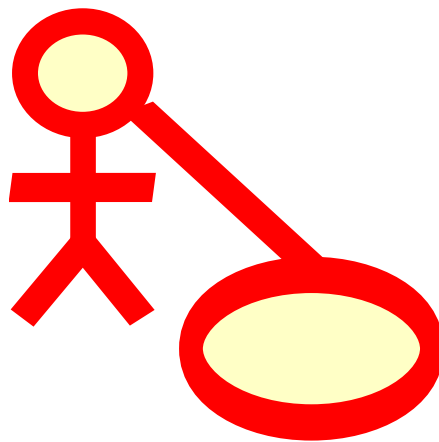


# Manual do Umbrello UML Modeller



## Manual do Umbrello UML Modeller

# Índice

<b>1</b>	<b>Introdução</b>	<b>8</b>
<b>2</b>	<b>Fundamentos do UML</b>	<b>9</b>
2.1	Sobre o UML . . . . .	9
2.2	Elementos UML . . . . .	10
2.2.1	Diagrama de Caso de Uso . . . . .	10
2.2.1.1	Caso de Uso . . . . .	10
2.2.1.2	Ator . . . . .	11
2.2.1.3	Descrição do Caso de Uso . . . . .	11
2.2.2	Diagrama de Classe . . . . .	11
2.2.2.1	Classe . . . . .	12
2.2.2.1.1	Atributos . . . . .	12
2.2.2.1.2	Operações . . . . .	13
2.2.2.1.3	Modelos . . . . .	13
2.2.2.2	Associações de Classe . . . . .	13
2.2.2.2.1	Generalização . . . . .	13
2.2.2.2.2	Associações . . . . .	13
2.2.2.2.3	Agregação . . . . .	14
2.2.2.2.4	Composição . . . . .	14
2.2.2.3	Outros Itens do Diagrama de Classe . . . . .	14
2.2.2.3.1	Interfaces . . . . .	14
2.2.2.3.2	Tipos de dados . . . . .	15
2.2.2.3.3	Enumerações . . . . .	15
2.2.2.3.4	Pacotes . . . . .	15
2.2.3	Diagramas de Sequência . . . . .	15
2.2.4	Diagramas de Colaboração . . . . .	16
2.2.5	Diagrama de Estado . . . . .	16
2.2.5.1	Estado . . . . .	17
2.2.6	Diagrama de Atividade . . . . .	18
2.2.6.1	Atividade . . . . .	18
2.2.7	Elementos Auxiliares . . . . .	18
2.2.8	Diagramas de Componente . . . . .	19

2.2.9	Diagramas de Distribuição . . . . .	19
2.2.10	Diagramas de Entidade-Associação . . . . .	19
2.2.10.1	Entidade . . . . .	20
2.2.10.1.1	Atributos da Entidade . . . . .	20
2.2.10.1.2	Restrições . . . . .	20
2.2.11	Conceitos Extendidos dos Diagramas de Entidade-Associação (ER) . . . . .	21
2.2.11.1	Especialização . . . . .	21
2.2.11.1.1	Especialização Disjunta . . . . .	21
2.2.11.1.2	Especialização Sobreposta . . . . .	22
2.2.11.1.3	Categoria . . . . .	22
<b>3</b>	<b>Trabalhando com o Umbrello UML Modeller</b>	<b>24</b>
3.1	Interface do usuário . . . . .	24
3.1.1	Visão de Árvore . . . . .	25
3.1.2	Janela de histórico de comandos e documentação . . . . .	25
3.1.3	Área de Trabalho . . . . .	25
3.2	Criando, Carregando e Salvando Modelos . . . . .	26
3.2.1	Novo Modelo . . . . .	26
3.2.2	Salvar Modelo . . . . .	26
3.2.3	Carregar Modelo . . . . .	26
3.3	Editando Modelos . . . . .	26
3.4	Adicionando e Removendo Diagramas . . . . .	27
3.4.1	Criando Diagramas . . . . .	27
3.4.2	Removendo Diagramas . . . . .	27
3.4.3	Renomeando Diagramas . . . . .	27
3.5	Editando Diagramas . . . . .	28
3.5.1	Inserir Elementos . . . . .	28
3.5.2	Excluindo Elementos . . . . .	28
3.5.3	Editando Elementos . . . . .	29
3.5.4	Editando Classes . . . . .	29
3.5.4.1	Configurações Gerais da Classe . . . . .	29
3.5.4.2	Configurações de Atributos da Classe . . . . .	29
3.5.4.3	Configurações de Operações da Classe . . . . .	29
3.5.4.4	Configurações de Modelo de Classe . . . . .	30
3.5.4.5	Página Associações da Classe . . . . .	30
3.5.4.6	Página de Exibição da Classe . . . . .	30
3.5.4.7	Página de estilo da classe . . . . .	30
3.5.5	Associações . . . . .	30
3.5.5.1	Pontos de Ancoragem . . . . .	31
3.5.6	Notas, Texto e Caixas . . . . .	31
3.5.6.1	Âncoras . . . . .	31

<b>4</b>	<b>Importação de Código e Geração de Código</b>	<b>32</b>
4.1	Geração de Código	32
4.1.1	Gerando Código	32
4.1.1.1	Opções Gerais	33
4.1.1.1.1	Detalhes dos comentários	33
4.1.1.1.2	Pastas	33
4.1.1.1.3	Política de Sobrescrever	34
4.1.1.1.4	Linguagem	34
4.1.1.2	Geração do Assistente de Geração	34
4.2	Importar Código	34
<b>5</b>	<b>Outros Recursos</b>	<b>36</b>
5.1	Outros Recursos do Umbrello UML Modeller	36
5.1.1	Copiar objetos como imagens PNG	36
5.1.2	Exportando para uma Imagem	36
5.1.3	Imprimindo	36
5.1.4	Pastas Lógicas	36
<b>6</b>	<b>Configurações</b>	<b>38</b>
6.1	Configurações gerais	38
6.1.1	Diversos	38
6.1.2	Salvar automaticamente	39
6.1.3	Inicialização	39
6.1.4	Notificações	39
6.2	Configurações de fonte	40
6.3	Configurações da interface do usuário	41
6.3.1	Geral	41
6.3.2	Associações	41
6.3.3	Cor	41
6.4	Configurações da classe	42
6.4.1	Mostrar	42
6.4.2	Escopo inicial	42
6.5	Configurações do importador de código	43
6.5.1	Incluir os caminhos de pesquisa	43
6.5.2	Importação de C++	43
6.6	Configurações de geração de código	44
6.6.1	Aba Geral das configurações de geração de código	44
6.6.1.1	Linguagem	44
6.6.1.2	Pastas	44
6.6.1.3	Política de sobrescrita	44
6.6.2	Aba de formatação das configurações de geração de código	45
6.6.2.1	Detalhes dos comentários	45

## Manual do Umbrello UML Modeller

6.6.2.2	Linhas . . . . .	45
6.6.3	Opções de linguagem . . . . .	46
6.6.3.1	Geração de código C++ . . . . .	46
6.6.3.1.1	Documentação . . . . .	46
6.6.3.1.2	Geral . . . . .	46
6.6.3.1.3	Método de geração de código . . . . .	47
6.7	Configurações do visualizador de código . . . . .	48
6.8	Configurações de layout automático . . . . .	49
<b>7</b>	<b>Autores e histórico</b>	<b>50</b>
<b>8</b>	<b>Direitos autorais</b>	<b>51</b>

## **Resumo**

O Umbrello UML Modeller auxiliar o processo de desenvolvimento de software através do uso do padrão industrial da Linguagem de Modelagem Unificada (UML) para permitir-lhe criar diagramas para o desenvolvimento e documentação de seus sistemas.

# Capítulo 1

## Introdução

O Umbrello UML Modeller é uma ferramenta de diagrama UML que pode auxiliá-lo no processo de desenvolvimento de software. Especialmente durante as fases de análise e desenho, o Umbrello UML Modeller lhe auxilia a obter um produto de alta qualidade. A UML pode também ser usada para documentar o desenho de seu software para auxiliá-lo e seus companheiros de desenvolvimento.

Ter um bom modelo de seu software é a melhor maneira de comunicar-se com outros desenvolvedores que estão trabalhando no projeto e com seus clientes. Um bom modelo é extremamente importante para projetos de tamanho médio e grande, mas é também muito útil para projetos pequenos. Caso esteja trabalhando num projeto pequeno, de um desenvolvedor, você se beneficiará com um bom modelo porque ele lhe dará uma visão geral que lhe auxiliará a codificar as coisas da maneira correta na primeira vez.

UML é uma linguagem de diagramação usada para descrever modelos. Você pode representar suas ideias em UML usando diferentes tipos de diagramas. O Umbrello UML Modeller 2.11 tem suporte aos seguintes tipos:

- Diagrama de Classe
- Diagrama de Sequência
- Diagrama de Colaboração
- Diagrama de Caso de Uso
- Diagrama de Estado
- Diagrama de Atividade
- Diagrama do componente
- Diagrama de Distribuição
- Diagrama de relacionamento de entidade

Mais informações sobre UML podem ser encontradas no site web da [OMG](http://www.omg.org), <http://www.omg.org>., criadora do padrão UML.

Nós esperamos que você goste do Umbrello UML Modeller e que ele lhe auxilie a criar software de alta qualidade. O Umbrello UML Modeller é um Software Livre e disponível gratuitamente, e a única coisa que lhe pedimos é reportar qualquer erro, problemas ou sugestões para os desenvolvedores do Umbrello UML Modeller em [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org) ou <https://bugs.kde.org>.

## Capítulo 2

# Fundamentos do UML

### 2.1 Sobre o UML

Este capítulo lhe fornecerá uma visão rápida dos fundamentos da UML. Tenha em mente que isto não é um tutorial detalhado sobre UML, mas apenas uma rápida introdução a UML que pode ser lida como um tutorial UML. Se você gostaria de aprender mais sobre a Linguagem de Modelagem Unificada, ou generalidades sobre análise e desenho de software, consulte um dos muitos livros disponíveis sobre o tópico. Existem também muitos tutoriais na Internet os quais você pode 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 objetos. O UML não é um método de desenvolvimento, o que significa que não lhe diz o que fazer primeiro ou o que fazer depois 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'.

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

A UML é composta por muitos elementos de modelo que representam as diferentes partes de um sistema de software. Os elementos UML são usados para criar diagramas, que representam um determinada parte, ou um ponto de vista do sistema. Os seguintes tipos de diagramas são suportados pelo Umbrello UML Modeller:

- *Diagrama de Caso de Uso* mostra atores (pessoas ou outros usuários do sistema), casos de uso (os cenários onde eles usam o sistema), e seus relacionamentos
- *Diagrama de Classe* mostra classes e os relacionamentos entre elas
- *Diagrama de Sequência* mostra objetos e uma sequência das chamadas do método feitas para outros objetos.
- *Diagrama de Colaboração* mostra objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens
- *Diagrama de Estado* mostra estados, mudanças de estado e eventos num objeto ou uma parte do sistema
- *Diagrama de Atividade* mostra atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema
- *Diagrama de Componente* mostra os componentes de programação de alto nível (como KParts ou Java Beans).

