Transacção: 0352

Cartão: 59035088870812

Data: 21/05/94

Conta: 00015905472620

Hora: 18:35

Operação: Depósito

Número Movimento Cartão:22

Montante: 15000\$00

Saldo Após Operação:140000\$00

Multibanco num minuto você e o seu banco

No seu interesse faculte-nos o Nr Fiscal

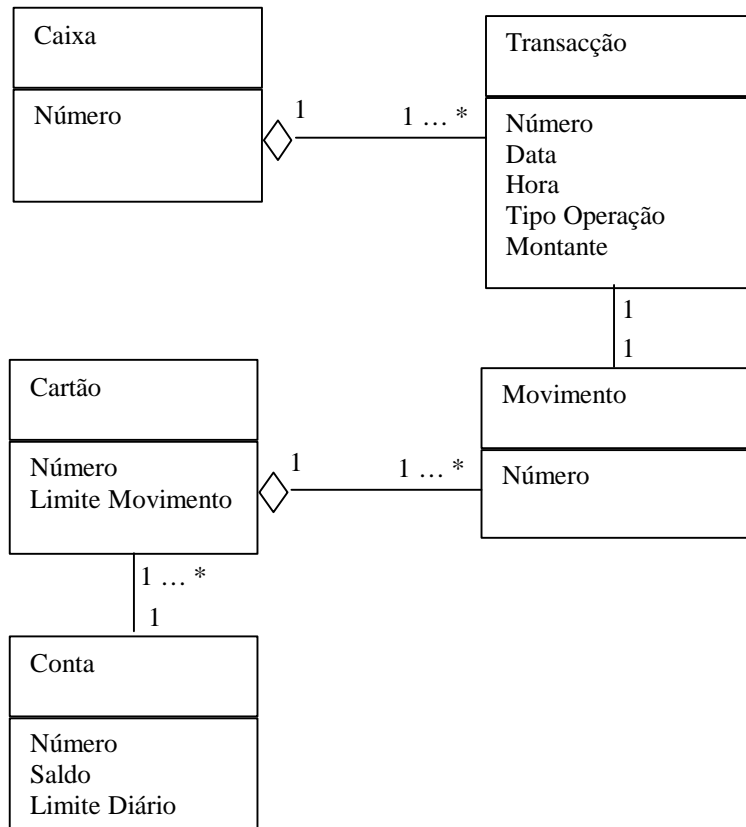
** Obrigado **

O número de caixa corresponde ao número da caixa onde foi efectuada a operação e o número de transacção respeita ao número de transacções efectuadas naquela caixa. O cartão e a conta dizem respeito ao cliente. Considere que estas máquinas estão afectas a apenas um banco, ou seja, só aceita cartões emitidos pelo banco proprietário da caixa. Considere ainda que associado a cada cartão (número), para cada operação de levantamento, existe um limite máximo por operação e um limite diário máximo.

O valor do Saldo após Operação corresponde ao saldo existente na conta menos (ou mais no caso de uma operação de depósito) o valor levantado.

Só poderá ser efectuado um levantamento desde que não ultrapasse os limites antes indicados e no caso de o Saldo Após Operação não ficar negativo.

Pretende-se um modelo que armazene a informação de modo a que um sistema proceda à geração automática dos extractos de depósito e levantamento.

Resolução

IV Transposição: Modelo de Classes / Modelo Relacional

As regras de transposição que de seguida se enumeram visam obter um modelo relacional completo e coerente, isto é, um modelo que, sem incoerências e redundâncias, permite representar toda a informação do modelo de classes. A aplicação das regras por vezes geram um modelo pouco eficiente que posteriormente terá que ser otimizado.

IV.1 Regras

Tabelas

Todas as classes e as associações “muitos para muitos” dão origem a tabelas, e nada mais dá origem a tabelas.

Atributos não chave

Todos os atributos de uma classe são atributos da tabela que implementa a classe, inclusive os atributos das classes associativas. Os atributos das associações que não deram origem a tabelas são: atributos das tabelas que implementam as classes do lado “muitos”, no caso das associações “um para muitos”; atributos das tabelas que implementam uma das classes do lado “um”, no caso das associações “um para um”.

