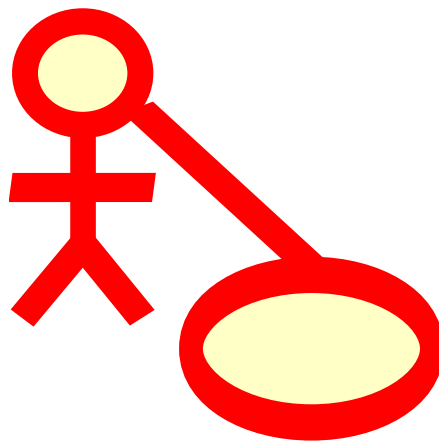


# Manual do Umbrello UML Modeller



## Manual do Umbrello UML Modeller

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>7</b>
<b>2</b>	<b>Bases de UML</b>	<b>8</b>
2.1	Acerca do UML	8
2.2	Elementos de UML	9
2.2.1	Diagrama de Casos de Utilização	9
2.2.1.1	Caso de Utilização	9
2.2.1.2	Actor	10
2.2.1.3	Descrição de Caso de Utilização	10
2.2.2	Diagrama de Classes	10
2.2.2.1	Classe	11
2.2.2.1.1	Atributos	11
2.2.2.1.2	Operações	12
2.2.2.1.3	Modelos	12
2.2.2.2	Associações da Classe	12
2.2.2.2.1	Generalização	12
2.2.2.2.2	Associações	12
2.2.2.2.3	Agregação	13
2.2.2.2.4	Composição	13
2.2.2.3	Outros Itens do Diagrama de Classes	13
2.2.2.3.1	Interfaces	13
2.2.2.3.2	Tipos de dados	14
2.2.2.3.3	Enumerados	14
2.2.2.3.4	Pacotes	14
2.2.3	Diagramas de Sequência	14
2.2.4	Diagramas de Colaboração	15
2.2.5	Diagramas de Estados	15
2.2.5.1	Estado	16
2.2.6	Diagrama de Actividades	17
2.2.6.1	Actividade	17
2.2.7	Elementos Auxiliares	17
2.2.8	Diagramas de Componentes	18

2.2.9	Diagramas de Entrada em Produção . . . . .	18
2.2.10	Diagramas de Entidade-Associação . . . . .	18
2.2.10.1	Entidade . . . . .	19
2.2.10.1.1	Atributos da Entidade . . . . .	19
2.2.10.1.2	Restrições . . . . .	19
2.2.11	Conceitos Estendidos dos Diagramas de Entidade-Associação (ER) . . . . .	20
2.2.11.1	Especialização . . . . .	20
2.2.11.1.1	Especialização Disjunta . . . . .	20
2.2.11.1.2	Especialização Sobreposta . . . . .	21
2.2.11.1.3	Categoria . . . . .	21
<b>3</b>	<b>Trabalhar com o Umbrello UML Modeller</b>	<b>23</b>
3.1	Interface do Utilizador . . . . .	23
3.1.1	Vista em Árvore . . . . .	24
3.1.2	Janela de Histórico de Comandos e Documentação . . . . .	24
3.1.3	Área de Trabalho . . . . .	24
3.2	Criar, Carregar e Gravar Modelos . . . . .	25
3.2.1	Novo Modelo . . . . .	25
3.2.2	Gravar Modelo . . . . .	25
3.2.3	Carregar Modelo . . . . .	25
3.3	Editar Modelos . . . . .	25
3.4	Adicionar e Remover Diagramas . . . . .	26
3.4.1	Criar Diagramas . . . . .	26
3.4.2	Remover Diagramas . . . . .	26
3.4.3	Mudar o Nome a Diagramas . . . . .	26
3.5	Editar Diagramas . . . . .	26
3.5.1	Inserir Elementos . . . . .	27
3.5.2	Apagar Elementos . . . . .	27
3.5.3	Editar Elementos . . . . .	28
3.5.4	Editar Classes . . . . .	28
3.5.4.1	Configuração Geral das Classes . . . . .	28
3.5.4.2	Configuração dos Atributos das Classes . . . . .	28
3.5.4.3	Configuração das Operações das Classes . . . . .	28
3.5.4.4	Configuração do Modelo de Classe . . . . .	28
3.5.4.5	Página de Associações da Classe . . . . .	29
3.5.4.6	Página de Visualização de Classes . . . . .	29
3.5.4.7	Página de Estilo da Classe . . . . .	29
3.5.5	Associações . . . . .	29
3.5.5.1	Pontos-Âncora . . . . .	30
3.5.6	Notas, Texto e Caixas . . . . .	30
3.5.6.1	Âncoras . . . . .	30

<b>4</b>	<b>Importação e Geração de Código</b>	<b>31</b>
4.1	Geração de Código . . . . .	31
4.1.1	Gerar Código . . . . .	31
4.1.1.1	Opções de Geração . . . . .	32
4.1.1.1.1	Detalhe dos Comentários . . . . .	32
4.1.1.1.2	Pastas . . . . .	32
4.1.1.1.3	Política de Sobreposição . . . . .	33
4.1.1.1.4	Linguagem . . . . .	33
4.1.1.2	Assistente de Geração - Geração . . . . .	33
4.2	Importação de Código . . . . .	33
<b>5</b>	<b>Outras Características</b>	<b>35</b>
5.1	Outras Características do Umbrello UML Modeller . . . . .	35
5.1.1	Copiar objectos como imagens PNG . . . . .	35
5.1.2	Exportar para uma Imagem . . . . .	35
5.1.3	Impressão . . . . .	35
5.1.4	Pastas Lógicas . . . . .	35
<b>6</b>	<b>Autores e História</b>	<b>37</b>
<b>7</b>	<b>Copyright</b>	<b>38</b>

## **Resumo**

O Umbrello UML Modeller ajuda o processo de desenvolvimento de 'software' usando a norma da indústria que é a Unified Modelling Language (UML) para lhe permitir criar diagramas para desenhar e documentar os seus sistemas.

# Capítulo 1

## Introdução

O Umbrello UML Modeller é uma ferramenta de diagramas de UML que o pode suportar no processo de desenvolvimento de 'software'. Especialmente durante o processo de análise e de desenho deste processo, o Umbrello UML Modeller ajudá-lo-á a obter um produto de alta qualidade. O UML pode também ser usado para documentar os seus desenhos de 'software' e a ajudá-lo a si e aos seus colegas programadores.

Se tiver um bom modelo do 'software', será a melhor forma de comunicar com outros programadores a trabalhar no projecto e com os seus clientes. Um bom modelo é extremamente importante para os projectos de dimensão média e grande, mas também é bastante útil para os projectos pequenos. Mesmo que você esteja a trabalhar num projecto pequeno de uma pessoa só, irá beneficiar de um bom modelo porque terá uma boa ideia geral que o ajudará a codificar bem as coisas logo à primeira vez.

O UML é a linguagem de diagramas usada para descrever esses modelos. Você poderá representar as suas ideias em UML, usando vários tipos de diagramas. O Umbrello UML Modeller 2.11 suporta os seguintes tipos:

- Diagrama de Classes
- Diagrama de Sequência
- Diagrama de Colaboração
- Diagrama de Casos de Utilização
- Diagrama de Estados
- Diagrama de Actividades
- Diagrama de Componentes
- Diagrama de Entrada em Produção
- Diagrama de Entidade-Relação

Para mais informações sobre o UML poderá consultar a página Web da [OMG](http://www.omg.org), <http://www.omg.org>, que foi quem criou a norma UML.

Esperemos que você goste do Umbrello UML Modeller e que o ajude a criar 'software' de alta qualidade. O Umbrello UML Modeller é 'Software' Livre e está disponível sem custos alguns, a única coisa que lhe pedimos é para comunicar todos os erros, problemas ou sugestões para os programadores do Umbrello UML Modeller em [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org) ou <http://bugs.kde.org>.

## Capítulo 2

# Bases de UML

### 2.1 Acerca do UML

Este capítulo dar-lhe-á uma ideia geral sobre as bases do UML. Tenha em mente que isto não é um tutorial compreensivo sobre o UML mas sim uma breve introdução ao mesmo, como tal poderá ser lido como um tutorial de UML. Se você quiser aprender mais sobre a Unified Modelling Language ou, na generalidade, sobre a análise e desenho de 'software', baseie-se num dos vários livros disponíveis sobre o tópico. Existem também vários tutoriais na Internet que você poderá usar como ponto de partida.

A Unified Modelling Language (UML) é uma linguagem ou notação de diagramas para especificar, visualizar e documentar modelos de 'software' orientados por objectos. O UML não é um método de desenvolvimento, o que significa que não lhe diz o que fazer primeiro ou o que fazer a seguir ou como desenhar o seu sistema, mas ajuda-o a visualizar o seu desenho e a comunicar com os outros. O UML é controlado pelo Object Management Group (OMG) e é a norma da indústria para descrever graficamente o 'software'.

O UML está desenhado para o desenho de 'software' orientado por objectos e tem uma utilização limitada para outros paradigmas de programação.

O UML é composto por vários elementos do modelo que representam as diferentes partes de um sistema de 'software'. Os elementos de UML são usados para criar diagramas que representam uma dada parte ou um ponto de vista do sistema. São suportados os seguintes tipos de diagramas pelo Umbrello UML Modeller:

- Os *Diagramas de Casos de Uso* mostram os actores (as pessoas ou os outros utilizadores do sistema), os casos de utilização (os cenários em que eles usam o sistema) e as suas relações
- Os *Diagramas de Classes* mostram as classes e as relações entre elas
- Os *Diagramas de Sequência* mostram os objectos e uma sequência de chamadas de métodos que eles fazem a outros objectos.
- Os *Diagramas de Colaboração* mostram os objectos e as suas relações, colocando alguma ênfase nos objectos que participam na troca de mensagens
- Os *Diagramas de Estados* mostram os estados, as mudanças de estado e os eventos num objecto ou numa parte do sistema
- Os *Diagramas de Actividades* mostram as actividades e as mudanças de uma actividade para outra com os eventos a ocorrerem numa parte do sistema
- Os *Diagramas de Componentes* mostram os componentes de alto-nível de programação (como as KParts ou os Java Beans).



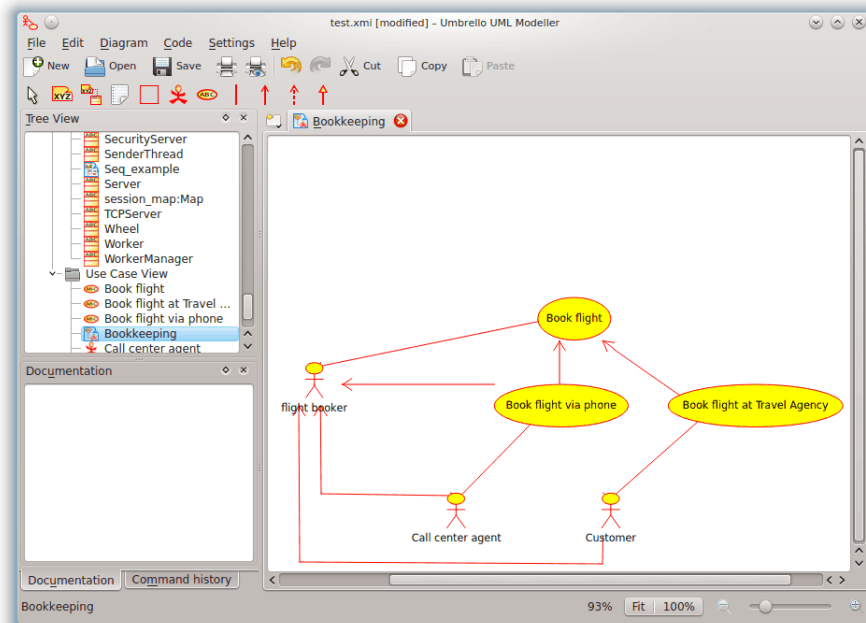
- Os *Diagramas de Entradas em Produção* mostram as instâncias dos componentes e as suas relações.
- Os *Diagramas de Entidade-Associação* mostram os dados e as relações e as restrições entre os dados.