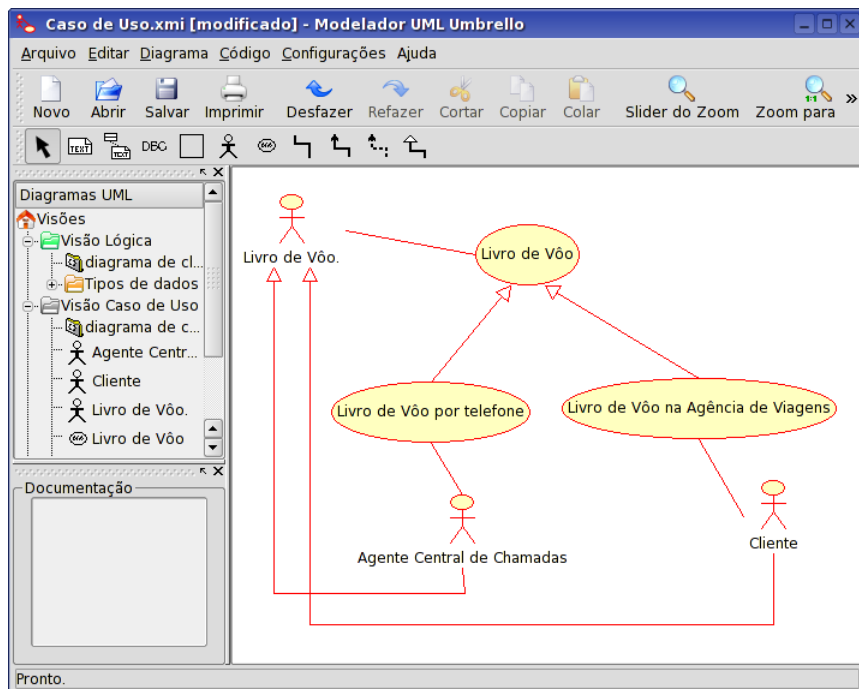
- *Diagrama de Distribuição* mostra as instâncias dos componentes e seus relacionamentos.
- Os *Diagramas de Entidade-Associação* mostram os dados e as relações e as restrições entre os dados.

## 2.2 Elementos UML

### 2.2.1 Diagrama de Caso de Uso

Diagramas de Caso de Uso descrevem relacionamentos e dependências entre um grupo de *Caso de Uso* e os Atores participantes no processo.

É importante observar que Diagramas de Caso de Uso não são adequados para representar o desenho, e não podem descrever os mecanismos internos de um sistema. Diagramas de Caso de Uso são feitos para facilitar a comunicação com os futuros usuários do sistema, e com o cliente, e são especialmente úteis para determinar os recursos necessários que o sistema deve ter. Diagramas de Caso de Uso dizem *o quê* o sistema deve fazer, mas não fazem — e não podem — especificar *como* isto será conseguido.



O Umbrello UML Modeller mostrando um Diagrama de Caso de Uso

#### 2.2.1.1 Caso de Uso

Um *Caso de Uso* descreve — do ponto de vista dos atores — um grupo de atividades num sistema que produz um resultado concreto e tangível.

Casos de Uso são descrições de interações típicas entre os usuários de um sistema e o sistema propriamente dito. Eles representam a interface externa do sistema e especificam um conjunto de exigências do que o sistema deve fazer (lembre-se: somente o *quê*, não *como*).

Quando trabalhar com Casos de Uso, é importante lembrar-se de algumas regras simples:

- Cada Caso de Uso está relacionado com no mínimo um ator

- Cada Caso de Uso possui um iniciador (isto é um ator)
- Cada Caso de Uso liga-se a um resultado relevante (um resultado com 'valor de negócio')

Casos de Uso também podem ter relacionamentos com outros Casos de Uso. Os três tipos mais comuns de relacionamento entre Casos de Uso são:

- «*inclui-se*» que especifica que um Caso de Uso toma lugar *dentro* de outro Caso de Uso
- «*estende*» que especifica que em determinadas situações, ou em algum ponto (chamado um ponto de extensão) um Caso de Uso será estendido por outro.
- *Generalização* especifica que um Caso de Uso herda as características do 'Super' Caso de Uso, e pode sobrepor algumas delas ou adicionar novas de maneira semelhante a herança entre classes.

### 2.2.1.2 Ator

Um ator é uma entidade externa (fora do sistema) que interage com o sistema participando (e frequentemente iniciando) um Caso de Uso. Atores podem ser pessoas reais (por exemplo usuários do sistema), outro sistema de computador ou eventos externos.

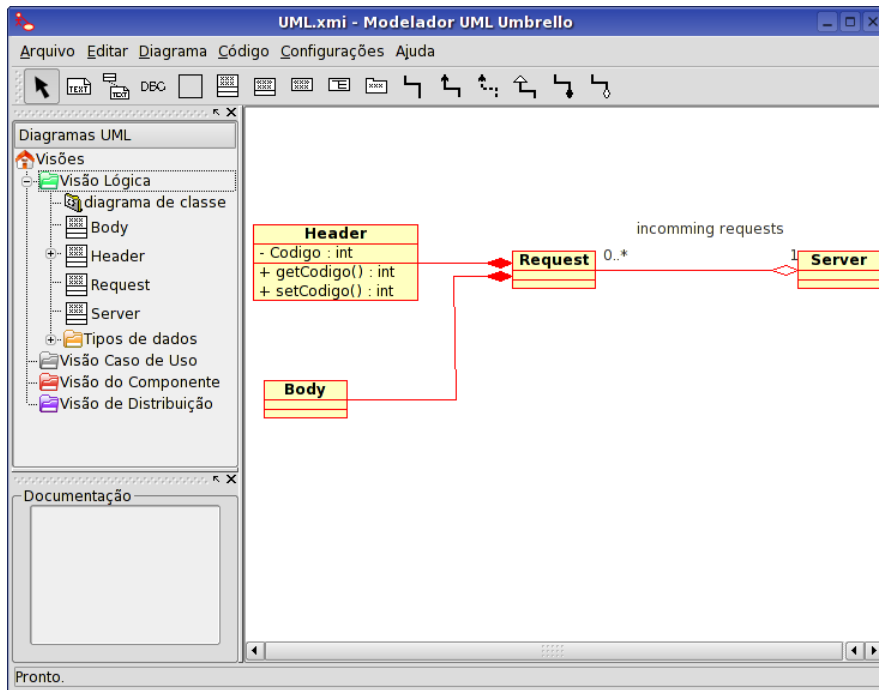
Atores não representam as pessoa *física* ou sistemas, mas sua *regra*. Isto significa que quando uma pessoa interage com o sistema de diferentes maneiras (assumindo diferentes regras) ela será representada por diversos atores. Por exemplo um pessoa que fornece suporte ao cliente por telefone e recebe ordens do cliente para o sistema pode ser representado por um ator da 'Equipe de Suporte' e um ator 'Representante de Vendas'

### 2.2.1.3 Descrição do Caso de Uso

Descrição do Caso de Uso são narrativas de texto do Caso de Uso. Elas usualmente tomam a forma de uma nota ou um documento que é de alguma maneira ligado ao Caso de Uso, e explica o processo ou atividades que tomarão lugar no Caso de Uso.

## 2.2.2 Diagrama de Classe

Diagramas de Classe mostram as diferentes classes que fazem um sistema e como elas se relacionam. Os Diagramas de Classe são chamados diagramas 'estáticos' porque mostram as classes, com seus métodos e atributos bem como os relacionamentos estáticos entre elas: quais classes 'conhecem' quais classes ou quais classes 'são parte' de outras classes, mas não mostram a troca de mensagens entre elas.

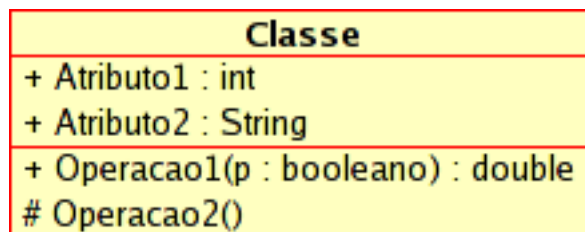


O Umbrello UML Modeller mostrando um Diagrama de Classe

### 2.2.2.1 Classe

Um Classe define os atributos e os métodos de um conjunto de objetos. Todos os objetos desta classe (instâncias desta classe) compartilham o mesmo comportamento, e possuem o mesmo conjunto de atributos (cada objeto possui seu próprio conjunto). O termo 'Tipo' é algumas vezes usado ao invés de Classe, mas é importante mencionar que estes dois termos não são a mesma coisa, e Tipo é um termo mais genérico.

Em UML Classes são representadas por retângulos, com o nome da classe, e podem também mostrar os atributos e operações da classe em dois outros 'compartimentos' dentro do retângulo.



Representação visual de uma Classe em UML

#### 2.2.2.1.1 Atributos

Na UML, atributos são mostrados com pelo menos seu nome, e podem também mostrar seu tipo, valor inicial e outras propriedades. Atributos podem também ser exibidos com sua visibilidade:

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

### 2.2.2.1.2 Operações

Operações (métodos) também são exibidos com pelo menos seu nome, e podem também mostrar seus parâmetros e valores de retorno. Operações podem, como os Atributos, mostrar sua visibilidade:

- + indica operações *públicas*
- # indica operações *protegidas*
- - indica operações *privadas*

### 2.2.2.1.3 Modelos

Classes podem ter modelos, um valor que é usado para uma classe ou tipo não especificado. O tipo de modelo é especificado quando uma classe é iniciada (isto é um objeto é criado). Modelos existem no C++ moderno e foram introduzidos no Java 1.5 onde eles são chamados de Genéricos.

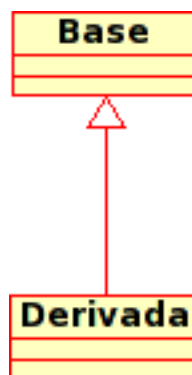
### 2.2.2.2 Associações de Classe

Classes podem relacionar-se (ser associada com) com outras de diferentes maneiras:

#### 2.2.2.2.1 Generalização

A herança é um dos conceitos fundamentais da programação orientada por objetos, 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.

EM UML, uma associação *Generalização* entre duas classes coloca-as numa hierarquia representando o conceito de herança de uma classe derivada de uma classe base. Em UML, Generalizações são representadas por uma linha conectando duas classes, com uma seta no lado da classe base.



Representação visual de uma generalização em UML

#### 2.2.2.2.2 Associações

Um associação representa um relacionamento entre classes, e fornece a semântica comum e a estrutura para muitos tipos de 'conexões' entre objetos.

Associações são o mecanismo que permite objetos comunicarem-se entre si. Elas descrevem a conexão entre diferentes classes (a conexão entre os objetos atuais é chamada conexão do objeto, ou *link*).

Associações podem ter um regra que especifica o propósito da associação e pode ser uni ou bi-direcional (indicando se os dois objetos participantes do relacionamento podem mandar mensagens para o outro, ou se apenas um deles sabe sobre o outro). Cada ponta da associação também possui um valor de multiplicidade, que dita como muitos objetos neste lado da associação pode relacionar-se com o outro lado.