Chave Primária

Todas as tabelas possuem uma chave primária. Para cada tabela é necessário encontrar o conjunto mínimo de atributos que permite identificar univocamente um registo. No caso das tabelas que não resultam de classes associativas, de generalizações ou das sub classes das agregações, a chave tem que corresponder a um conjunto de atributos que constam na classe – caso não exista esse conjunto de atributos, é necessário definir um novo atributo (ou atributos). Casos especiais:

Classes Associativas – a chave da tabela corresponde ao conjunto de atributos chave das tabelas que implementam as classes argumento da classe associativa;

Agregações (“um para muitos”) – a chave da tabelas correspondentes às sub classes correspondem à chave da tabela que implementa a supra classe à qual é necessário

acrescentar um atributo (*chave parcial*) da sub classe (caso não exista esse atributo, terá que ser criado);

Agregações (“muitos para muitos”) – este tipo de agregação, tal como se encontra ilustrado na Figura 9 pode ser decomposto numa Agregações - “um para muitos”, aplicando-se a regra anterior;

Generalizações – duas situações distintas podem ocorrer numa generalização: (i) as classes *filhas* são autónomas, isto é, existem independentemente da classe *pai* e, (ii) as classes *filhas* apenas existem enquanto existirem a classe *pai*.

No primeiro cenário (i), a chave das tabelas que implementam as classes *filhas* é obtida através dos atributos da própria tabela. Terá que ser criado um atributo chave para a tabela que implementa a classe *pai*, sendo que essa tabela deverá ter uma propriedade discriminante (um atributo) que indique a qual das *filhas* o registo diz respeito. Todos os atributos chave das tabelas *filhas* terão que constar na tabela *pai* (não como atributos chave).

No segundo cenário (ii), a chave da tabelas que implementam a classe *pai* é obtida através dos atributos da própria tabela. As tabelas correspondentes às classes *filhas* terão a mesma chave da tabela *pai*.

Chaves Estrangeiras

Para além das chaves estrangeiras referidas anteriormente (Classes Associativas, Agregações e Generalizações), no caso das associações “um para muitos” e “um para um” considera-se as seguintes regras:

- (i) associações “um para muitos” – a tabela correspondente à classe do lado “muitos” tem como chave estrangeira a chave da tabela correspondente à classe do lado “um”;
- (ii) associações “um para um” – uma das tabelas tem como chave estrangeira a chave da outra tabela.

IV.2 Exemplos

1) Modelo da Figura 7

Produto (Designação, Preço Unitário)

Factura (Número de Factura, data Emissão, data pagamento, Valor)

Item (Número de Factura, Número de Item, Quantidade, *Designação Produto*)

2) Modelo da Figura 11

Assume-se que a empresa implementou uma codificação autónoma para os produtos acabados e para as matérias primas.

Produto (ID Produto, Designação, Custo, Stock, *Código MP*, *Código PA*, Tipo Produto)

Matéria Prima (Código MP, Fornecedor)

Produto Acabado (Código PA, Preço)

Produto Tóxico (ID Produto, Condições de Armazenamento)

3) Exercício IRS

Remuneração(ID Intervalo, Limite Inferior, Limite Superior)

Dependentes (Número Dependentes)

Tabela (Número Tabela, Estado Civil, Número Titulares, Deficiente)

Taxa IRS (ID Intervalo , Número Dependentes, Número Tabela, Taxa)

Pessoa (BI, Remuneração, *ID Intervalo*, *Número Tabela*, *Número Dependentes*)

4) Exercício Exames

Ano Licenciatura (Ano)

Licenciatura (Designação)

Disciplina (Designação)

Docente (ID Docente, Nome)

Data Período Avaliação (Data)

Hora (Hora)

Intervalo Tempo (Hora Inicio, Hora Fim)



Sala (Sala, Lotação)

Disciplina do Curso (Ano, Designação Lic, Designação Disc, *ID Docente*, Taxa)

Marcação (Data, Hora Inicio, Hora Fim, Ano, Designação Lic, Designação Disc)

Marcação-Sala (Data, Hora Inicio, Hora Fim, Ano, Designação Lic, Designação Disc, Sala)

5) Exercício Multibanco

Caixa(Número)

Transacção(Número Caixa, Número Transacção, data, Hora, Tipo Operação, Montante)

Cartão(Número, Limite Movimento, *Número Conta*)

Movimento(Número Cartão, Número Movimento, *Número Caixa*, *Número Transacção*)

Conta(Número, Saldo, Limite Diário)