## 2.2 Elementos de UML

### 2.2.1 Diagrama de Casos de Utilização

Os Diagramas de Casos de Uso descrevem as relações e as dependências entre um grupo de *Casos de Uso* e os Actores que participam no processo.

É importante reparar que os Diagramas de Casos de Uso não são adequados para representar o desenho e também não podem descrever os detalhes internos de um sistema. Os Diagramas de Casos de Uso pretendem facilitar a comunicação com os utilizadores futuros do sistema e com o cliente, e são especialmente úteis para determinar as funcionalidades necessárias que o sistema deverá ter. Os Diagramas de Casos de Uso indicam *o que* o sistema deverá fazer mas não devem — e não podem — especificar *como* isto deverá ser feito.



*Umbrello UML Modeller a mostrar um diagrama de Casos de Utilização*

#### 2.2.1.1 Caso de Utilização

Um *Caso de Utilização* descreve — do ponto de vista dos actores — um grupo de actividades de um sistema que produzem um resultado concreto e tangível.

Os Casos de Uso são descrições das interacções típicas entre os utilizadores de um sistema e o sistema propriamente dito. Eles representam a interface externa do sistema e especificam um dado tipo de requisitos sobre o que o sistema tem de fazer (lembre-se, só 'o quê', não 'como').

Ao lidar com os Casos de Uso, é importante recordar algumas regras simples:

- Cada Caso de Utilização está relacionado com pelo menos um actor
- Cada Caso de Utilização tem um iniciador (ou seja, um actor)
- Cada Caso de Uso conduz a um resultado relevante (um resultados com 'valor de negócio')

Os Casos de Uso também poderão ter relações com outros Casos de Uso. Os três tipos de relações mais típicos entre Casos de Uso são:

- «*include*» (*incluir*) que indica que um Caso de Uso toma lugar *dentro* de outro Caso de Uso
- «*extends*» (*extende*) que indica que, em certas situações ou numa dada altura (chamada de ponto de extensão), um Caso de Uso será estendido por outro.
- *Generalization* (generalização) indica que um Caso de Uso herda as características do 'Super'-Caso de Uso e poderá implementar novamente algumas delas ou adicionar novas de uma forma semelhante à da herança de classes.

#### 2.2.1.2 Actor

Um actor é uma entidade externa (fora do sistema) que interage com o sistema ao participar (ou iniciar normalmente) um Caso de Uso. Os actores poderão ser pessoas reais (como, por exemplo, utilizadores do sistema), outros sistemas informáticos ou eventos exteriores.

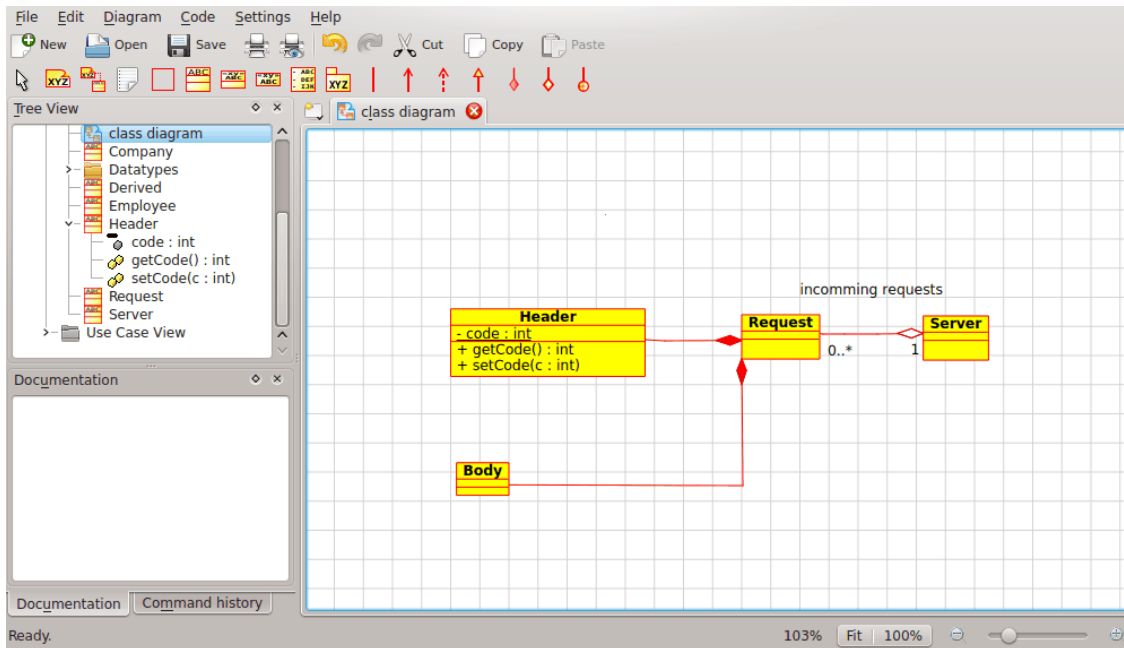
Os actores não representam as pessoas *físicas* ou os sistemas, mas sim o seu *papel*. Isto significa que, quando uma pessoa interage com o sistema de formas diferentes (assumindo papéis diferentes), ele será representado por vários actores. Por exemplo, uma pessoa que dá suporte ao cliente por telefone e que recebe encomendas do cliente para colocar no sistema seria representado por um actor 'Técnico de Suporte' e por um actor 'Representante de Vendas'

#### 2.2.1.3 Descrição de Caso de Utilização

As descrições dos Casos de Uso são narrativas textuais do Caso de Uso. Elas normalmente tomam a forma de uma nota ou de um documento que esteja associado de alguma forma ao Caso de Uso e explicam os processos ou actividades que tomam lugar no Caso de Uso.

### 2.2.2 Diagrama de Classes

Os Diagramas de Classes mostram as diferentes classes que compõem um sistema e como elas se relacionam umas com as outras. Os Diagramas de Classes são apontados normalmente como 'estáticos' porque mostram as classes, em conjunto com os seus métodos e atributos, assim como as relações estáticas entre elas, quais as classes que 'conhecem' outras classes ou que 'fazem parte' de outra classe, mas não mostram as chamadas de métodos entre elas.

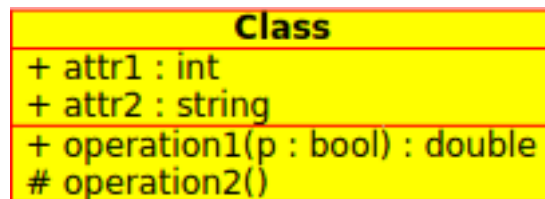


O Umbrello UML Modeller a mostrar um diagrama de Classes

### 2.2.2.1 Classe

Uma classe define os atributos e os métodos para um conjunto de objectos. Todos os objectos desta classe (as instâncias da mesma) partilham o mesmo comportamento e têm o mesmo conjunto de atributos (cada objecto tem o seu próprio conjunto). O termo 'Tipo' é usado em algumas ocasiões em vez de Classe, mas é importante mencionar que os dois conceitos não são iguais e que o Tipo é um termo mais genérico.

No UML, as Classes são representadas por rectângulos com o nome da classe, e poderão também mostrar os atributos e as operações da classe em outros dois 'compartimentos' dentro do rectângulo.



Representação visual de uma Classe em UML

#### 2.2.2.1.1 Atributos

No UML, os Atributos são mostrados como tendo o seu nome, pelo menos, e poderão mostrar também o seu tipo, o seu valor inicial e outras propriedades. Os atributos também poderão ser mostrados com a sua visibilidade:

- + Representa atributos *públicos*
- # Representa atributos *protegidos*
- - Representa atributos *privados*

### 2.2.2.1.2 Operações

As operações (métodos) são também mostradas contendo pelo menos o seu nome e poderão também mostrar os seus parâmetros e tipos de valor devolvidos. As operações podem, assim como os Atributos, mostrar a sua visibilidade:

- + Corresponde a operações *públicas*
- # Corresponde a operações *protegidas*
- - Corresponde a operações *privadas*

### 2.2.2.1.3 Modelos

As Classes poderão ter modelos, um valor que é usado para uma classe ou tipo não especificado. O tipo de modelo é definido quando uma classe é iniciada (isto é um objecto é criado). Os modelos existem como 'template' no C++ actual e serão introduzidos no Java 1.5 onde serão chamados de 'Generics'.

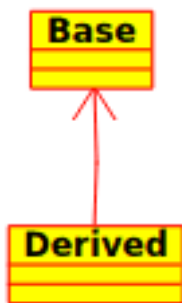
## 2.2.2 Associação da Classe

Uma classe pode-se relacionar (ser associada) com outra qualquer de diferentes maneiras:

### 2.2.2.2.1 Generalização

A herança é um dos conceitos fundamentais da programação orientada por objectos, nos quais uma classe 'ganha' todos os atributos e operações da classe que herda, podendo sobrepor ou modificar algumas delas, assim como adicionar mais atributos ou operações próprias.

No UML, uma associação por *Generalização* entre duas classes representa o conceito de herança entre uma classe de base e uma classe derivada. No UML, as Generalizações são representadas com uma linha que liga as duas classes, em que existe uma seta do lado da classe de base.



Representação visual de uma generalização em UML

### 2.2.2.2.2 Associações

Uma associação representa uma relação entre classes e dá a semântica e a estrutura comum para vários tipos de 'ligações' entre os objectos.

As associações são o mecanismo que permite aos objectos comunicarem uns com os outros. Descreve a ligação entre as diferentes classes (a ligação entre os objectos em si é chamada de *ligação do objecto*).

As associações podem ter um papel que indica o objectivo da associação e pode ser unidireccionais ou bidireccionais (indica se os dois objectos que participam na relação poderão enviar mensagens um para o outro, ou se só um deles é que conhece o outro). Cada extremo da associação também tem um valor de multiplicidade, que define quantos objectos desse lado da associação poder-se-ão relacionar com um objecto do outro lado.

No UML, as associações são representadas como linhas que ligam as classes que participam na relação e poderá também mostrar o papel e a multiplicidade de cada um dos participantes. A multiplicidade é mostrada como um intervalo [mín..máx] de valores não-negativos ou com um asterisco (\*) no lado do máximo que representa o infinito.



Uma representação visual de uma associação em UML

### 2.2.2.2.3 Agregação

As agregações são um tipo especial de associações nas quais as duas classes participantes não têm um estado igual, mas têm uma relação 'do 'todo' para as partes'. Uma agregação diz como é que a classe que têm o papel do 'todo' é composta (ou tem) as outras classes, que têm o papel das partes. Para as agregações, a classe que actua como o 'todo' tem sempre uma multiplicidade de um.