Em UML, associações são representadas como linhas conectando as classes participantes do relacionamento, e podem também mostrar a regra e a multiplicidade de cada um dos participantes. A multiplicidade é exibida como um intervalo [min...máx] de valores não negativos, com uma estrela (\*) no lado máximo representando infinito.



Representação visual de uma Associação em UML

#### 2.2.2.3 Agregação

Agregações são um tipo especial de associação no qual as duas classes participantes não possuem em nível igual, mas fazem um relacionamento 'todo-parte'. Uma Agregação descreve como a classe que possui a regra do todo, é composta (tem) de outras classes, que possuem a regra das partes. Para Agregações, a classe que age como o todo sempre tem uma multiplicidade de um.

Em UML, Agregações são representadas por uma associação que mostra um romboide no lado do todo.



Representação visual de um relacionamento Agregação em UML

#### 2.2.2.4 Composição

Composições são associações que representam agregações  *muito fortes*. Isto significa que Composições formam relacionamentos todo-parte também, mas o relacionamento é tão forte que as partes não pode existir independentes. Elas existem somente dentro do todo, e se o todo é destruído as partes morrem também.

Em UML, Composições são representadas por um romboide sólido no lado do todo.



#### 2.2.2.3 Outros Itens do Diagrama de Classe

Diagramas de Classe podem conter diversos outros itens além das classes.

##### 2.2.2.3.1 Interfaces

Interfaces são classes abstratas que significam instâncias que não podem ser diretamente criadas delas. Elas podem conter operações mas não podem conter atributos. Classes podem derivar de interfaces (através da realização de uma associação) e instâncias podem então ser feitas destas classes.

### 2.2.2.3.2 Tipos de dados

Tipos de dados são primitivos uma vez que são tipicamente construídos numa linguagem de programação. Exemplos comuns são inteiros e lógicos. Eles não podem ser relacionados a classes, mas as classes podem se relacionar com eles.

### 2.2.2.3.3 Enumerações

Enumerações são uma lista simples de valores. Um exemplo típico é uma enumeração para dias da semana. As opções de uma enumeração são chamadas Literais de Enumeração. Como tipos de dados, elas não podem ter relacionamentos para classes, mas as classes podem relacionar-se com elas.

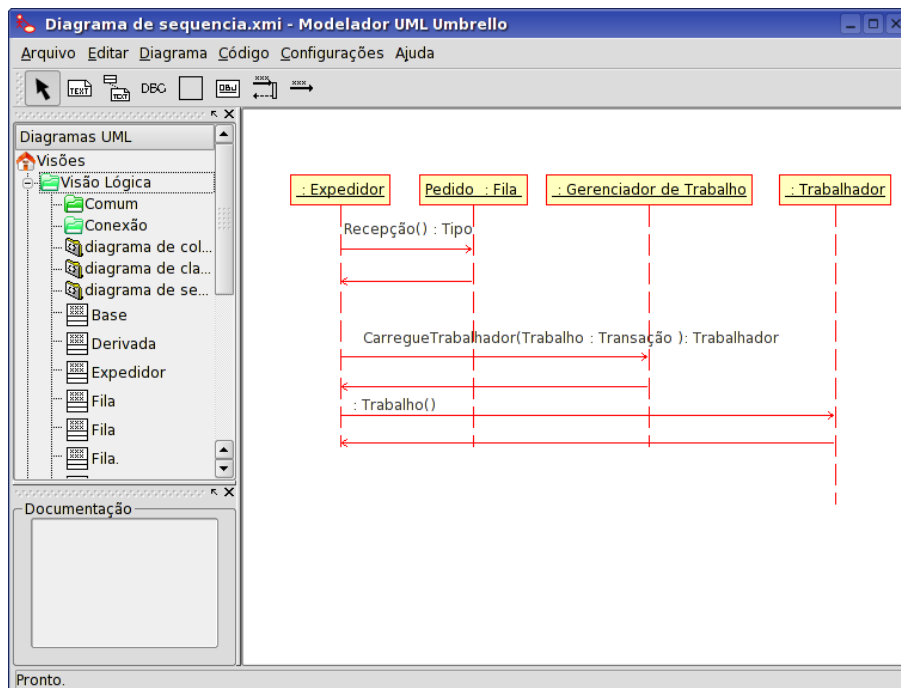
### 2.2.2.3.4 Pacotes

Pacotes representam um espaço de nomes numa linguagem de programação. Num diagrama eles são usados para representar partes de um sistema que contém mais de uma classe, talvez centenas de classes.

## 2.2.3 Diagramas de Sequência

Diagramas de Sequência mostram a troca de mensagens (isto é chamada de método) entre diversos Objetos, numa situação específica e delimitada no tempo. Objetos são instâncias de classes. Diagramas de Sequência colocam ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas.

Em Diagramas de Sequência objetos são representados através de linhas verticais tracejadas, com o nome do Objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um Objeto para outro na forma de setas com a operação e os nomes dos parâmetros.



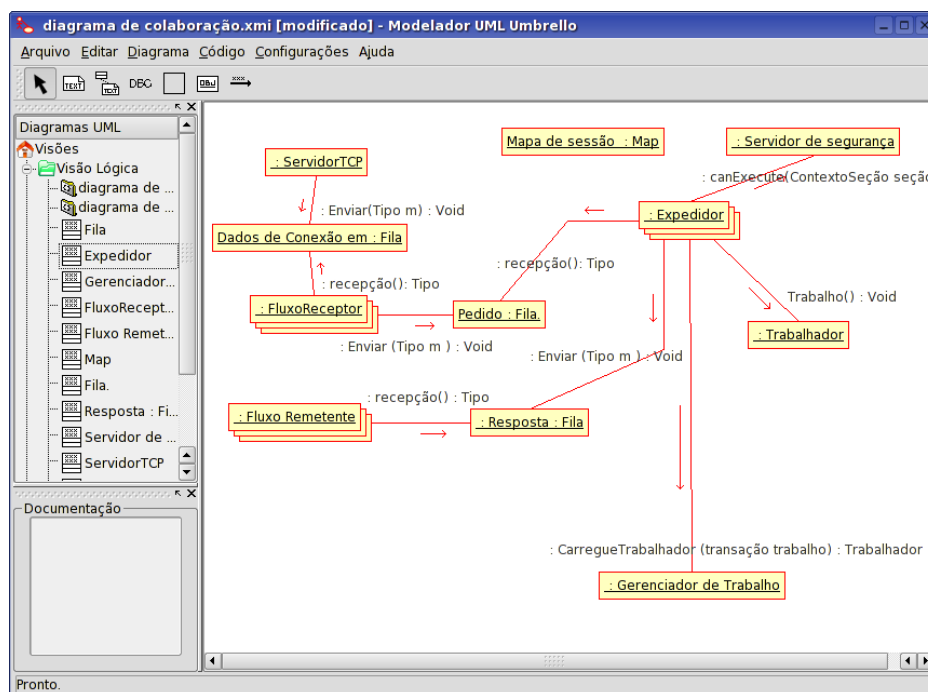
O Umbrello UML Modeller mostrando um Diagrama de Sequência

Mensagens pode ser síncronas, o tipo normal de mensagem de chamada onde o controle é passado para o objeto chamado até o método ter terminado sua execução, ou assíncronas onde o controle é passado diretamente para o objeto chamado. Mensagens síncronas possui uma caixa vertical no lado do objeto chamado para mostrar o controle do fluxo do programa.

## 2.2.4 Diagramas de Colaboração

Diagramas de Colaboração mostram as interações que ocorrem entre os objetos participantes numa situação específica. Isto é mais ou menos a mesma informação mostrada pelos Diagramas de Sequência, mas neste a ênfase é colocada em como as interações ocorrem no tempo, enquanto os Diagramas de Colaboração colocam os relacionamentos entre os objetos e sua topologia em destaque.

Em Diagramas de Colaboração as mensagens enviadas de um objeto para outro são representadas por setas, mostrando o nome da mensagem, parâmetros, e a sequência da mensagem. Diagramas de Colaboração são especialmente indicados para mostrar um fluxo ou situação específica do programa e são um dos melhores tipos de diagrama para rapidamente demonstrar ou explicar um processo na lógica do programa.



O Umbrello UML Modeller mostrando um Diagrama de Colaboração

## 2.2.5 Diagrama de Estado

Diagramas de Estado mostram os diferentes estados de um Objeto durante sua vida, e o estímulo que faz com que o Objeto mude seu estado.

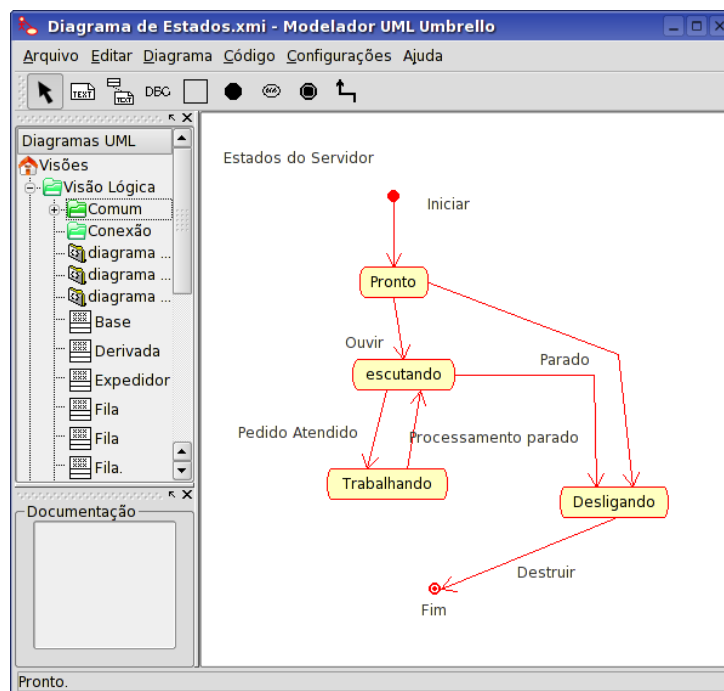
Diagramas de Estado veem Objetos como *máquinas de estado* ou automatismos finitos que podem ser um de um conjunto de estados finitos e que podem mudar seu estado através de um de um conjunto finito de estímulos. Por exemplo um tipo de Objeto *ServidorRede* pode estar em um dos seguintes estados durante sua vida:

- Pronto

- Ouvindo
- Trabalhando
- Parado

e os eventos que podem fazer com que o Objeto mude de estado são

- Objeto é criado
- Objeto recebe mensagem ouvir
- Um Cliente solicita uma conexão através da rede
- Um Cliente termina um pedido
- O pedido é executado e terminado
- Objeto recebe mensagem parar
- etc



O Umbrello UML Modeller mostrando um Diagrama de Estado

### 2.2.5.1 Estado

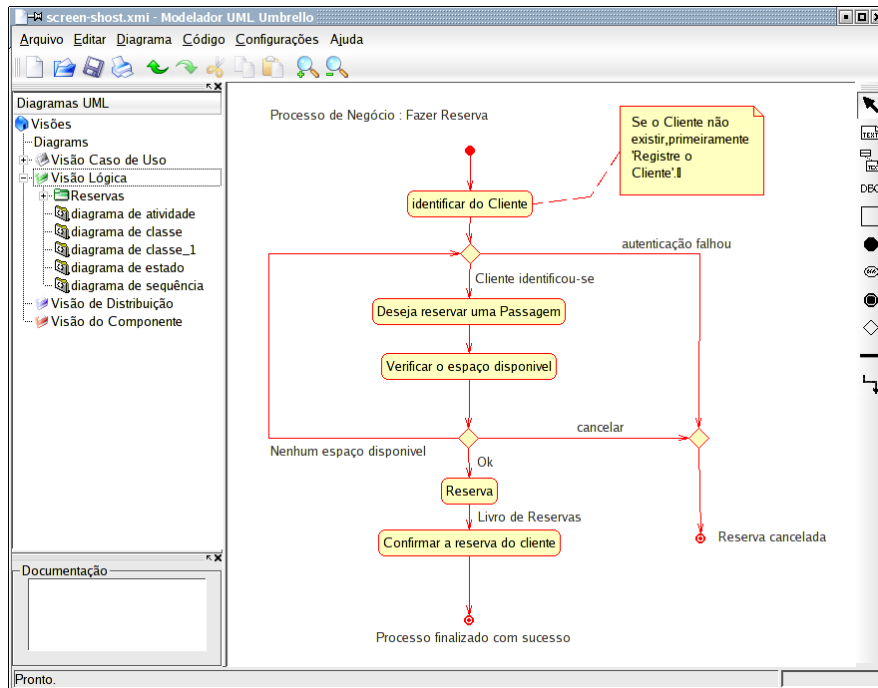
Estados são os blocos construídos dos Diagramas de Estado. Um Estado pertence a exatamente uma classe e representa um resumo dos valores dos atributos que uma classe pode tomar. Um Estado UML descreve o estado interno de um objeto para uma classe em particular

Observe que nem toda mudança em um dos atributos de um objeto pode ser representada por um Estado mas somente aquelas mudanças que podem afetar significativamente o trabalho do objeto

Existem dois tipos especiais de Estados: Inicial e Final. Eles são especiais porque nenhum evento pode fazer com que um Objeto retorne para seu estado Inicial, e da mesma maneira nenhum evento pode tirar um Objeto de seu estado Final uma vez que ele já o tenha alcançado.

## 2.2.6 Diagrama de Atividade

O Diagrama de Atividade descreve a sequência de atividades num sistema com a ajuda as Atividades. Diagramas de Atividade são uma forma especial de Diagramas de Estado, que somente (ou principalmente) contém Atividades.



O Umbrello UML Modeller mostrando um Diagrama de Atividade

Diagramas de Atividade são similares as Diagramas de Fluxo de procedimentos, com a diferença de que todas as Atividades são claramente anexas aos Objetos.

Diagramas de Atividade são sempre associados a um *Classe*, uma *Operação* ou um *Caso de Uso*.

Diagramas de Atividade suportam Atividades sequenciais bem como paralelas. A execução paralela é representada pelos ícones Forquilha/Esperar, e para as Atividades executadas em paralelo, não é importante a ordem na qual elas se executam (elas podem ser executadas ao mesmo tempo ou uma após a outra).

### 2.2.6.1 Atividade

Uma Atividade é um passo simples num processo. Uma Atividade é um estado no sistema com atividade interna e, pelo menos, uma transição de saída. Atividades podem também ter mais de uma transição de saída se elas possuem condições diferentes.

Atividades podem formar hierarquias, isto significa que uma Atividade pode ser composta por diversas Atividades em 'detalhe', na qual as transições de entrada e saída devem corresponder às transições de entrada e saída do diagrama de detalhe.

### 2.2.7 Elementos Auxiliares

Existem dois elementos em UML que não possuem nenhum valor real semântico para o modelo, mas auxiliam a elucidar partes do diagrama. Estes elementos são

- Linhas de texto

- Notas de Texto e âncoras
- Caixas

Linhas de texto são úteis para adicionar informações curtas de texto ao diagrama. São textos livres e não possuem nenhum significado para o Modelo propriamente dito.

Notas são úteis para adicionar informações mais detalhadas sobre um objeto ou situação específica. Elas possuem a grande vantagem de poderem ser ancoradas a Elementos UML para mostrar que a nota 'pertence' a um objeto específico ou situação.

Caixas são retângulos de forma livre que podem ser usados para agrupar itens tornando os diagramas mais legíveis. Eles não possuem nenhum significado lógico no modelo.

### 2.2.8 Diagramas de Componente

Diagramas de Componente mostram os componentes do software (sejam componentes de tecnologias como KParts, componentes CORBA ou Java Beans ou apenas seções do sistema que são claramente distintas) e os artefatos de que eles são feitos como arquivos de código-fonte, bibliotecas de programação ou tabelas de bancos de dados relacionais.

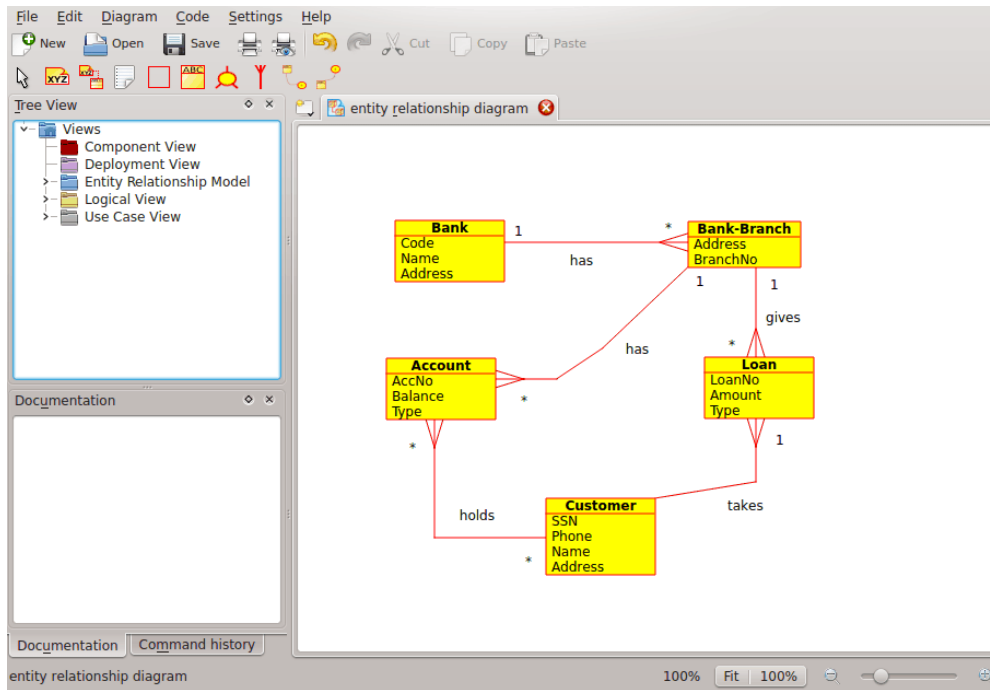
Componentes pode possui interfaces (isto é classes abstratas com operações) que permitem associações entre componentes.

### 2.2.9 Diagramas de Distribuição

Diagramas de distribuição mostram as instâncias dos componentes de tempo de execução e suas associações. Eles incluem Nós que são recursos físicos, tipicamente um computador simples. Eles também mostram interfaces e objetos (instâncias da classe).

### 2.2.10 Diagramas de Entidade-Associação

Os Diagramas de Entidade-Associação (Diagramas ER) mostram o desenho conceitual dos aplicativos de bancos de dados. Eles definem as várias entidades (conceitos) no sistema de informação e as relações e restrições entre eles. Uma extensão dos Diagramas Entidade-Associação, chamada 'Diagramas Extendidos Entidade-Associação' (EER) são usados para incorporar as técnicas de desenho Orientado por Objetos nos diagramas ER.



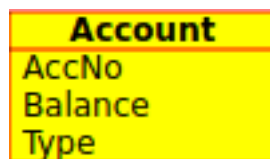
O Umbrello mostrando um Diagrama de Entidade-Associação

### 2.2.10.1 Entidade

Uma *Entidade* é qualquer conceito no mundo real com uma existência independente. Poderá ser um objeto com uma existência física (exemplo, Computador, Robô) ou poderá ser um objeto com uma existência conceitual ( p.ex.: Curso Universitário). Cada entidade possui 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 4<sup>a</sup> ed. Addison Wesley*

Num Diagrama ER, as Entidades são representadas por retângulos com o nome da classe, e poderão também mostrar os atributos de uma entidade noutro ‘compartimento’ dentro do retângulo.



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 atualizar um item numa tabela de um banco 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 Extendidos dos Diagramas de Entidade-Associação (ER)

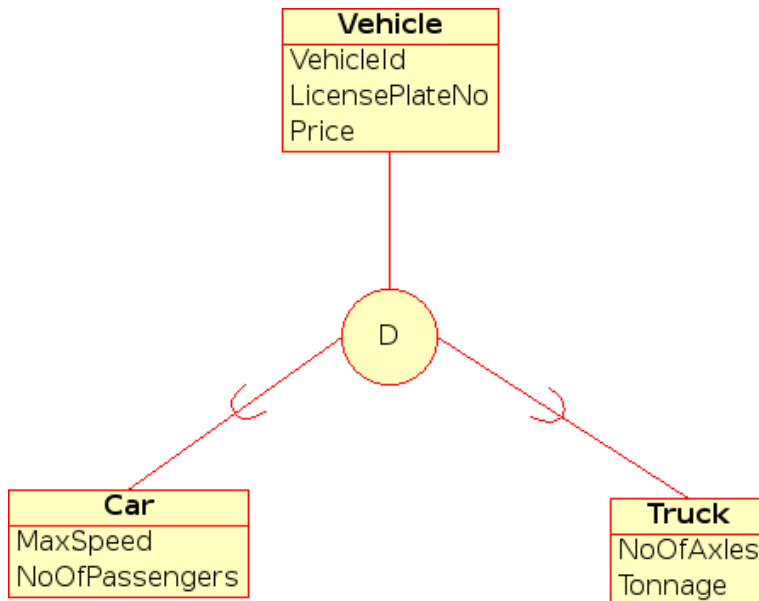
### 2.2.11.1 Especialização

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 pouca ou nenhuma modificação.