No UML, as agregações são representados por uma associação com um losango do lado do 'todo'.

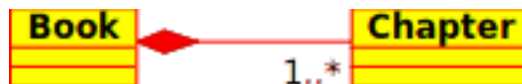


Uma representação visual de uma relação de agregação no UML

### 2.2.2.2.4 Composição

As composições são associações que representam agregações *muito fortes*. Isto significa que as composições formam também relações do 'todo' para as partes, mas a relação é tão forte que as partes não podem existir por si só. Elas só existem dentro do todo e se o todo for destruído, as partes desaparecem também.

No UML, as Composições são representadas por um losango a cheio do lado do 'todo'.



### 2.2.2.3 Outros Itens do Diagrama de Classes

Os diagramas de classe poderão conter outros itens para além das classes.

#### 2.2.2.3.1 Interfaces

As interfaces são classes abstractas, o que significa que as instâncias não podem ser criadas directamente a partir delas. Elas poderão conter operações, mas não podem conter atributos. As classes podem herdar das interfaces (através de uma relação de realização) e as instâncias poderão então ser compostas a partir desses diagramas.

### 2.2.2.3.2 Tipos de dados

Os tipos de dados são primitivas que vêm tipicamente incorporadas numa linguagem de programação. Os exemplos comuns incluem os inteiros e os booleanos. Eles não poderão ter relações com as classes, mas as classes poderão ter relações com eles.

### 2.2.2.3.3 Enumerados

Os enumerados são uma lista simples de valores. Um exemplo típico são os enumerados para os dias da semana. Como os tipos de dados, os enumerados não poderão ter relações com as classes, mas as classes poderão ter relações com eles.

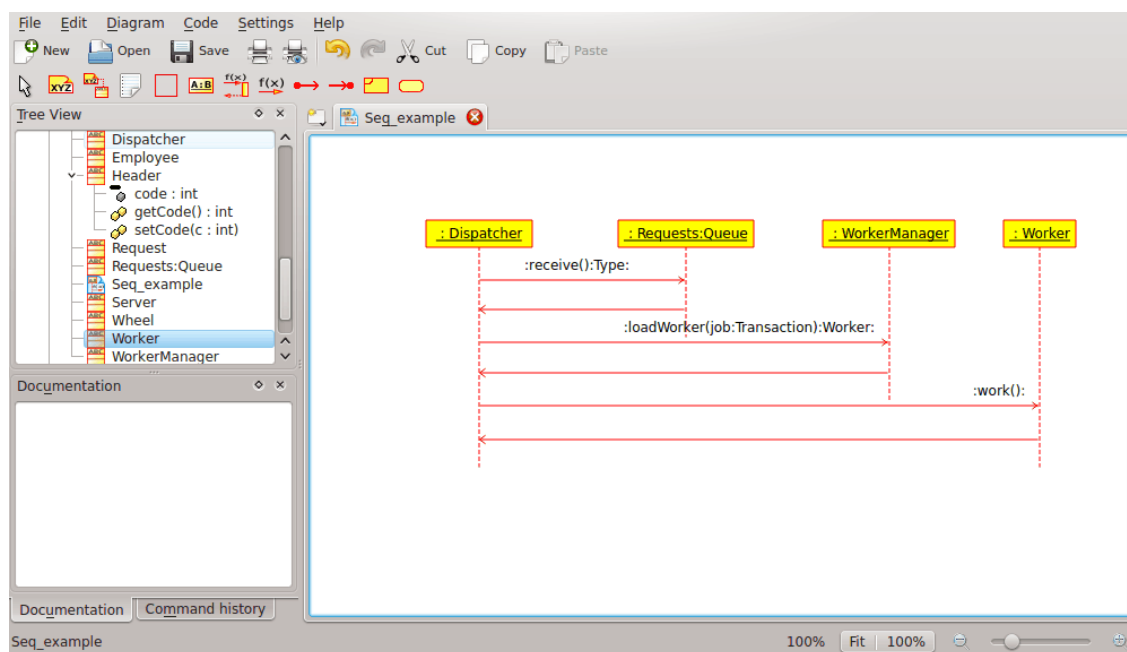
### 2.2.2.3.4 Pacotes

Os pacotes representam um espaço de nomes numa linguagem de programa. Num diagrama, eles são usados para representar partes de um sistema que contém mais do que uma classe, podendo ser mesmo centenas de classes.

## 2.2.3 Diagramas de Sequência

Os Diagramas de Sequência mostram a troca de mensagens (isto é as chamadas aos métodos) entre os vários objectos numa situação específica delimitada no tempo. Os objectos são instâncias das classes. Os Diagramas de Sequência colocam uma ênfase especial na ordem e nas alturas em que as mensagens são enviadas para os objectos.

Nos Diagramas de Sequência, os objectos são representados através de linhas tracejadas verticais, com o nome do objecto no topo. O eixo do tempo também é vertical, e vai aumentando de cima para baixo, de modo a que as mensagens sejam enviadas de um objecto para outro, sob o formato de setas com o nome da operação e dos parâmetros.



*O Umbrello UML Modeller a mostrar um Diagrama de Sequência*

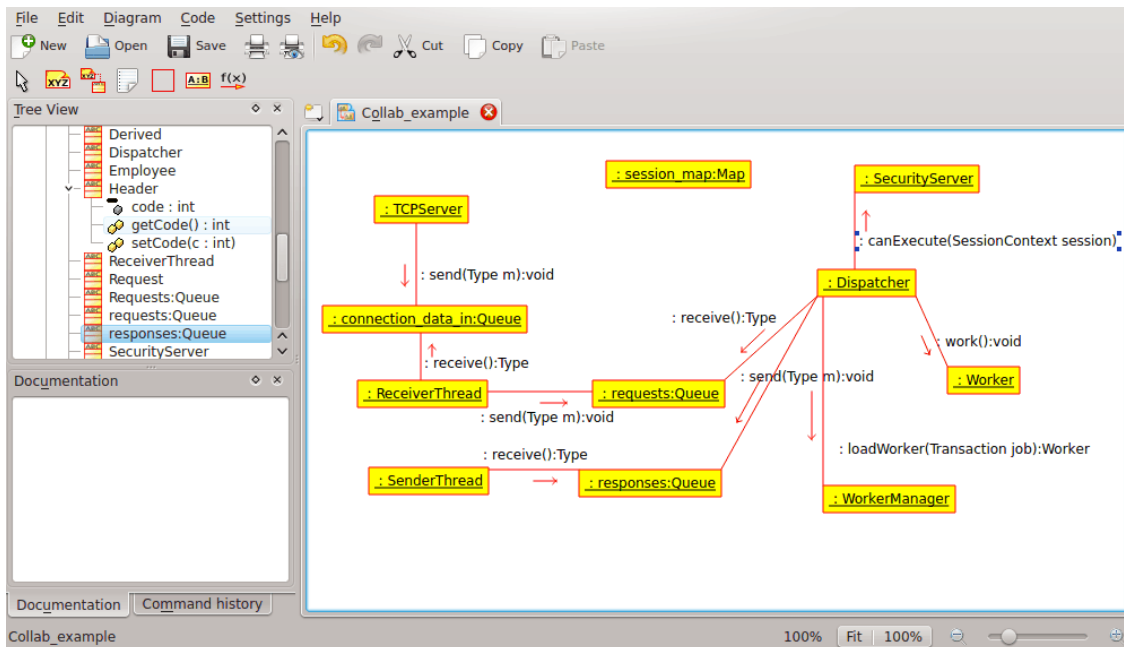
As mensagens tanto podem ser síncronas, o que acontece normalmente nas chamadas de mensagens quando o controlo é passado para o objecto que é invocado até que esse método termine

a sua execução, ou poderão também ser assíncronas, onde o controlo é passado de volta directamente para o objecto que invoca o método. As mensagens síncronas têm uma caixa vertical do lado do objecto que é chamado para mostrar o fluxo de controlo do programa.

## 2.2.4 Diagramas de Colaboração

Os Diagramas de Colaboração mostram as interações entre os objectos que participam numa dada situação. Esta é mais ou menos a mesma informação que é mostrada pelos Diagramas de Sequência, mas existe também uma ênfase posta sobre como as interações ocorrem no tempo, enquanto que os Diagramas de Colaboração colocam as relações entre os seus objectos e a sua topologia em primeiro plano.

Nos Diagramas de Colaboração, as mensagens enviadas de um objecto para o outro são representadas por setas que mostram o nome da mensagem, os parâmetros e a sequência da mensagem. Os Diagramas de Colaboração são particularmente adequados a mostrar um fluxo específico de um programa ou situação, e são uns dos melhores tipos de diagramas para demonstrar ou explicar rapidamente um processo na lógica do programa.



O Umbrello UML Modeller a mostrar um Diagrama de Colaboração

## 2.2.5 Diagramas de Estados

Os Diagramas de Estados mostram os diferentes estados de um objecto durante a sua vida, bem como os estímulos que fazem com que o objecto mude o seu estado.

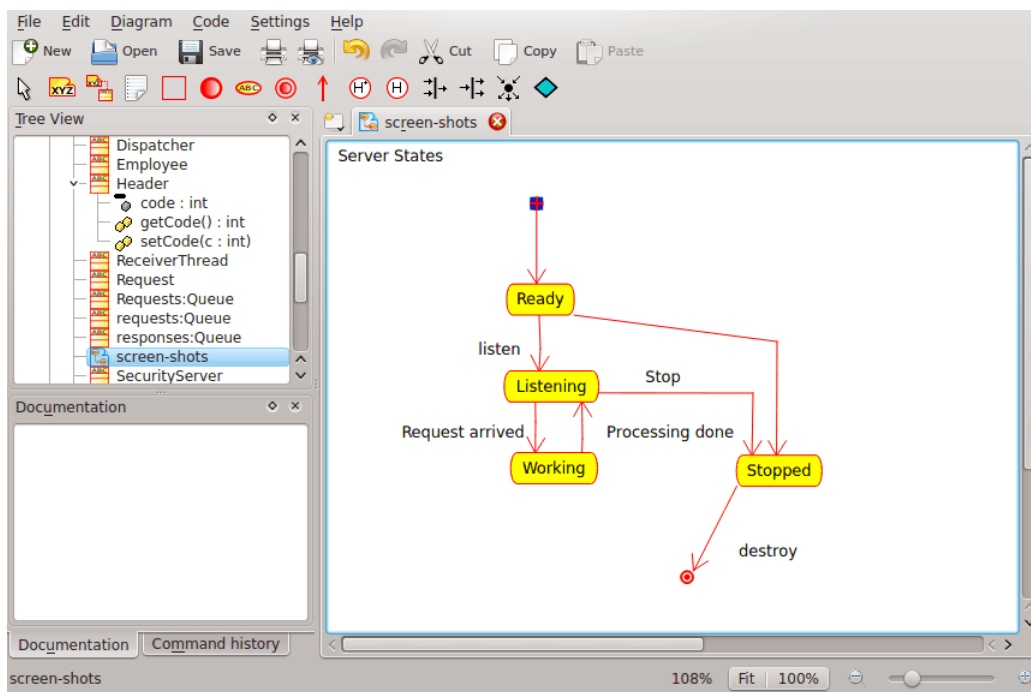
Os Diagramas de Estados vêem os objectos como *máquinas de estados* ou autónomos finitos que poderão estar num estado pertencente a uma lista de estados finitos e que poderão mudar o seu estado através de um estímulo pertencente a um conjunto finito de estímulos. Por exemplo, um objecto do tipo *ServidorRede* poderá estar num dos seguintes estados durante a sua vida:

- Pronto
- À espera
- A trabalhar

- Parado

e os eventos que poderão fazer com que o Objecto mude de estado são

- O objecto é criado
- O objecto atende a mensagem
- Um cliente pede uma ligação pela rede
- Um cliente termina um pedido
- O pedido é executado e terminado
- O objecto recebe a mensagem parar
- etc



*O Umbrello UML Modeller a mostrar um Diagrama de Estado*

### 2.2.5.1 Estado

Os estados são os blocos de construção dos Diagramas de Estado. Um estado pertence a exactamente uma classe e representa um resumo dos valores que os atributos de uma classe poderão obter. Um Estado no UML descreve o estado interno de um objecto de uma determinada classe

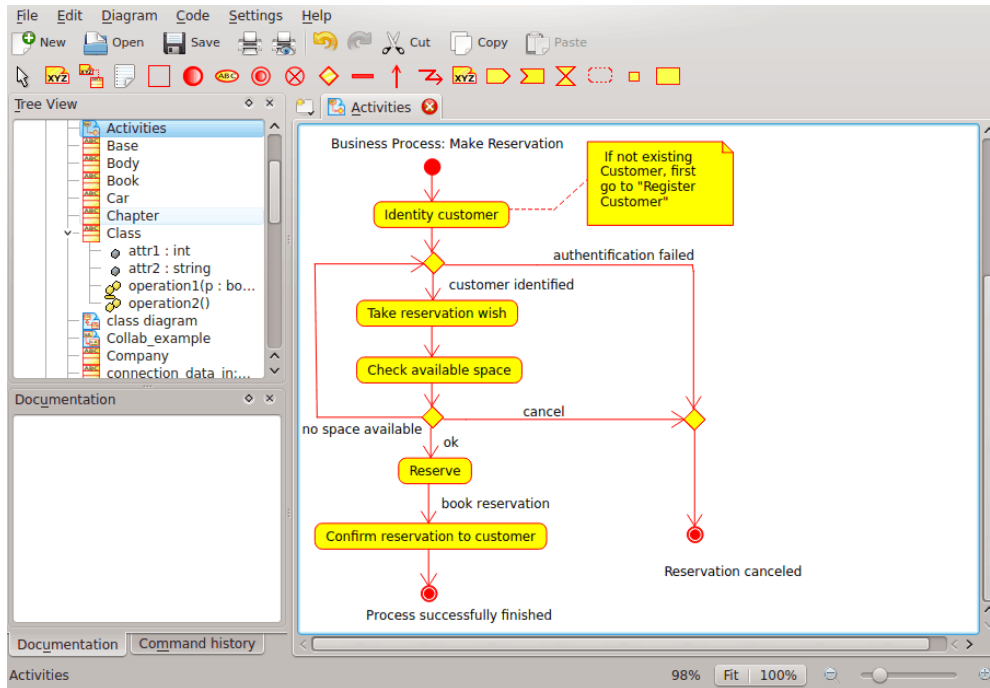
Tenha em atenção que nem todas as alterações de um atributo de um objecto deverão ser representadas por um estado, mas só mesmo aquelas alterações que poderão afectar significativamente o funcionamento do objecto

Existem dois tipos especiais de Estados: o Inicial e o Final. Eles são especiais na medida em que não existe nenhum evento que possa fazer um objecto voltar ao seu estado inicial, da mesma forma que não há nenhum evento que possa retirar um objecto do seu estado final, logo que o tenha atingido.



## 2.2.6 Diagrama de Actividades

Os Diagramas de Actividades descrevem a sequência de actividades de um sistema, com a ajuda das Actividades propriamente ditas. Os Diagramas de Actividades são um tipo especial de Diagramas de Estados, que só (ou em grande parte) contêm Actividades.



*O Umbrello UML Modeller a mostrar um Diagrama de Actividade*

Os Diagramas de Actividades são semelhantes aos fluxogramas procedimentais, com a diferença que todas as Actividades estão claramente associadas a objectos.

Os Diagramas de Actividades estão sempre associados a uma *Classe*, uma *Operação* ou um *Caso de Uso*.

Os Diagramas de Actividades suportam as Actividades sequências, assim como as paralelas. A execução paralela é representada através de ícones de 'Fork' (Bifurcação) ou 'Wait' (Espera) e, para as actividades que decorrem em paralelo, não é importante a ordem pela qual são desempenhadas (elas poderão ser executadas ao mesmo tempo ou uma a seguir à outra)

### 2.2.6.1 Actividade

Uma Actividade é um passo único num processo. Uma Actividade é um estado no sistema com actividade interna e, pelo menos, uma transição de saída. As Actividades também poderão conter mais do que uma transição de saída se tiverem diferentes condições.

As Actividades poderão formar hierarquias, o que significa que uma Actividade poderá ser composta por várias Actividades de 'detalhe', onde nesse caso as transições de entrada e de saída deverão corresponder às transições de entrada e de saída do diagrama de detalhe.

## 2.2.7 Elementos Auxiliares

Existem alguns elementos no UML que não têm nenhum valor semântico real para o modelo, mas que ajudam a clarificar partes do diagrama. Esses elementos são

- Linhas de texto

- Notas de Texto e âncoras
- Caixas

As linhas de texto são úteis para adicionar algumas informações textuais breves para um diagrama. Correspondem a texto livre e não têm nenhum significado para o modelo em si.

As notas são úteis para adicionar informações mais detalhada sobre um objecto ou uma dada situação. Têm a grande vantagem de poderem ser anexadas a Elementos de UML para mostrar que a nota 'pertence' a um dado objecto ou situação.

As caixas são rectângulos livres que poderão ser usados para agrupar os itens em conjunto para tornar os diagramas mais legíveis. Elas não têm significado lógico no modelo.

### 2.2.8 Diagramas de Componentes

Os Diagramas de Componentes mostram os componentes do 'software' (sejam referentes a tecnologias de componentes como os KParts, os componentes do CORBA ou Java Beans, ou mesmo secções do sistema que sejam claramente distintas), bem como os artefactos de que são compostos, como os ficheiros de código-fonte, as bibliotecas de programação ou as tabelas de bases de dados relacionais.

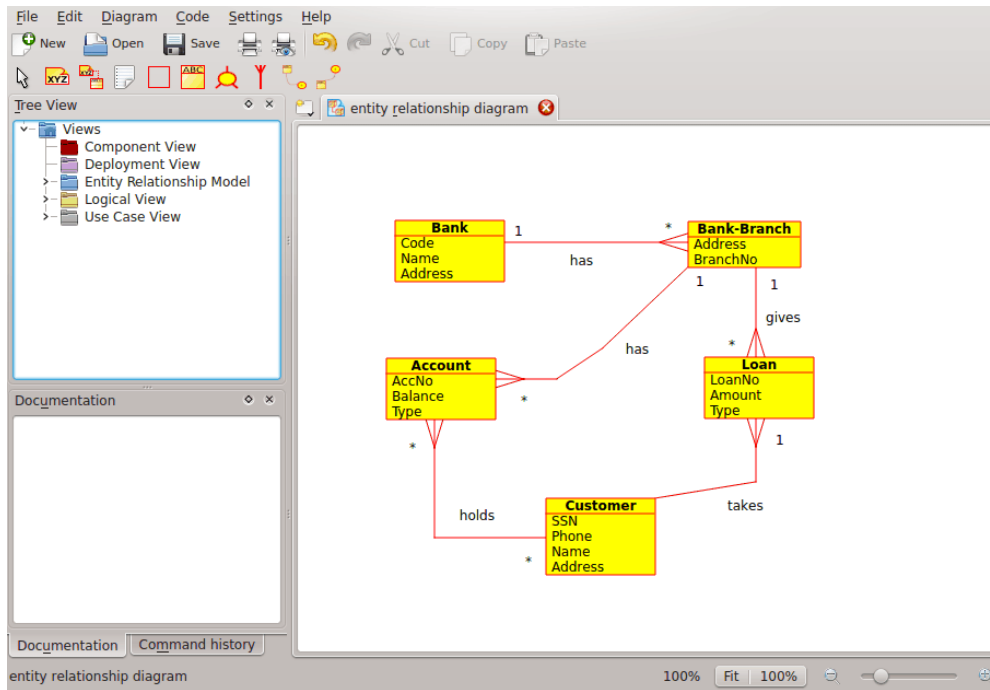
Os componentes poderão ter interfaces (isto é classes abstractas com operações) quer permitem associar os componentes.

### 2.2.9 Diagramas de Entrada em Produção

Os Diagramas de Entrada em Produção mostram as instâncias em execução, bem como as suas associações. Eles incluem os Nós, que são recursos físicos, correspondendo tipicamente a um único computador. Mostram também as interfaces e os objectos (as instâncias das classes).

### 2.2.10 Diagramas de Entidade-Associação

Os Diagramas de Entidade-Associação (Diagramas ER) mostram o desenho conceptual das aplicações de bases de dados. Estes definem as várias entidades (conceitos) no sistema de informação e as relações e restrições entre eles. Uma extensão dos Diagramas Entidade-Associação, chamada 'Diagramas Estendidos Entidade-Associação' (EER) são usados para incorporar as técnicas de desenho Orientado por Objectos nos diagramas ER.



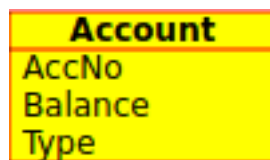
O Umbrello a mostrar um Diagrama de Entidade-Associação

### 2.2.10.1 Entidade

Uma *Entidade* é qualquer conceito no mundo real com uma existência independente. Poderá ser um objecto com uma existência física (exemplo, Computador, Robot) ou poderá ser um objecto com uma existência conceptual ( p.ex.: Curso Universitário). Cada entidade tem um conjunto de atributos que descrevem as propriedades da Entidade.

*Nota:* Não existem ainda notações normalizadas para desenhar Diagramas ER. Os diferentes textos sobre este assunto usam notações diferentes. Os conceitos e notações para os diagramas ER usados no Umbrello são do seguinte livro: *Elmasri R. e Navathe S. (2004). Fundamentals of Database Systems 4a ed. Addison Wesley*

Num Diagrama ER, as Entidades são representadas por rectângulos com o nome da classe, e poderão também mostrar os atributos de uma entidade noutra 'compartimento' dentro do rectângulo.



Representação visual de uma entidade num Diagrama ER

#### 2.2.10.1.1 Atributos da Entidade

Nos Diagramas ER, os Atributos das Entidades aparecem com o seu nome num compartimento diferente da Entidade a que pertencem.

#### 2.2.10.1.2 Restrições

As restrições nos diagramas ER definem as limitações sobre os dados no esquema de informação.