No Umbrello, uma pessoa poderá definir Especializações Disjuntas e Sobrepostas

#### 2.2.11.1.1 Especialização Disjunta

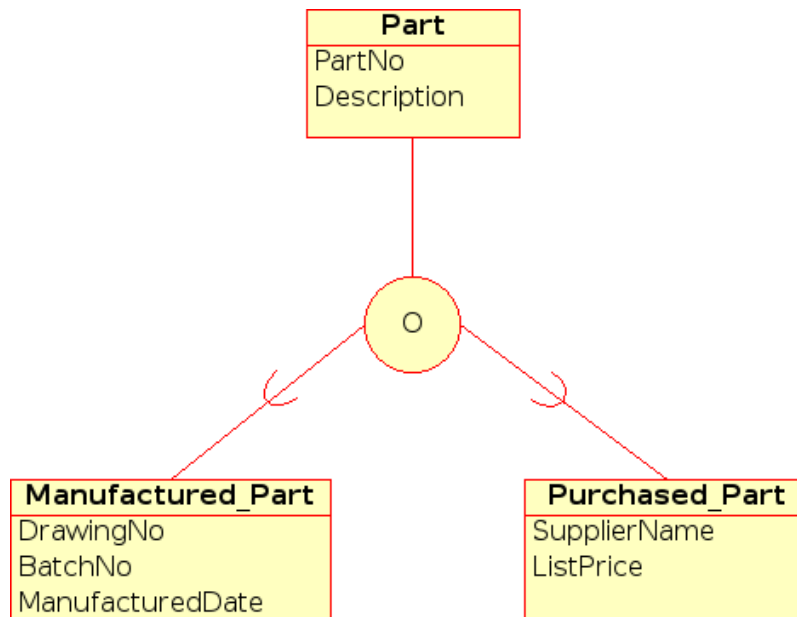
A Especialização Disjunta especifica que as subclasses de especialização devem ser disjuntas. Isto significa que uma entidade pode ser um membro de pelo menos uma das entidades derivadas da especialização



Representação visual de uma Especialização Disjunta num Diagrama EER

### 2.2.11.1.2 Especialização Sobreposta

Quando as entidades derivadas não são forçadas a serem disjuntas, este conjunto de entidades são ditos como sendo de uma especialização sobreposta. Isto significa que a mesma entidade do mundo real pode ser membro de mais de uma entidade derivada de especialização.

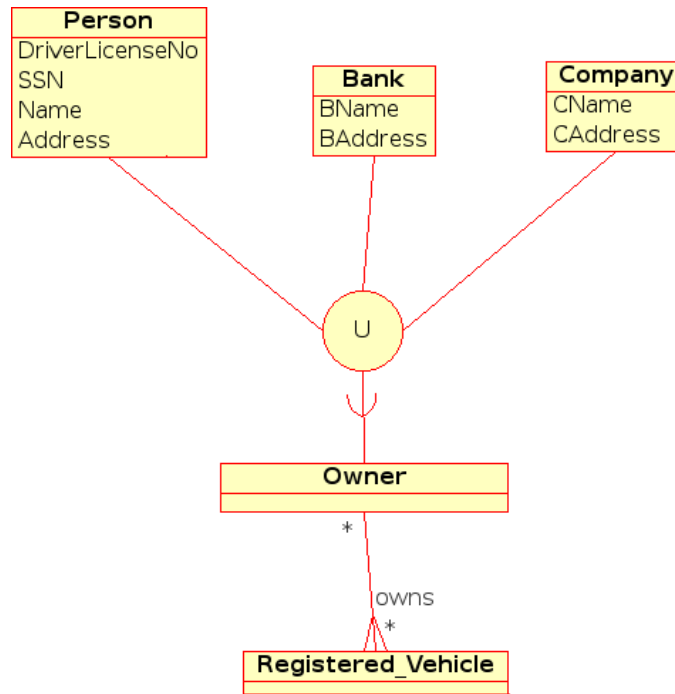


Representação visual de uma Especialização Sobreposta num Diagrama EER

### 2.2.11.1.3 Categoria

Uma Entidade derivada é chamada de *Categoria* quando ela representa uma coleção de objetos que é um subconjunto da União de tipos de entidade distintos. Uma Categoria é modelada

quando existe a necessidade de um único relacionamento de superclasse/subclasse com mais de uma superclasse, onde as superclasses representam diferentes tipos de entidade. ( Como a herança múltipla em Programação Orientada a Objeto ).



Representação visual de uma Categoria num Diagrama EER

## Capítulo 3

# Trabalhando com o Umbrello UML Modeller

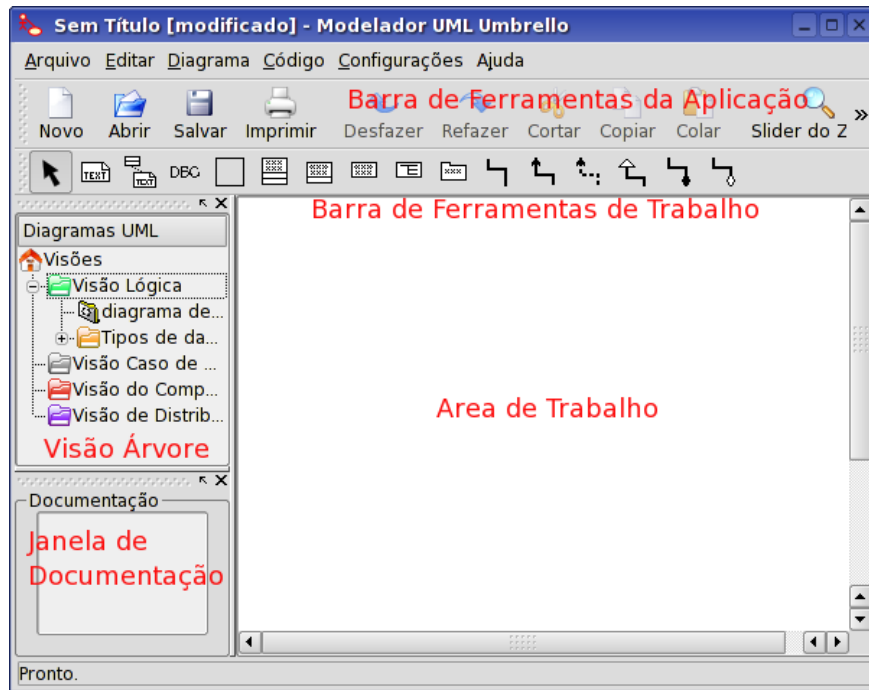
Este capítulo lhe introduzirá na interface com o usuário do Umbrello UML Modeller e lhe dirá tudo que você precisa saber para iniciar um modelo. Todas as ações no Umbrello UML Modeller estão acessíveis através do menu e das barras de ferramentas, mas o Umbrello UML Modeller também faz uso intensivo de menus de contexto do botão direito do mouse. Você pode clicar o botão direito do mouse na maioria dos elementos da área de trabalho do Umbrello UML Modeller ou visão de árvore para obter um menu com as funções mais úteis que podem ser aplicadas ao elemento em particular que você está trabalhando. Alguns usuários acharão isto um pouco confuso no início, porque eles estão mais acostumados a trabalhar com o menu ou barras de ferramentas, mas uma vez que você acostume-se a usar o clique direito, isto aumentará significativamente a velocidade do seu trabalho.

### 3.1 Interface do usuário

A janela principal do Umbrello UML Modeller é dividida em três áreas que lhe auxiliam a manter uma visão geral de todo sistema e acessar os diferentes diagramas rapidamente enquanto trabalha em seu modelo.

Estas áreas são chamadas:

- Visão de Árvore
- Área de Trabalho
- Janela de histórico de comandos e documentação



*Interface com o Usuário do Umbrello UML Modeller*

### 3.1.1 Visão de Árvore

A Visão de Árvore é localizada na parte superior esquerda da janela, e mostra todos os diagramas, classes, atores e casos de uso que compuserem o seu modelo. A Visão de Árvore lhe permite ter uma rápida visão geral dos elementos componentes de seu modelo. A Visão de Árvore também lhe fornece uma maneira rápida de alternar entre os diferentes diagramas em seu modelo e inserir elementos de seu modelo no diagrama atual.

Se você está trabalhando em um modelo com mais do que algumas poucas classes e diagramas, a Visão de Árvore pode auxiliá-lo a manter no topo as coisas organizando seus elementos do modelo em pastas. Você pode criar pastas selecionando a opção apropriada no menu de contexto (clique botão direito do mouse em uma das pastas na visão de árvore) e você pode organizar seus elementos movendo-os para a pasta apropriada (arrastar e soltar)

### 3.1.2 Janela de histórico de comandos e documentação

A janela de histórico de comandos e documentação é a pequena janela localizada na parte inferior esquerda do Umbrello UML Modeller, e ela lhe fornece uma rápida previsão da documentação para o item atualmente selecionado e o histórico de comandos da sua sessão de trabalho. A janela de documentação é um pouco pequena porque ela pretende permitir-lhe apenas uma rápida observação dos elementos da documentação e uma visão geral do histórico de comandos, ocupando o menor espaço possível na tela. Se você precisar ver a documentação em mais detalhes você pode 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 é onde as ações reais ocorrem. Você usa a Área de Trabalho para editar e ver os diagramas em seu modelo. A Área

de Trabalho mostra o diagrama atualmente ativo. Atualmente somente um diagrama pode ser mostrado na Área de Trabalho de cada vez.

## 3.2 Criando, Carregando e Salvando Modelos

A primeira coisa que você precisa para iniciar a fazer algo útil com o Umbrello UML Modeller é criar um modelo para trabalho. Quando você inicia o Umbrello UML Modeller ele sempre carrega o último modelo usado ou cria um novo modelo vazio (dependendo de suas configurações de preferências no diálogo de configuração). Isto lhe permitirá iniciar trabalhando rapidamente.

### 3.2.1 Novo Modelo

Se a qualquer momento você precisar criar um novo modelo você pode fazer isto selecionando a entrada **Novo** no menu **Arquivo**, ou clicando o ícone **Novo** na barra de ferramentas do aplicativo. Se você estiver trabalhando no momento em um modelo que foi modificado, o Umbrello UML Modeller lhe perguntará se deseja salvar as mudanças antes de carregar o novo modelo.

### 3.2.2 Salvar Modelo

Você pode salvar seu modelo a qualquer momento selecionando a opção **Salvar** do Menu **Arquivo** ou clicando no botão **Salvar** na barra de ferramentas do aplicativo. Se você precisar salvar seu modelo com um nome diferente você pode usar a opção **Salvar Como** no Menu **Arquivo**.

Para sua conveniência, o Umbrello UML Modeller também oferece a opção de salvar automaticamente seu trabalho a cada período de tempo determinado. Você pode configurar se deseja esta opção bem como o intervalo de tempo nas **Configurações** do Umbrello UML Modeller

### 3.2.3 Carregar Modelo

Para carregar um modelo já existente você pode selecionar a opção **Abrir** no Menu **Arquivo** ou clicar no ícone **Abrir** na barra de ferramentas do aplicativo. Os modelos mais recentemente usados também estão disponíveis no submenu **Abrir Recente** do Menu **Arquivo** para acelerar o acesso aos seus modelos mais frequentemente usados.

O Umbrello UML Modeller pode trabalhar com somente um modelo por vez, logo se você pedir que o programa carregue um modelo e o modelo atual foi modificado desde a última vez que você o salvou, o Umbrello UML Modeller lhe perguntará se suas mudanças devem ser salvas para evitar qualquer perda de seu trabalho. Você pode iniciar duas ou mais instâncias do Umbrello UML Modeller ao mesmo tempo, de modo que você possa copiar e colar entre as instâncias.

## 3.3 Editando Modelos

No Umbrello UML Modeller, existem basicamente duas maneiras de editar os elementos em seu modelo.

- Editar elementos do modelo diretamente através da Visão de Árvore
- Editar elementos do modelo através de um Diagrama

Usando o menu de contexto dos diferentes itens na Visão de Árvore você será capaz de adicionar, remover e modificar a maioria dos elementos em seu modelo. Clique direito nas pastas na Visão de Árvore lhe fornecerá as opções para criação dos diferentes tipos de diagramas bem como, dependendo se a pasta é uma *Visão de Caso de Uso* ou uma *Visão Lógica*, Atores, Casos de Uso, Classes, etc.

Uma vez que você tenha adicionado elementos ao seu modelo você pode também editá-los acessando seus diálogos de propriedades, que será encontrado selecionando a opção *Propriedades* no menu de contexto mostrado quando der um clique direito nos itens na Visão de Árvore.

Você pode também editar seu modelo criando e modificando elementos pelos diagramas. Mais detalhes sobre como fazer isto são fornecidos nas seguintes seções.

## 3.4 Adicionando e Removendo Diagramas

Seu modelo UML consiste de um conjunto de elementos UML e associações entre eles. No entanto você não pode ver o modelo diretamente, você usa os *Diagramas* para vê-lo.

### 3.4.1 Criando Diagramas

Para criar um novo diagrama em seu modelo simplesmente selecione o tipo de diagrama que você deseja a partir do submenu **Novo** no menu **Diagrama** e dê um nome a ele. O diagrama será criada e tornado ativo, e você imediatamente o verá na visão de árvore.

Lembre-se que o Umbrello UML Modeller faz um uso intensivo de menus de contexto: você pode também clicar o botão direito do mouse numa pasta na Visão de Árvore e selecionar o tipo apropriado de diagrama a partir do submenu **Novo** no menu de contexto. Observe que você pode criar Diagramas de Caso de Uso somente em pastas de Visão de Caso de Uso, e os outros tipos de diagrama podem somente ser criados em pastas de Visão Lógica.

### 3.4.2 Removendo Diagramas

Caso você precise remover um diagrama de seu modelo, você pode fazer isto tornando-o ativo e selecionando **Excluir** a partir do Menu **Diagrama**. Você pode também conseguir isto selecionando **Excluir** no menu de contexto do diagrama na Visão de Árvore

Uma vez que excluir um diagrama é algo sério que pode causar perda de trabalho se feito por acidente, o Umbrello UML Modeller lhe pedirá para confirmar a operação de exclusão antes de remover o Diagrama atual. Uma vez que o diagrama seja excluído e o arquivo salvo não existe nenhuma maneira de desfazer esta ação.

### 3.4.3 Renomeando Diagramas

Se você deseja mudar o nome de um diagrama existente você pode facilmente fazer isto selecionando a opção Renomear no menu botão direito do mouse na Visão de Árvore.

Outra maneira de renomear um diagrama é fazer isto através de seu diálogo de propriedades, que pode ser obtido selecionando Propriedades a partir do Menu de Contexto ou através de um clique duplo nele na Visão de Árvore.

## 3.5 Editando Diagramas

Enquanto trabalha num diagrama, o Umbrello UML Modeller tentará guiá-lo aplicando algumas regras simples como quais elementos são validados nos diferentes tipos de diagramas, bem como os relacionamentos que podem existir entre eles. Se você é experiente em UML você provavelmente não será noticiado, mas isto auxiliará novatos em UML a criar diagramas em conformidade com os padrões.

Uma vez que você tenha criado seus diagramas é hora de iniciar a editá-los. Aqui você deve notar (para iniciantes perspicazes) a diferença entre editar seu diagrama e editar o *modelo*. Como você já deve saber, Diagramas são *visões* do seu modelo. Por exemplo, se você criar uma classe editando um Diagrama de Classe, você estará realmente editando ambos, seu Diagrama e seu modelo. Se você mudar a cor ou outra opção de exibição de uma Classe em seu Diagrama de Classe, você estará somente editando o Diagrama, mas nada será mudado no seu modelo.

### 3.5.1 Inserir Elementos

Uma das primeiras coisas que você deverá fazer ao editar um novo diagrama é inserir elementos nele (Classes, Atores, Casos de Uso, etc.). Existem basicamente duas maneiras de fazer isto:

- Arrastando elementos existentes em seu modelo da Visão de Árvore
- Criando novos elementos em seu modelo e adicionando-os ao seu diagrama ao mesmo tempo, usando uma das ferramentas de edição da barra de ferramentas de trabalho

Para inserir elementos que já existam em seu modelo, apenas arraste-os da Visão de Árvore e solte-os onde você deseja posicioná-los em seu diagrama. Você sempre poderá mover elementos pelo Diagrama usando a Ferramenta Selecionar

A segunda maneira de adicionar elementos ao seu diagrama é usando as ferramentas de edição da barra de ferramentas de trabalho (observe que isto também adicionará elementos ao seu modelo).

A barra de ferramentas de trabalho era localizada por padrão na parte superior da janela. As ferramentas disponíveis nesta barra de ferramentas (os botões que você vê nela) mudam dependendo do tipo de diagrama em que você está atualmente trabalhando. O botão para a ferramenta atualmente selecionada é ativado na barra de ferramentas. Você pode alternar para a ferramenta selecionada pressionando a tecla **Esc**.

Quando você tiver selecionado uma ferramenta de edição da Barra de Ferramentas de Trabalho (por exemplo, a ferramenta para inserir classes) o ponteiro do mouse muda para uma cruz, e você pode inserir os elementos em seu modelo através de um clique simples em seu diagrama. Observe que elementos em UML devem ter um *Nome Único*. Logo se você tiver uma classe em um diagrama cujo nome é 'ClasseA' e então usar a ferramenta de inserção de classe para inserir um classe em outro diagrama você não pode nomear esta classe com 'ClasseA' também. Se estes dois serão supostamente dois diferentes elementos, você deve fornecer-lhes um nome único. Se você está tentando adicionar o *mesmo* elemento em seu diagrama, então a Inserir Classe não é a ferramenta adequada para isto. Ao invés disso, você deve arrastar e soltar a classe da Visão de Árvore.

### 3.5.2 Excluindo Elementos

Você pode excluir qualquer elemento selecionando a opção **Excluir** do seu menu de contexto.

Novamente, aqui existe uma *grande* diferença entre remover um objeto de um diagrama, e remover um objeto de seu modelo. Se você excluir um objeto de dentro de um diagrama, você está somente removendo o objeto daquele diagrama em particular: o elemento ainda faz parte do seu modelo e se existirem outros diagramas usando o mesmo elemento eles não sofrerão nenhuma

mudança. Se, por outro lado, você excluir o elemento da Visão de Árvore, você está atualmente excluindo o elemento de seu *modelo*. Uma vez que o elemento não mais exista em seu modelo, ele será automaticamente removido de todos os diagramas em que ele aparece.

### 3.5.3 Editando Elementos

Você pode editar a maioria dos elementos UML em seu modelo e diagramas abrindo seu diálogo de Propriedades e selecionando as opções apropriadas. Para editar as propriedades de um objeto, selecione **Propriedades** no seu menu de contexto (clique botão direito do mouse). Cada elemento possui um diálogo consistindo de diversas páginas de onde você pode configurar as opções correspondentes aquele elemento. Para alguns elementos, como atores, você pode somente configurar um par de opções, como o nome do objeto e documentação, enquanto outros elementos, como classes, você pode editar seus atributos e operações, selecionar o que você deseja que seja exibido no diagrama (toda a assinatura da operação ou apenas nomes da operação, etc) e também as cores que deseja usar para a linha e preenchimento da representação da classe no diagrama.

Para elementos UML, você também poderá abrir a janela de propriedades com um clique duplo sobre ela, caso esteja usando a ferramenta de seleção (seta).

Observe que você também pode selecionar a opção propriedades do menu de contexto dos elementos na Visão de Árvore. Isto lhe permite também editar as propriedades para os diagramas, como configurar se a grade deve ser mostrada ou não.

### 3.5.4 Editando Classes

Apesar da edição de propriedades para todos os objetos já ter sido abordada na seção anterior, as classes merecem uma seção especial porque elas são um pouco mais complexas e possuem mais opções que a maioria dos outros elementos UML.

No diálogo de propriedades para uma classe você pode configurar tudo, da cor que ela usa até as operações e atributos que ela possui.

#### 3.5.4.1 Configurações Gerais da Classe

A página de Configurações Gerais do diálogo de propriedades é autoexplicativa. Aqui você pode mudar o nome da classe, visibilidade, documentação, etc. Esta página está sempre disponível.

#### 3.5.4.2 Configurações de Atributos da Classe

Na página de Configurações de Atributos você pode adicionar, editar ou excluir atributos (variáveis) para esta classe. Você pode também mover atributos para cima e para baixo da lista pressionando o botão de seta ao lado. Esta página está sempre disponível.

#### 3.5.4.3 Configurações de Operações da Classe

Semelhante à Página de Configurações dos Atributos, na Página de Configurações das Operações você pode adicionar, editar ou remover operações para sua classe. Quando adiciona ou edita uma operação, você insere os dados básicos no diálogo *Propriedades da Operação*. Para adicionar parâmetros a sua operação você precisa clicar no botão **Novo Parâmetro**, que lhe mostrará o diálogo de *Propriedades do Parâmetro*. Esta página está sempre disponível

#### 3.5.4.4 Configurações de Modelo de Classe

Esta página permite-lhe adicionar modelos de classe que serão classes não especificadas ou tipos de dados. Em Java 1.5 eles são chamados Genéricos.

#### 3.5.4.5 Página Associações da Classe

A página **Associações da Classe** mostra todas as associações desta classe no diagrama atual. Um clique duplo numa associação mostra suas propriedades, e dependendo do tipo de associação você pode modificar alguns parâmetros como um configuração de multiplicidade e nome da Regra. Se a associação não permitir que opções sejam modificadas, o diálogo de Propriedades da Associação é somente leitura e você pode somente modificar a documentação associada com esta associação.

Esta página está disponível somente se você abrir as Propriedades da Classe de dentro de um diagrama. Se você selecionar as propriedades da classe no menu de contexto na Visão de Árvore esta página não estará disponível.

#### 3.5.4.6 Página de Exibição da Classe

Na página de **Opções de Exibição**, você pode configurar como a classe será mostrada no diagrama. Uma classe pode ser mostrada como somente um retângulo com o nome da classe nele (útil se você possui muitas classes em seu diagrama, ou não está interessado no momento nos detalhes de cada classe) ou completa mostrando pacotes, estereótipos, atributos e operações com assinatura completa e visibilidade

Dependendo da quantidade de informação que você deseja ver você pode selecionar as opções correspondentes nesta página. As mudanças que você realizar aqui serão somente as *opções de exibição* para o diagrama. Isto significa que 'ocultar' uma operação de classe somente faz com que ele não seja exibida no diagrama, mas a operação ainda faz parte de seu modelo. Esta opção está somente disponível se você selecionou as propriedades da classe de dentro de um Diagrama. Se você abrir as propriedades de classe a partir da Visão de Árvore esta página estará faltando uma vez que as Opções de Exibição não fazem sentido neste caso

#### 3.5.4.7 Página de estilo da classe

Na página **Estilo do widget** você pode configurar as cores que deseja para a linha e o preenchimento do widget. Esta opção obviamente faz sentido somente para classes exibidas em diagramas, e não aparecerá se você abrir o diálogo de propriedades da classe a partir da Visão de Árvore.