Existem quatro tipos de restrições suportados no Umbrello :

- *Chave Primária*: O conjunto de atributos declarado como *chave primária* é único para a entidade. Só poderá existir uma chave primária numa Entidade e nenhum dos seus atributos constituintes poderá ser nulo.
- *Chave Única*: O conjunto de atributos declarado como *chave única* é único para a entidade. Poderão existir várias restrições de chaves únicas. Os seus atributos constituintes poderão ser nulos. As Chaves Únicas e Chaves Primárias identificam de forma única uma linha numa tabela (entidade)
- *Chave Estrangeira*: Uma Chave Estrangeira é uma restrição de referência entre duas tabelas. A chave estrangeira identifica uma coluna ou conjunto de colunas numa tabela (de referência) que aponta para uma coluna ou conjunto de colunas noutra tabela (referenciada). As colunas da tabela referenciada deverão formar uma chave primária ou única.
- *Restrição de Verificação*: Uma restrição de verificação (também conhecida como restrição de verificação de tabelas) é uma condição que define os dados válidos ao adicionar ou actualizar um item numa tabela de uma base de dados relacional. Uma restrição de verificação é aplicada a cada linha na tabela. A tabela deverá ser um predicado. Poderá fazer uma referência a uma ou várias colunas da tabela.  
Exemplo: preço >= 0

## 2.2.11 Conceitos Estendidos dos Diagramas de Entidade-Associação (ER)

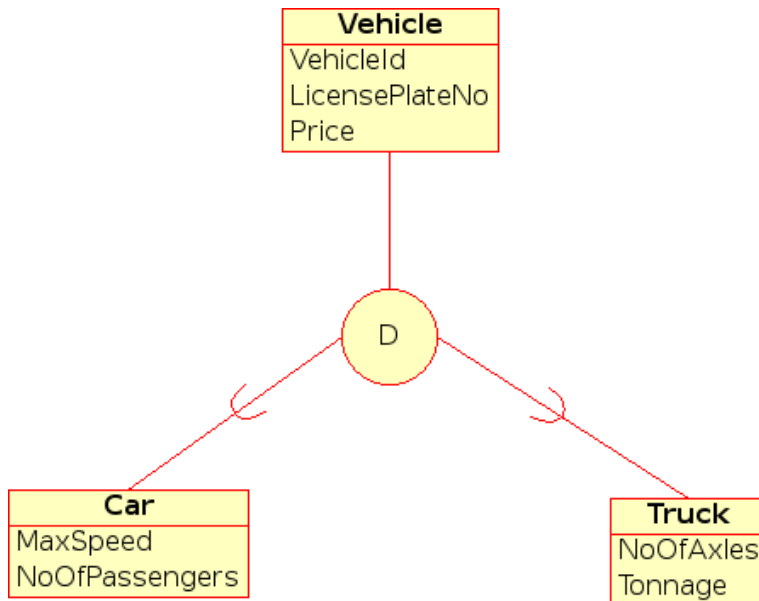
### 2.2.11.1 Especialização

A especialização é uma forma de criar novas entidades com base em entidades que já tenham sido definidas. As entidades novas, conhecidas como entidades derivadas, recebem (ou herdam) os atributos das entidades pré-existentes, as quais são referenciadas como entidades de base. Pretende ajudar a reutilizar os dados existentes com poucas ou nenhuma modificação.

No Umbrello, uma pessoa poderá definir Especializações Disjuntas e Sobrepostas

#### 2.2.11.1.1 Especialização Disjunta

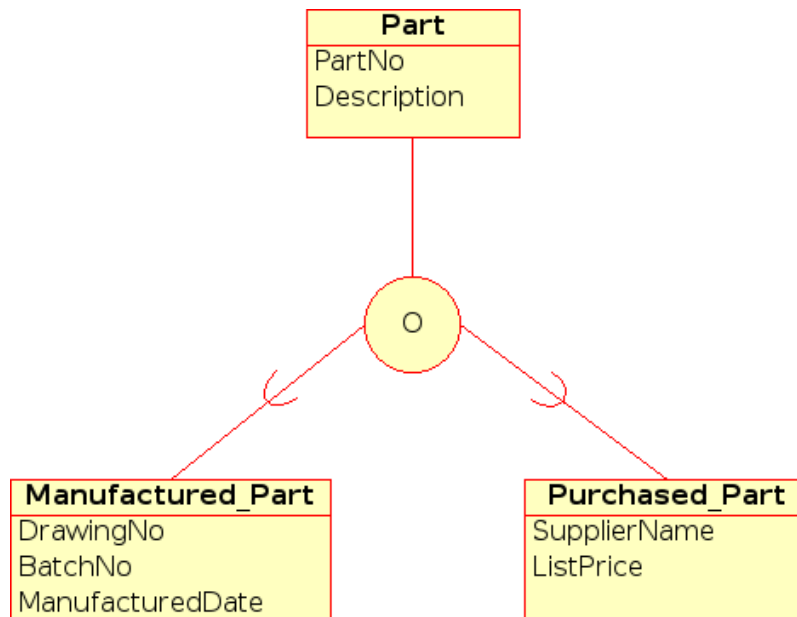
A Especialização Disjunta define que as sub-classes da especialização deverão ser disjuntas. Isto significa que uma entidade poderá ser um membro de no máximo uma das entidades derivadas da especialização



Representação visual de uma Especialização Disjunta num Diagrama ER

### 2.2.11.1.2 Especialização Sobreposta

Quando as entidades derivadas não estão restritas para serem disjuntas, o seu conjunto de entidades são consideradas uma especialização sobreposta. Isto significa que a mesma entidade no mundo real poderá ser um membro de mais de uma entidade derivada da especialização

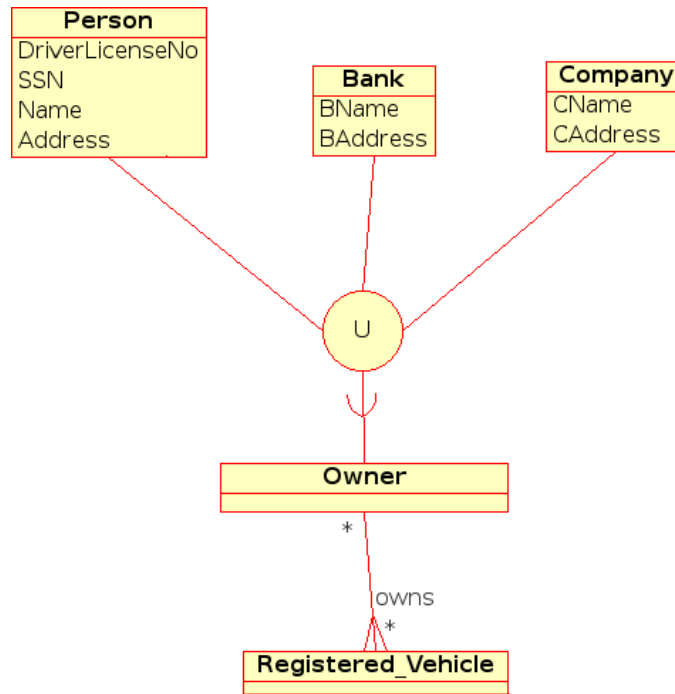


Representação visual de uma Especialização Sobreposta num Diagrama ER

### 2.2.11.1.3 Categoria

Diz-se que uma entidade derivada é uma *Categoria* quando representa uma colecção de objectos que é um sub-conjunto da União dos tipos de entidades distintos. Uma categoria modela-se

quando surge a necessidade de uma relação única de super-classe/sub-classe com mais de uma super-classe, onde as super-classes representam diferentes tipos de entidades (como acontece na herança múltipla da Programação Orientada por Objectos ).



Representação visual de uma categoria num Diagrama ER

## Capítulo 3

# Trabalhar com o Umbrello UML Modeller

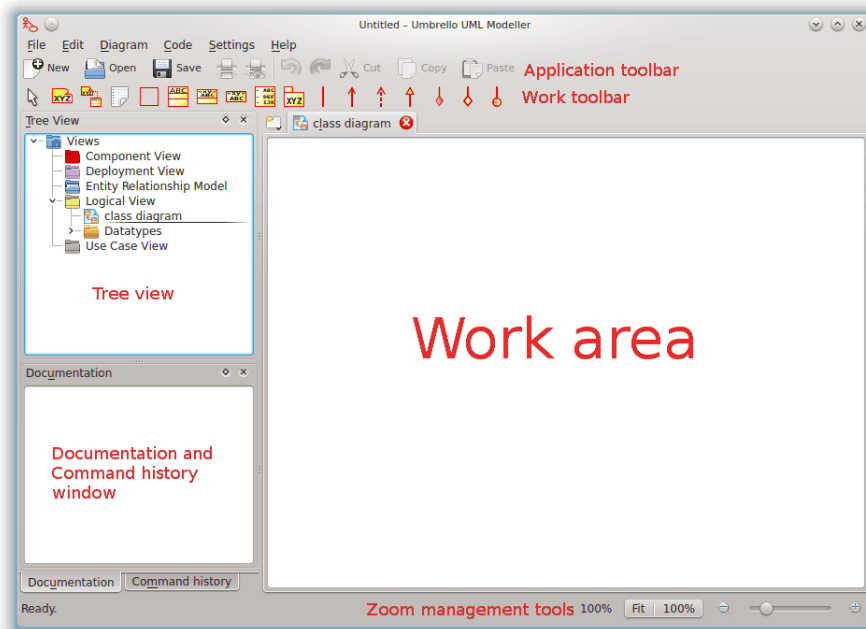
Este capítulo introduzi-lo-á à interface de utilizador do Umbrello UML Modeller e indicar-lhe-á o que precisa de saber para começar a modelar. Todas as acções no Umbrello UML Modeller são acedidas através do menu e das barras de ferramentas, mas o Umbrello UML Modeller também recorre com frequência aos menus de contexto do botão direito do rato. Você poderá carregar com o botão direito do rato em praticamente qualquer elemento da área de trabalho ou da árvore do Umbrello UML Modeller para obter um menu com as funções mais úteis que poderão ser aplicadas ao elemento em particular com que está a lidar. Alguns dos utilizadores poderão achar um pouco confuso ao início porque estão mais habituados a trabalhar com o menu ou com as barra de ferramentas, mas logo que esteja habituado a carregar com o botão direito irá aumentar bastante a velocidade do seu trabalho.

### 3.1 Interface do Utilizador

A janela principal do Umbrello UML Modeller está dividida em três áreas que o ajudarão a manter uma ideia geral do seu sistema inteiro e a aceder mais depressa aos diferente diagramas enquanto trabalha no seu modelo.

Estas áreas são chamadas:

- Vista em Árvore
- Área de Trabalho
- Janela de histórico de comandos e documentação



*Interface de Utilizador do Umbrello UML Modeller*

### 3.1.1 Vista em Árvore

A Vista em Árvore, ou simplesmente Árvore, está localizada no lado esquerdo e na parte superior da janela, e mostra todos os diagramas, classes, actores e casos de utilização que compõem o seu modelo. A vista em árvore também lhe dá uma forma rápida de alternar entre os diferentes diagramas do seu modelo e de inserir elementos do seu modelo no diagrama actual.