### 3.5.5 Associações

Associações relacionam dois objetos UML um com o outro. Normalmente associações são definidas entre duas classes, mas alguns tipos de associações podem também existir entre casos de uso e atores.

Para criar uma associação seleciona a ferramenta apropriada na Barra de Ferramentas de Trabalho (Associação genérica, Generalização, Agregação, etc.) e dê um clique simples no primeiro elemento participante na associação e então um clique simples no segundo elemento participante. Observe que são dois cliques, um em cada um dos objetos participantes da associação, e *não* arrastar um objeto para outro.

Se você tentar usar uma associação de uma maneira contrária as especificações UML o Umbrello UML Modeller se recusará a criar a associação e você receberá uma mensagem de erro. Isto pode

acontecer se, por exemplo, uma Generalização existir da classe A para classe B e então você tentar criar outra Generalização da classe B para a classe A

Um clique direito numa associação mostrará um menu de contexto com as ações que você pode aplicar nela. Se você precisar excluir uma associação, simplesmente selecione a opção **Excluir** deste menu de contexto. Você pode também selecionar a opção **Propriedades** e, dependendo do tipo de associação, editar os atributos como regras e multiplicidade.

#### 3.5.5.1 Pontos de Ancoragem

Associações são desenhadas, por padrão, como uma linha reta conectando dois objetos no diagrama.

Você pode adicionar pontos de ancoragem dando um clique duplo ao longo da linha de associação. Isto irá inserir um ponto de ancoragem (exibido como um ponto azul quando a linha de associação é selecionada) o qual você poderá mover para obter a forma da associação

Se você precisar remover um ponto de ancoragem, dê um clique duplo nele novamente para removê-lo

Observe que a única maneira de editar as propriedades de uma associação é através do menu de contexto. Se você tentar dar um clique duplo nela como é feito nos outros objetos UML, isto somente irá inserir um ponto de ancoragem.

#### 3.5.6 Notas, Texto e Caixas

Notas, Linhas de Texto e Caixas são elementos que podem ser representados em qualquer tipo de diagrama e não possuem nenhum valor semântico, mas são muito úteis para adicionar comentários extras ou explicações que podem tornar seu diagrama mais fácil de entender.

Para adicionar uma Nota ou Linha de Texto, selecione a ferramenta correspondente na barra de ferramentas de trabalho e dê um clique simples no diagrama onde você deseja colocar seu comentário. Você pode editar o texto abrindo o elemento através do menu de contexto ou, no caso de notas, através de um clique duplo nelas.

##### 3.5.6.1 Âncoras

Âncoras são usadas para manter uma nota de texto e outro Elemento UML juntos. Por exemplo, você normalmente usa uma nota de texto para explicar ou fazer algum comentário sobre uma classe ou uma associação em particular, neste caso você pode usar a âncora para tornar claro que a nota 'pertence' aquele elemento em particular.

Para adicionar uma âncora entre uma nota e outro elemento UML, use a ferramenta âncora da barra de ferramentas de trabalho. Você primeiro precisa clicar na nota e então clicar no elemento UML ao qual deseja que a nota se ligue.

## Capítulo 4

# Importação de Código e Geração de Código

O Umbrello UML Modeller é uma ferramenta de modelagem UML, e como tal seu principal propósito é auxiliá-lo na *análise e desenho* de seus sistemas. No entanto, para fazer a transição entre seu desenho e sua *implementação*, o Umbrello UML Modeller permite gerar código-fonte numa linguagem de programação que você conheça. Se você quiser iniciar o uso do UML num projeto em C++ já iniciado, o Umbrello UML Modeller pode auxiliá-lo a criar um modelo de seu sistema a partir do código-fonte analisando seu código e importando as classes encontradas nele.

### 4.1 Geração de Código

O Umbrello UML Modeller pode gerar código-fonte a partir de várias linguagens de programação baseadas no seu Modelo UML para auxiliá-lo no início com a implementação do seu projeto. O código gerado consiste de declarações de classe, com seus métodos e atributos de modo que você possa ‘preencher as lacunas’ fornecendo a funcionalidade das suas operações de classes.

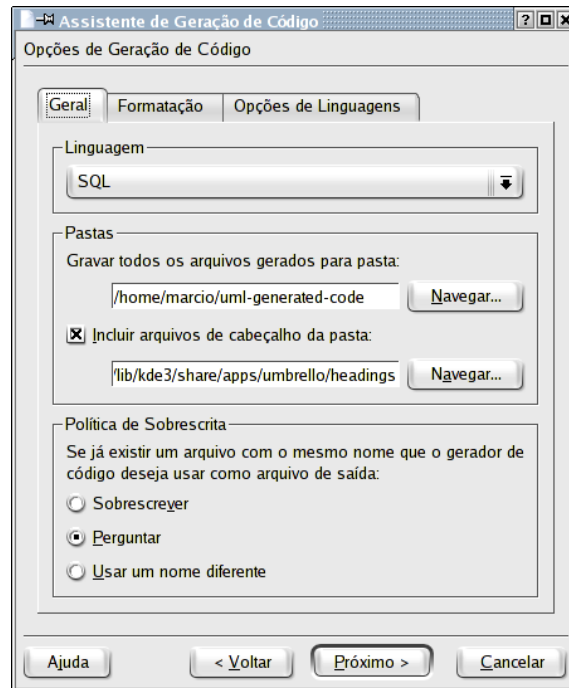
O Umbrello UML Modeller 2 vem com suporte para geração de código para ActionScript, Ada, C++, C#, D, IDL, Java™, JavaScript, MySQL, Pascal, Perl, PHP, PHP5, PostgreSQL, Python, Ruby, Tcl, Vala e XMLSchema.

#### 4.1.1 Gerando Código

Para gerar código como o Umbrello UML Modeller, você primeiro precisa criar ou carregar um Modelo contendo pelo menos uma classe. Quando você estiver pronto para iniciar a escrita de algum código, selecione a entrada do **Assistente de Geração de Código** no menu **Código** para iniciar um Assistente que o guiará através do processo de geração de código.

O primeiro passo é selecionar as classes para quais você deseja gerar código fonte. Por padrão todas as classes de seu modelo serão selecionadas, e você pode remover as que não deseja que seja gerado código movendo-as para o lado esquerdo da lista.

O próximo passo do assistente lhe permite modificar os parâmetros que o Gerador de Código usa ao escrever seu código. As seguintes opções estão disponíveis:



*Opções para a Geração de Código no Umbrello UML Modeller*

#### 4.1.1.1 Opções Gerais

##### 4.1.1.1.1 Detalhes dos comentários

A opção **Escrever comentários da documentação se vazios** instrui o Gerador de Código a escrever comentários no estilo `/** blah */` sempre que o bloco de comentários estiver vazio. Se você adicionar documentação a sua classe, métodos ou atributos em seu Modelo, o Gerador de Código escreverá estes comentários como documentação Doxygen a menos que você configure algo aqui, mas se você selecionar esta opção o Umbrello UML Modeller escreverá blocos de comentários para todas as classes, métodos e atributos se não houver nenhuma documentação no Modelo, caso contrário você deverá documentar suas classes posteriormente diretamente no código-fonte.

**Escrever comentários para seções se seção estiver vazia** faz com que o Umbrello UML Modeller escreva comentários no código-fonte para delimitar as diferentes seções de uma classe. Por exemplo 'métodos públicos' ou 'Atributos' antes das seções correspondentes. Se você selecionar esta opção o Umbrello UML Modeller escreverá comentários para todas as seções da classe se a seção estiver vazia. Por exemplo, ele escreverá um comentário 'métodos protegidos' se existem métodos não protegidos em sua classe.

##### 4.1.1.1.2 Pastas

**Gravar todos os arquivos gerados para pasta.** Aqui você pode selecionar a pasta onde deseja que o Umbrello UML Modeller coloque os fontes gerados.

A opção **Incluir arquivos de cabeçalho da pasta** permite-lhe inserir um cabeçalho no início de cada arquivo gerado. Arquivos de cabeçalho podem conter informações de direito de cópia ou licença e conter variáveis que serão avaliados no momento da geração. Você pode dar uma olhada nos arquivos modelos de cabeçalho fornecidos com o Umbrello UML Modeller para ver como usar estas variáveis para substituir seu nome ou a data atual no momento da geração.

#### 4.1.1.1.3 Política de Sobrescrever

Esta opção diz ao Umbrello UML Modeller o que fazer se o arquivo que está sendo criado já existe na pasta destino. O Umbrello UML Modeller *não pode modificar arquivos fonte existentes*, logo você deve escolher entre sobrescrever o arquivo existente, pular a geração do arquivo em particular ou deixar o Umbrello UML Modeller escolher um nome de arquivo diferente. Se você escolher a opção de usar um nome diferente, o Umbrello UML Modeller adicionará um sufixo ao nome do arquivo.

#### 4.1.1.1.4 Linguagem

O Umbrello UML Modeller irá gerar por padrão código na linguagem que você tiver selecionado como Linguagem Ativa, mas com o Assistente de Geração de Código você tem a opção de mudar isto para outra linguagem.

#### 4.1.1.2 Geração do Assistente de Geração

O terceiro e último passo do assistente mostra o estado do processo de Geração de Código. Você precisa somente clicar no botão Gerar para ter suas classes escritas para você.

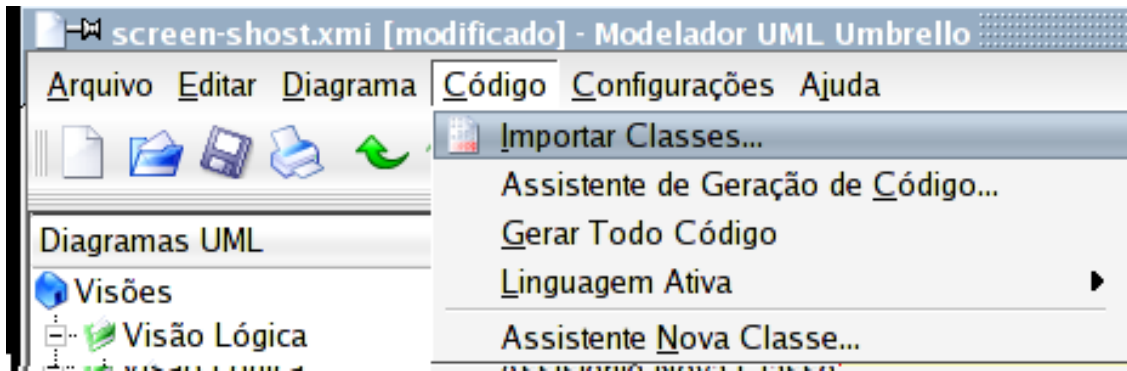
Observe que as Opções que você selecionou durante o assistente de Geração de Código são válidas somente para a geração atual. Na próxima vez que executar o Assistente você precisará re-selecionar todas as opções (sua pasta de cabeçalhos, política de sobrescrever, e assim por diante). Você pode configurar os padrões usados pelo Umbrello UML Modeller na seção **Geração de Código** das configurações do Umbrello UML Modeller disponível em **Configurações** → **Configurar o Umbrello UML Modeller...**

Se você tiver configurado suas opções de Geração de Código para as configurações corretas e desejar gerar algum código da mesma maneira sem passar pelo assistente, você pode selecionar a entrada **Gerar Todo Código** no menu **Código**. Isto irá gerar código para todas as classes em seu Modelo usando suas configurações atuais (incluindo Pasta de Saída e Política de Sobrescrever, logo use com cuidado).