Se você estiver a trabalhar num modelo com bastantes classes e diagramas, a Árvore poderá ajudá-lo a ter a noção das coisas, organizando os elementos do seu modelo em pastas. Você poderá criar pastas se seleccionar a opção apropriada no menu de contexto (se carregar com o botão direito do rato numa das pastas da árvore) e poderá organizar os seus elementos, movendo-os para a pasta apropriada (arrastando-os e largando-os)

### 3.1.2 Janela de Histórico de Comandos e Documentação

A Janela de Histórico de Comandos e Documentação é a pequena janela que se localiza em baixo e à esquerda na janela do Umbrello UML Modeller, e dá-lhe uma antevisão rápida da documentação para o item seleccionado no momento. A Janela de Documentação é relativamente pequena porque pretende dar-lhe apenas uma ideia rápida da documentação do elemento, mas ocupando o mínimo espaço no ecrã possível. Se você precisar de ver a documentação com mais detalhe, poderá sempre abrir as propriedades do item.

### 3.1.3 Área de Trabalho

A Área de Trabalho é a janela principal do Umbrello UML Modeller e onde a acção real toma lugar. Você usa a Área de Trabalho para editar e ver os diagramas do seu modelo. A Área de Trabalho mostra o diagrama activo no momento. Actualmente só é possível mostrar um diagrama na Área de Trabalho em qualquer altura.



## 3.2 Criar, Carregar e Gravar Modelos

A primeira coisa que você precisa para começar a fazer algo de útil com o Umbrello UML Modeller é criar um modelo sobre o qual trabalhar. Quando você inicia o Umbrello UML Modeller ele carregar sempre o último modelo usado ou cria um novo modelo em branco (dependendo das suas opções definidas na janela de configuração). Isto permitir-lhe-á começar a trabalhar logo num instante.

### 3.2.1 Novo Modelo

Se a qualquer altura você precisar de criar um novo modelo, poderá fazê-lo seleccionando para tal o item **Novo** do menu **Ficheiro**, ou carregando no ícone **Novo** da barra de ferramentas da aplicação. Se você estiver a trabalhar num modelo que foi modificado, o Umbrello UML Modeller perguntar-lhe-á se deverá gravar as suas alterações antes de carregar o novo modelo.

### 3.2.2 Gravar Modelo

Você poderá gravar o seu modelo em qualquer altura se seleccionar a opção **Gravar** do menu **Ficheiro** ou carregando no botão **Gravar** da barra de ferramentas da aplicação. Se precisar de gravar o seu modelo com outro nome, você poderá usar a opção **Gravar Como** do menu **Ficheiro**.

Para sua conveniência, o Umbrello UML Modeller também lhe oferece a possibilidade de gravar automaticamente o seu trabalho com um período de tempo determinado. Você poderá configurar se deseja esta opção, assim como definir os intervalos de tempo na **Configuração** do Umbrello UML Modeller

### 3.2.3 Carregar Modelo

Para carregar um modelo já existente, você poderá seleccionar a opção **Abrir** do menu **Ficheiro** ou ainda carregar no ícone **Abrir** da barra de ferramentas da aplicação. Os modelos usados mais recentemente estão disponíveis no sub-menu **Abrir Recente** do menu **Ficheiro** para acelerar o acesso aos seus modelos usados mais frequentemente.

O Umbrello UML Modeller só poderá lidar com um modelo de cada vez, por isso se pedir ao programa para carregar um modelo para si, e se o seu modelo actual tiver sido modificado desde a última vez que o gravou, o Umbrello UML Modeller irá perguntar se você deseja gravar as suas alterações para evitar que perca o trabalho. Você poderá iniciar duas ou mais instâncias do Umbrello UML Modeller em qualquer altura, assim como também poderá copiar e colar entre instâncias.

## 3.3 Editar Modelos

No Umbrello UML Modeller, existem basicamente duas formas de editar os elementos do seu modelo.

- Editar directamente os elementos do modelo na Árvore
- Editar os elementos do modelo através de um diagrama

Ao usar o menu de contexto dos diferentes itens da Árvore você será capaz de adicionar, remover e modificar quase todos os elementos do seu modelo. Se carregar com o botão direito nas pastas da Árvore, você ficará com algumas opções para criar os diferentes tipos de diagramas assim

como, dependendo se a pasta pertence a uma *Vista de um Caso de Uso* ou de uma *Vista Lógica*, a Actores, Casos de Utilização, Classes, etc.

Logo que você tenha adicionado elementos ao seu modelo, você poderá também editar um elemento se aceder à sua janela de propriedades, a qual poderá encontrar se seleccionar a opção *Propriedades* do menu de contexto que é mostrado quando se carrega com o botão direito nos itens da Árvore.

Você também poderá editar o seu modelo, criando ou modificando elementos através dos diagramas. Poderão ser obtidos mais detalhes sobre como fazê-lo nas secções seguintes.

## 3.4 Adicionar e Remover Diagramas

O seu modelo UML consiste num conjunto de elementos UML e de associações entre eles. Todavia, você não consegue ver o modelo directamente, como tal você usa os *Diagramas* para olhar para ele.

### 3.4.1 Criar Diagramas

Para criar um novo diagrama no seu modelo, basta seleccionar o tipo de diagrama que precisa no sub-menu **Novo** do menu **Diagrama** e dar-lhe um nome. O diagrama será criado e tornado activo, e você vê-lo-á imediatamente na árvore.

Lembre-se que o Umbrello UML Modeller tira bastante partido dos menus de contexto: você também poderá carregar com o botão direito do rato numa pasta da Árvore e seleccionar o tipo de diagrama apropriado no sub-menu **Novo** do menu de contexto. Lembre-se que você poderá criar Diagramas de Casos de Uso apenas na Vista de Casos de Uso, enquanto que os outros tipos de diagramas só poderão ser criados nas pastas das Vistas Lógicas.

### 3.4.2 Remover Diagramas

Se você necessitar de remover um diagrama do seu modelo deve torná-lo activo e seleccionar a opção **Remover** do menu **Diagrama**. Você também poderá conseguir isto se seleccionar **Remover** do menu de contexto dos diagramas na árvore

Dado que a remoção de um diagrama é algo sério que poderá dar origem à perda de algum trabalho se for feito por acidente, o Umbrello UML Modeller perguntar-lhe-á se confirma a operação de remoção antes de remover efectivamente o diagrama. Assim que um diagrama tenha sido removido e o ficheiro tenha sido gravado não existe forma de anular essa acção.

### 3.4.3 Mudar o Nome a Diagramas

Se quiser mudar o nome de um diagrama existente, você-lo-á fazer facilmente se seleccionar a opção Mudar o Nome do seu menu do botão direito do rato da árvore.

Outra forma de mudar o nome de um diagrama é através da sua janela de propriedades, a qual poderá aceder se seleccionar Propriedades do seu menu de contexto ou se carregar duas vezes na árvore.

## 3.5 Editar Diagramas

Ao trabalhar num diagrama, o Umbrello UML Modeller irá tentar guiá-lo na aplicação de algumas regras simples como quais os elementos que são válidos nos diferentes tipos de diagramas,

assim como as relações entre eles. Se você for um perito em UML poderá até nem dar por isso, mas isto poderá ajudar as pessoas menos experientes em UML a criar diagramas em conformidade com as normas.

Logo que você tenha criado os seus diagramas, é a altura de começar a editá-los. Aqui você deverá reparar na diferença (subtil, para os principiantes) entre a edição do seu diagrama e a edição do *modelo*. Como já deverá saber, os diagramas são *vistas* do seu modelo. Por exemplo, se você criar uma classe ao editar um Diagrama de Classes, você está de facto a alterar ambos, o seu diagrama e o seu modelo. Se você alterar a cor ou outras opções de visualização de uma classe no seu Diagrama de Classes, você só estará a editar o diagrama, mas nada se altera no seu modelo.

### 3.5.1 Inserir Elementos

Uma das primeiras coisas que você irá fazer quando estiver a editar um diagrama novo é inserir elementos nele (Classes, Actores, Casos de Uso, etc.). Existem basicamente duas formas de o fazer:

- Arrastar os elementos existentes no seu modelo a partir da árvore
- Criando novos elementos no seu modelo e adicionando-os ao seu diagrama ao mesmo tempo, usando uma das ferramentas de edição da Barra de Trabalho

Para inserir elementos que já existam no seu modelo, basta arrastá-los da árvore e largá-los onde os desejar no seu diagrama. Você poderá sempre mover os elementos do seu diagrama com a ferramenta de selecção.

A segunda forma de adicionar elementos ao seu diagrama é usando as ferramentas de edição da Barra de Trabalho (repare que isto também irá adicionar os elementos ao seu modelo).

A Barra de Trabalho localizava-se por omissão no topo da janela. As ferramentas disponíveis nesta barra de ferramentas (os botões que vê nela) mudam de acordo com o tipo de diagrama com que está a lidar no momento. O botão para a ferramenta seleccionada no momento está activo na barra de ferramentas. Você poderá mudar para a ferramenta de selecção se carregar na tecla **Esc**.

Quando você tiver seleccionado uma ferramenta de edição da Barra de Ferramentas de Trabalho (como, por exemplo, a ferramenta de inserção de classes), o cursor do rato muda para uma cruz, e você poderá inserir os elementos no seu modelo se carregar uma vez no seu diagrama. Repare que os elementos no UML precisam de ter um *Nome Único*. Por isso, se você tiver uma classe num diagrama cujo nome é 'ClasseA' e se você quiser usar a ferramenta de inserção de classes para inserir uma classe noutra diagrama, você não poderá chamar a essa nova classe também 'ClasseA'. Se estes dois nomes são supostamente para estar em dois elementos diferentes, você precisa de dar a eles um nome único. Se você estiver a tentar adicionar o *mesmo* elemento ao seu diagrama, então a ferramenta de inserção de classes não é a melhor ferramenta para isso. Você deverá arrastar e largar a classe a partir da árvore, em alternativa.

### 3.5.2 Apagar Elementos

Você poderá remover qualquer elemento se seleccionar a opção **Remove** do seu menu de contexto.

Mais uma vez, existe uma *grande* diferença entre remover um objecto de um diagrama e remover um objecto do seu modelo. Se você remover um objecto de um diagrama, só o estará a remover desse diagrama em particular: o elemento continuará a fazer parte do seu modelo e, se existirem outros diagramas que usem o mesmo elemento, os mesmos não sofrerão qualquer alteração. Se, por outro lado, você remover o elemento da Árvore, você está efectivamente a remover o elemento do seu *modelo*. Dado que o elemento já não existe mais no seu modelo, ele será removido automaticamente de todos os diagramas em que aparece.

### 3.5.3 Editar Elementos

Você poderá editar a maioria dos elementos de UML do seu modelo e dos diagramas se abrir a sua janela respectiva de Propriedades e se seleccionar as opções apropriadas. Para editar as propriedades de um objecto, selecione as **Propriedades** no seu menu de contexto (com o botão direito do rato). Cada elemento tem uma janela que consiste em várias páginas onde você poderá configurar as opções correspondentes a esse elemento. Para alguns elementos, como os actores, você define poucas acções, mas para outros como as classes, você poderá editar os seus atributos e operações, seleccionar o que deseja mostrar no diagrama (a assinatura da operação ou apenas os seus nomes, etc.) e até as cores que deseja usar para as linhas e para o preenchimento da representação da classe no diagrama.

Para os elementos de UML, também poderá abrir a janela de propriedades, fazendo duplo-click sobre ela, caso esteja a usar a ferramenta de selecção (seta).

Repare que você também poderá seleccionar a opção de propriedades no menu de contexto dos elementos na árvore. Isto permite-lhe também editar as propriedades dos diagramas, como a definição da visibilidade da grelha ou não.

### 3.5.4 Editar Classes

Ainda que a edição das propriedades de todos os objectos já tenha sido falada na secção anterior, as classes merecem uma secção especial porque são um pouco mais complicadas e têm mais opções que a maioria dos outros elementos de UML.

Na janela de propriedades de uma classe você poderá configurar tudo, desde a cor que usa até às operações e atributos que tem.

#### 3.5.4.1 Configuração Geral das Classes

A página de Opções Gerais da janela de propriedades é auto-compreensível. Aqui você poderá alterar o nome da classe, a visibilidade, a documentação, etc.. Esta página está sempre disponível.

#### 3.5.4.2 Configuração dos Atributos das Classes

Na página de Opções dos Atributos você poderá adicionar, editar ou remover atributos (variáveis) da classe. Você poderá subir ou descer os atributos na lista se carregar nas setas ao lado. Esta página está sempre disponível.

#### 3.5.4.3 Configuração das Operações das Classes

De forma semelhante à página das Opções dos Atributos, na página de Opções das Operações você poderá adicionar, editar ou remover operações da sua classe. Ao adicionar ou editar uma operação, você indica os dados básicos na janela de *Propriedades da Operação*. Se você quiser adicionar parâmetros à sua operação você precisa de carregar no botão **Novo Parâmetro**, o qual irá mostrar a janela de *Propriedades do Parâmetro*. Esta página está sempre disponível

#### 3.5.4.4 Configuração do Modelo de Classe

Esta página permite-lhe adicionar modelos de classes, os quais são classes ou tipos de dados não especificados. No Java 1.5 chamar-se-ão 'Generics' (Genéricos).

#### 3.5.4.5 Página de Associações da Classe

A página **Associações da Classe** mostra todas as associações desta classe no diagrama actual. Se carregar duas vezes numa associação irá mostrar as suas propriedades e, dependendo do tipo de associação, você poderá modificar alguns parâmetros, como por exemplo a multiplicidade ou o nome do Papel. Se a associação não permitir que essas opções sejam modificadas, a janela de Propriedades da Associação fica apenas para leitura e só poderá modificar a documentação referente a esta associação.

Esta página só está disponível se você abrir as Propriedades da Classe dentro de um diagrama. Se você seleccionar as propriedades da classe no menu de contexto da árvore, esta página não estará disponível.

#### 3.5.4.6 Página de Visualização de Classes

Na página de **Opções de Visualização**, você poderá configurar o que deve ser mostrado no diagrama. Uma classe poderá ser mostrada apenas como um rectângulo com o nome da classe nele (útil se você tiver várias classes no seu diagrama, ou se não estiver interessado de momento nos detalhes de cada classe) ou então completo, mostrando por exemplo os pacotes, estereótipos, atributos e operações com a assinatura e visibilidade completas

Dependendo da quantidade de informação que você deseja ver, você poderá seleccionar as opções correspondentes nesta página. As alterações que fizer aqui são apenas *opções de apresentação* para o diagrama. Isto significa que, se 'esconder' as operações de uma classe, só fará com que elas não sejam apresentadas no diagrama, mas estas mantêm-se à mesma como parte do seu modelo. Esta opção só está disponível se você seleccionar as propriedades de uma classe dentro de um diagrama. Se você abrir as propriedades de uma classe a partir da árvore, esta página não aparece, porque essas opções de apresentação não farão sentido nesse caso

#### 3.5.4.7 Página de Estilo da Classe

Na página **Estilo do Elemento** você poderá configurar as cores que deseja usar para as linhas e para o enchimento do elemento. Esta opção só faz sentido obviamente para as classes mostradas nos diagramas e não aparece se você abrir a janela de propriedades da classe na árvore.

### 3.5.5 Associações

As associações relacionam dois objectos UML um com o outro. Normalmente as associações são definidas entre duas classes, mas alguns tipos de associações poderão também existir entre casos de uso e actores.

Para criar uma associação, seleccione a ferramenta apropriada na Barra de Trabalho (uma Associação genérica, uma Generalização, uma Agregação, etc.), e carregue no primeiro elemento que participa na associação, carregando depois no segundo item participante. Repare que deve carregar duas vezes, uma em cada objecto que participa na associação, e *não* efectuar um arrastamento de um objecto para outro.

Se tentar usar uma associação de um forma que não respeite a especificação do UML, o Umbrello UML Modeller recusar-se-á a criar a associação e você irá obter uma mensagem de erro. Este seria o caso se, por exemplo, existisse uma Generalização da classe A para a classe B e você quisesse criar outra generalização da classe B para a classe A

Se carregar com o botão direito numa associação, irá aparecer um menu de contexto com as acções que poderá aplicar nela. Se precisar de apagar uma associação, basta seleccionar a opção **Remover** deste menu de contexto. Você também poderá seleccionar a opção **Propriedades** e, dependendo do tipo de associação, editar os atributos como os papéis e a multiplicidade.

### 3.5.5.1 Pontos-Âncora

As associações são desenhadas, por omissão, como uma linha a direito a ligar os dois objectos no diagrama.

Você poderá adicionar pontos de âncora se fizer carregar duas vezes em qualquer ponto da linha da associação. Isto irá introduzir um ponto de âncora (representado como um ponto azul, quando a linha da associação fica seleccionada) que você poderá mover para dar forma à associação

Se precisar de remover um ponto de âncora, carregue duas vezes nele para o remover

Repare que a única forma de editar as propriedades de uma associação é através do menu de contexto. Se você tentar carregar duas vezes nele como faz com outros objectos UML, isto irá apenas introduzir um ponto de âncora.

### 3.5.6 Notas, Texto e Caixas

As Notas, as Linhas de Texto e as Caixas são elementos que poderão estar presentes em qualquer tipo de diagrama e não têm nenhum valor semântico, mas são muito úteis para adicionar comentários ou explicações-extra que possam tornar o seu diagrama mais simples de entender.

Para adicionar uma Nota ou uma Linha de Texto, seleccione a ferramenta correspondente na Barra de Ferramentas de Trabalho e carregue no diagrama onde deseja colocar o seu comentário. Você poderá editar o texto se abrir o elemento com o seu menu de contexto ou, no caso das notas, se carregar duas vezes nelas.

#### 3.5.6.1 Âncoras

As âncoras são usadas para ligar uma nota de texto com outro elemento de UML em conjunto. Por exemplo, você usa normalmente uma nota de texto para explicar ou fazer algum comentário acerca de uma classe ou de uma associação em particular onde, nesse caso, você poderá usar a âncora para clarificar que a nota 'pertence' a esse elemento em particular.

Para adicionar uma âncora entre uma nota e outro elemento de UML, use a ferramenta da âncora na barra de ferramentas de trabalho. Você precisa de carregar primeiro na nota e depois no elemento de UML a que deseja ligar a nota.

## Capítulo 4

# Importação e Geração de Código

O Umbrello UML Modeller é uma ferramenta de modelação de UML e, como tal, o seu objectivo principal é ajudá-lo na *análise e desenho* dos seus sistemas. Contudo, para fazer a transição entre o seu desenho e a sua *implementação*, o Umbrello UML Modeller permite-lhe gerar código-fonte em várias linguagens de programação diferentes para o preparar a si. Também, se você quiser começar a usar o UML num projecto já iniciado de C++, o Umbrello UML Modeller pode ajudá-lo a criar um modelo do seu sistema a partir do código-fonte, analisando-o e importando as classes que forem encontradas nele.

### 4.1 Geração de Código

O Umbrello UML Modeller pode gerar código para várias linguagens de programação, baseando-se no seu Modelo UML para o ajudar a iniciar a implementação do seu projecto. O código gerado consiste nas declarações das classes, com os seus métodos e atributos, de modo a que você possa ‘preencher as lacunas’, fornecendo a funcionalidade das operações das suas classes.

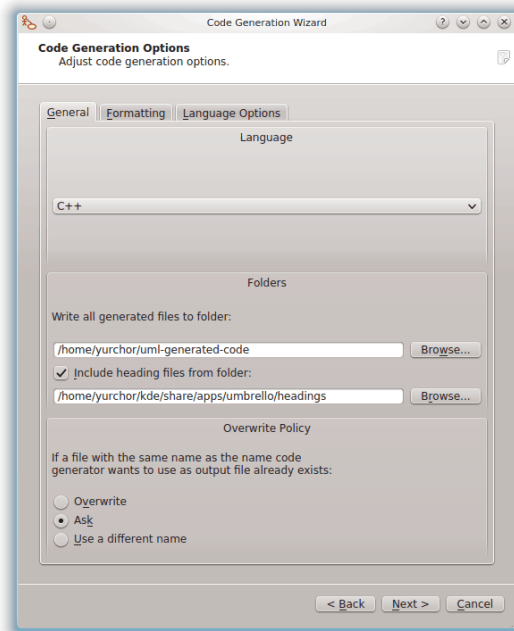
O Umbrello UML Modeller 2 vem com o suporte de geração de código para ActionScript, Ada, C++, C#, D, IDL, Java™, JavaScript, MySQL e Pascal.

#### 4.1.1 Gerar Código

Para poder gerar código com o Umbrello UML Modeller, você precisa primeiro de criar ou carregar um modelo que contenha pelo menos uma classe. Quando estiver pronto para começar a escrever algum código, seleccione a opção **Assistente de Geração de Código** do menu **Código** que o irá guiar no processo de geração de código.

O primeiro passo é seleccionar as classes para as quais deseja gerar código-fonte. Por omissão, todas as classes do seu modelo estão seleccionadas e remover as que não quer gerar, passando-as para a lista do lado esquerdo.

O próximo passo do assistente permite-lhe modificar os parâmetros que o Gerador de Código usa ao criar o seu código. Estão disponíveis as seguintes opções:



*Opções de Geração de Código no Umbrello UML Modeller*

#### 4.1.1.1 Opções de Geração

##### 4.1.1.1.1 Detalhe dos Comentários

A opção **Escrever os comentários de documentação mesmo que vazios** diz ao Gerador de Código para criar os comentários do tipo `/** texto */`, mesmo que os blocos de comentários estejam vazios. Se você tiver adicionado documentação às suas classes, métodos ou atributos do seu modelo, o Gerador de Código irá escrever esses comentários como documentação do Doxygen, independentemente do que tenha indicado aqui, mas se você seleccionar esta opção, o Umbrello UML Modeller irá escrever os blocos de comentários para todas as classes, métodos e atributos, mesmo que não exista nenhuma documentação no modelo, onde neste caso deverá documentar as suas classes mais tarde no próprio código-fonte.

A opção **Escrever os comentários para as secções mesmo que estas estejam vazias** faz com que o Umbrello UML Modeller escreva os comentários no código-fonte para delimitar as várias secções de uma classe. Por exemplo, os 'métodos públicos' ou os 'Atributos' antes das secções correspondentes. Se você seleccionar esta opção, o Umbrello UML Modeller irá escrever os comentários de todas as secções da classe, mesmo que a secção esteja vazia. Por exemplo, iria escrever um comentário a dizer 'métodos protegidos', mesmo que estes não existam na sua classe.

##### 4.1.1.1.2 Pastas

**Criar todos os ficheiros gerados na pasta.** Aqui você deverá indicar a pasta onde quer que o Umbrello UML Modeller coloque o código gerado.

A opção **Incluir os ficheiros da pasta** permite-lhe inserir um cabeçalho no início de cada ficheiro gerado. Os ficheiros de cabeçalho poderão conter alguma informação de direitos de cópia ou de licenças ou conterem variáveis que sejam avaliadas na altura da geração. Você poderá dar uma vista de olhos nos ficheiros de cabeçalho-modelos que vêm com o Umbrello UML Modeller para ver como é que poderá usar essas variáveis para substituir o seu nome ou a data actual na altura da geração.



#### 4.1.1.1.3 Política de Sobreposição

Esta opção diz ao Umbrello UML Modeller o que fazer se o ficheiro que ele deseja criar já existir na pasta de destino. O Umbrello UML Modeller *não pode modificar os ficheiros de código existentes*, por isso você terá de optar entre sobrepor o ficheiro existente, ignorar a geração desse ficheiro em particular ou deixar que o Umbrello UML Modeller escolha um nome de ficheiro diferente. Se você escolher a opção para usar um nome diferente, o Umbrello UML Modeller irá adicionar um sufixo ao nome do ficheiro.

#### 4.1.1.1.4 Linguagem

O Umbrello UML Modeller irá gerar por omissão o código para a linguagem que você tenha indicado como Linguagem Activa mas, com o Assistente de Geração de Código, você tem a opção de alterar isto para outra linguagem.

#### 4.1.1.2 Assistente de Geração - Geração

O terceiro e último passo do assistente mostra o estado do processo de Geração de Código. Você só terá de carregar no botão Gerar para ficar com as classes criadas para si.

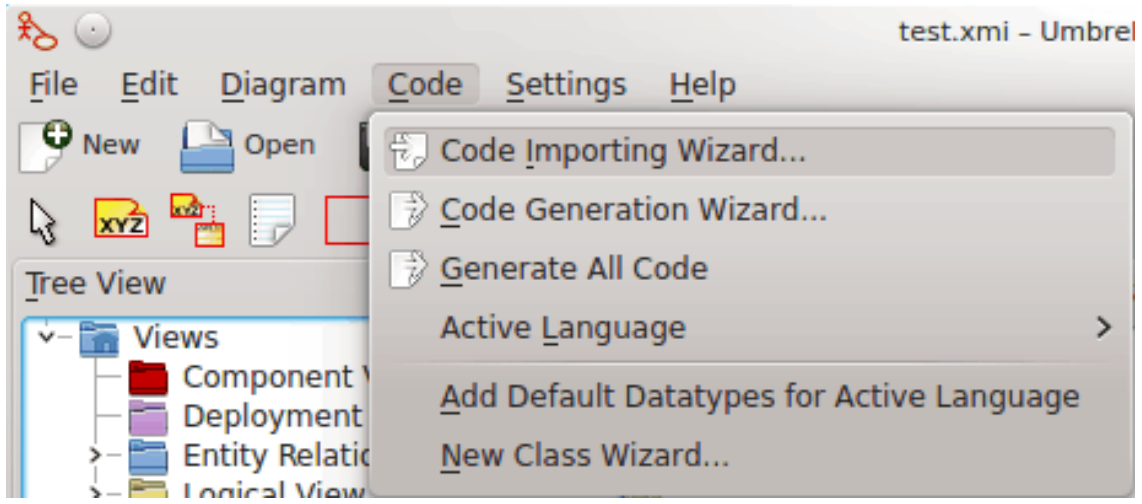
Tenha em atenção que as Opções que seleccionar durante o assistente de Geração de Código só são válidas para a geração actual. Da próxima vez que você executar o Assistente, você terá de seleccionar de novo todas as opções (a sua pasta de cabeçalhos, a política de sobreposição, e assim por diante). Você poderá configurar os valores por omissão usados pelo Umbrello UML Modeller na secção de **Geração de Código** da configuração do Umbrello UML Modeller, a qual está disponível em **Configuração** → **Configurar o Umbrello UML Modeller...**

Se você tiver configurado as suas opções de Geração de Código para os valores correctos e quiser gerar já algum código sem ter de percorrer o assistente, você poderá seleccionar a opção **Gerar Todo o Código** do menu **Código**. Isto irá gerar o código para todas as classes do seu modelo com as opções actuais (incluindo a Pasta do Resultado e a Política de Sobreposição, por isso use isto com cuidado).

## 4.2 Importação de Código

O Umbrello UML Modeller pode importar o código-fonte dos seus projectos existentes para o ajudar a criar o modelo dos seus sistemas. O Umbrello UML Modeller 2 suporta o ActionScript, Ada, C++, c#, D, IDL, Java™, JavaScript, MySQL e Pascal.

Para importar as classes para o seu modelo, seleccione o item **Assistente de Importação de Código...** do menu **Código**. Na janela de ficheiros, seleccione os que contêm os ficheiros com declarações de classes e carregue em **Seguinte >**, seguido de **Iniciar a Importação** e **Terminar**. As classes serão importadas e você achá-las-á como fazendo parte do seu modelo na **Árvore**. Repare que o Umbrello UML Modeller não irá criar nenhuma espécie de diagrama para mostrar as suas classes; elas serão apenas importadas para o seu modelo para que você as possa usar posteriormente em qualquer diagrama que desejar.



*Menu para importar código-fonte no Umbrello UML Modeller*

## Capítulo 5

# Outras Características

### 5.1 Outras Características do Umbrello UML Modeller

Este capítulo irá explicar brevemente algumas outras funcionalidades que o Umbrello UML Modeller lhe oferece.

#### 5.1.1 Copiar objectos como imagens PNG

À parte de lhe oferecer as operações normais de cópia, corte e colagem que estaria à espera para copiar os objectos entre os diferentes diagramas, o Umbrello UML Modeller pode copiar os objectos como imagens PNG de modo a que as possa inserir noutra tipo de documento. Você não precisa de fazer nada de especial para usar esta funcionalidade, bastando para tal seleccionar um objecto de um diagrama (uma Classe, um Actor, etc.) e copiá-lo (**Ctrl-C** ou com o menu), e abra depois um documento do Calligra Words (ou outro programa onde possa colar imagens) e seleccionar **Colar**. Esta é uma grande funcionalidade para exportar partes do seu diagrama como imagens simples.

#### 5.1.2 Exportar para uma Imagem

Você também poderá exportar um diagrama completo como uma imagem. A única coisa que terá de fazer é seleccionar o diagrama que deseja exportar, e seleccionar a opção **Exportar como Imagem...** no menu **Diagrama**.

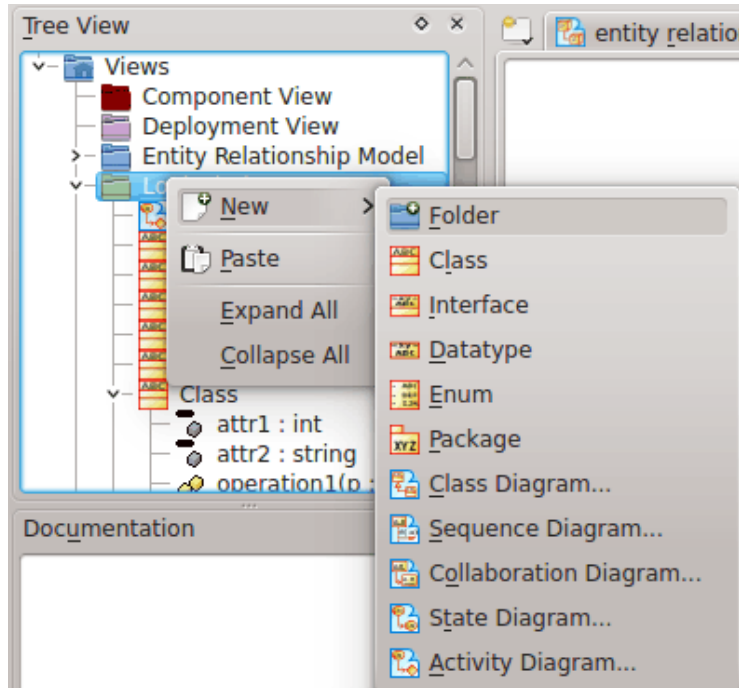
#### 5.1.3 Impressão

O Umbrello UML Modeller permite-lhe imprimir individualmente os diagramas. Carregue no botão **Imprimir** na barra da aplicação ou seleccionando a opção **Imprimir** do menu **Ficheiro**, para que lhe apareça uma janela normal de impressão do KDE, na qual você poderá imprimir os seus diagramas.

#### 5.1.4 Pastas Lógicas

Para organizar melhor o seu modelo, especialmente no caso de projectos grandes, você poderá criar pastas lógicas na Árvore. Basta seleccionar a opção **Novo** → **Pasta** do menu de contexto das pastas por omissão da árvore para criar essas pastas. As pastas podem ser aninhadas e você

poderá mover os objectos de uma para outra, se os arrastar do local de origem para os largar no destino.



*Organizar um Modelo com Pastas Lógicas no Umbrello UML Modeller*

## Capítulo 6

# Autores e História

Este projecto foi iniciado pelo Paul Hensgen com dos seus projectos universitários. O nome original da aplicação era UML Modeller (Modelador UML). O Paul fez todo o desenvolvimento até ao fim de 2001 quando o programa chegou à versão 1.0.

A versão 1.0 já oferecia bastantes funcionalidades, mas depois de o projecto ter sido revisto na Universidade do Paul, outros programadores puderam-se juntar e começaram a dar contribuições valiosas para o UML Modeller, como a mudança de um formato de ficheiro binário para um ficheiro XML, o suporte para mais tipos de diagramas UML, a Geração de Código e a Importação de Código, só para referir alguns.

O Paul teve de ser retirado da equipa de desenvolvimento no Verão de 2002 mas, como 'Software' Livre e Aberto que é, o programa continua a melhorar e a evoluir e é mantido por um grupo de programadores de diferentes partes do mundo. Da mesma forma, em Setembro de 2002, o projecto mudou o seu nome de UML Modeller, para Umbrello UML Modeller. Existem várias razões para a mudança de nomes, sendo a mais importante que apenas 'uml' — tal como era conhecido normalmente — era um nome demasiado genérico e causava alguns problemas com algumas distribuições. A outra razão importante era que o Umbrello era um nome muito mais giro.

O desenvolvimento do Umbrello UML Modeller, assim como as discussões sobre a estratégia para as versões futuras, é aberto e toma lugar na Internet. Se quiser contribuir para o projecto, por favor não hesite em contactar os programadores. Existem muitas formas pelas quais poderá ajudar o Umbrello UML Modeller:

- Comunicar erros ou sugerir melhoramentos
- Corrigir erros ou adicionar funcionalidades
- Escrever boa documentação ou traduzi-la para outras línguas
- E, claro... fazendo código connosco!

Como vê, existem muitas formas pelas quais você poderá contribuir. Todas elas são muito importantes e toda a gente é bem-vinda para participar.

Os programadores do Umbrello UML Modeller poderão ser contactados através de [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org).

## Capítulo 7

# Copyright

Copyright 2001, Paul Hensgen

'Copyright' 2002-2014 dos Autores do Umbrello UML Modeller.

A documentação está licenciada ao abrigo da [GNU Free Documentation License](#).

Este programa está licenciado ao abrigo da [GNU General Public License](#).