## 4.2 Importar Código

O Umbrello UML Modeller pode importar código-fonte de seus projetos existentes para auxiliá-lo a construir o Modelo de seus sistemas. O Umbrello UML Modeller 2 tem suporte para código-fonte em ActionScript, Ada, C++, C#, D, IDL, Java™, Javascript, MySQL, Pascal, PHP e Vala.

Para importar classes para seu Modelo, selecione a entrada **Assistente de importação de código...** no menu **Código**. No diálogo de arquivo selecione os arquivos contendo as declarações de classe e clique em **Próximo >**, **Iniciar importação** e **Concluir**. As classes serão importadas e você irá encontrá-las como parte de seu Modelo na Visão Árvore. Observe que o Umbrello UML Modeller não criará nenhum tipo de Diagrama para mostrar suas classes, elas serão apenas importadas para seu Modelo e então você poderá usá-las mais tarde em qualquer diagrama que desejar.



*Menu para importação de código-fonte no Umbrello UML Modeller*

## Capítulo 5

# Outros Recursos

### 5.1 Outros Recursos do Umbrello UML Modeller

Este capítulo explanará sumariamente alguns outros recursos que o Umbrello UML Modeller lhe oferece.

#### 5.1.1 Copiar objetos como imagens PNG

Além de lhe oferecer as funcionalidades normais de copiar, cortar e colar que você espera para copiar objetos entre diagramas diferentes, o Umbrello UML Modeller pode copiar os objetos como figuras PNG de modo que você possa inseri-las em outro tipo de documento. Você não precisa fazer nada em especial para usar este recurso, apenas selecione um objeto de um diagrama (Classe, Ator, etc.) e copie-o (**Ctrl-C**, ou usando o menu), e então abra um documento do Calligra Words (ou qualquer programa para o qual você possa colar imagens) e selecione **Colar**. Este é um grande recurso para exportar partes de seu diagrama como simples figuras.

#### 5.1.2 Exportando para uma Imagem

Você pode também exportar um diagrama completo como uma imagem. A única coisa que você precisa fazer é selecionar o diagrama que você deseja exportar, e então a opção **Exportar como Figura...** no menu **Diagrama**.

Você pode exportar vários diagramas de uma só vez usando a opção **Exportar diagramas como imagens...** do menu **Arquivo**. Com isso, você também pode definir a resolução da imagem, para que as imagens não fiquem tão desfocadas.

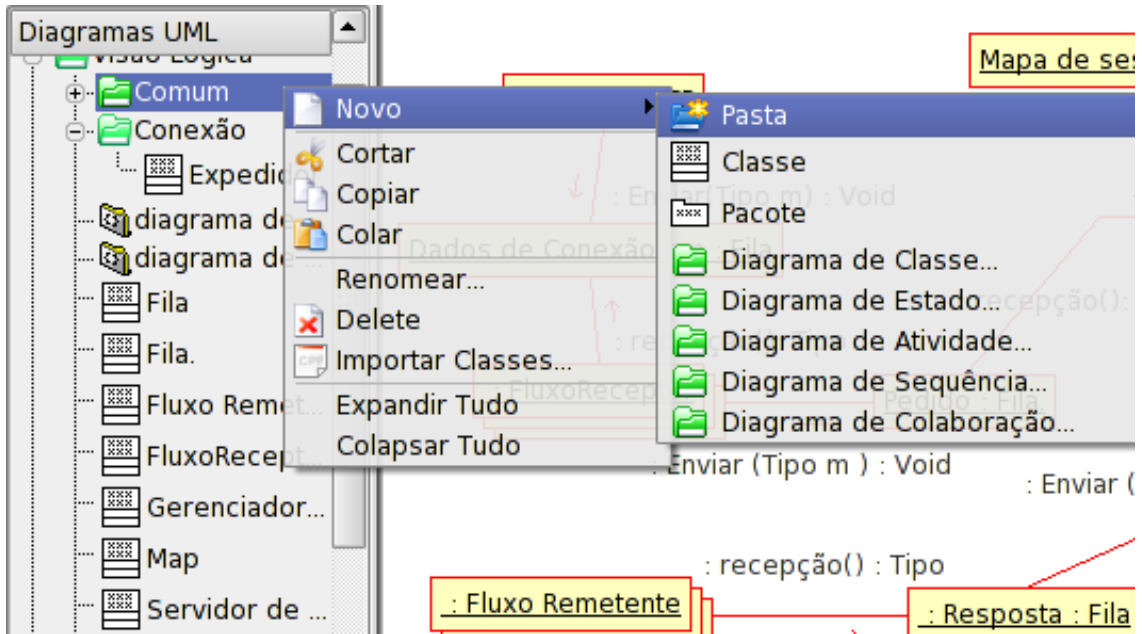
#### 5.1.3 Imprimindo

O Umbrello UML Modeller permite imprimir diagramas individuais. Pressionar o botão **Imprimir** na barra de ferramentas do aplicativo ou selecionar a opção **Imprimir** no menu **Arquivo** lhe trará o diálogo padrão de impressão do KDE a partir do qual você pode imprimir seus diagramas.

#### 5.1.4 Pastas Lógicas

Para melhor organizar o seu modelo, especialmente para grandes projetos, você pode criar pastas lógicas na Visão de Árvore. Apenas selecione a opção **Nova** → **Pasta** no menu de contexto das

pastas padrão na Visão de Árvore para criá-las. Pastas podem ser aninhadas, e você pode mover objetos arrastando-os de uma pasta e soltando-os em outra.

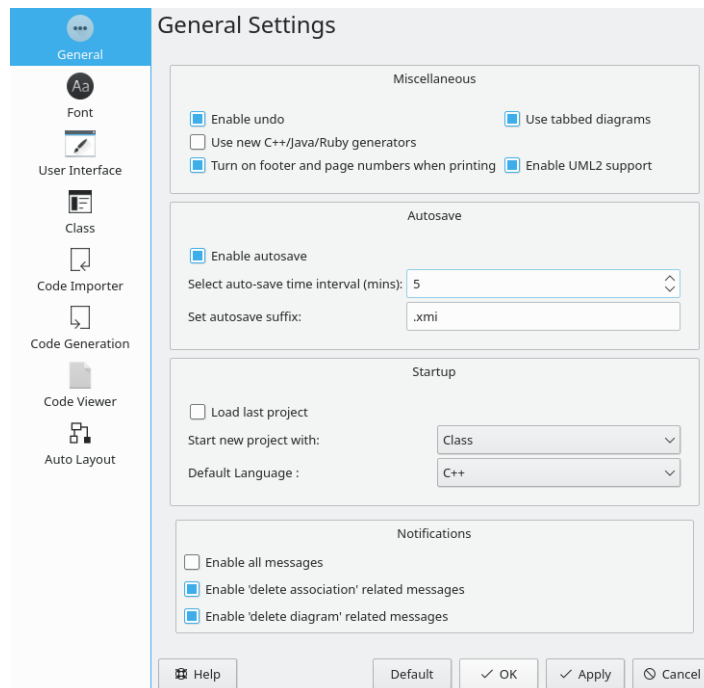


Organizando um Modelo com Pastas Lógicas no Umbrello UML Modeller

# Capítulo 6

## Configurações

### 6.1 Configurações gerais



*Opções para as configurações gerais no Umbrello UML Modeller*

#### 6.1.1 Diversos

- A opção **Habilitar desfazer** permite desfazer uma ação anterior.
- **Usar novos geradores de C++/Java/Ruby** permite ao usuário selecionar os geradores de código antigos ou novos
- **Ativar rodapé e números de página na impressão** quando selecionado, imprime as informações do diagrama que está sendo impresso e o número da página.
- **Usar diagramas em abas** oferece a opção de ter várias janelas de diagramas com abas abertas ao mesmo tempo.

### 6.1.2 Salvar automaticamente

- **Habilitar salvamento automático** oferece a opção de salvar o arquivo automaticamente.
- **Selecionar intervalo de tempo para salvamento automático (minutos):** permite definir o tempo antes do arquivo ser salvo automaticamente.
- **Definir sufixo de salvamento automático:** o padrão é .xmi, mas permite que uma extensão de arquivo diferente seja definida.

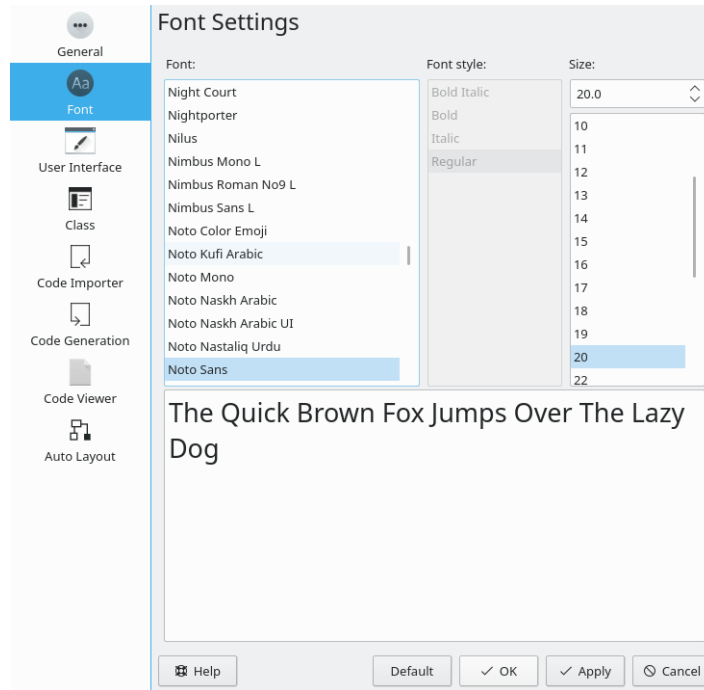
### 6.1.3 Inicialização

- **Carregar último projeto** se definido, sempre carrega o último projeto de trabalho ao iniciar o programa.
- **Iniciar novo projeto com:** oferece uma escolha de qual tipo de diagrama UML para começar em um novo projeto.
- **Linguagem padrão:** é uma configuração para a linguagem de programação padrão usada.

### 6.1.4 Notificações

- **Ativar todas as mensagens** é uma opção para visualizar todas as notificações ou um conjunto reduzido de notificações.
- **Ativar mensagens relacionadas a 'excluir associação'** garante que você receberá todas as mensagens desse tipo se estiver marcada.
- **Ativar mensagens relacionadas a 'excluir diagrama'** habilitará todas as mensagens desse tipo se marcadas.

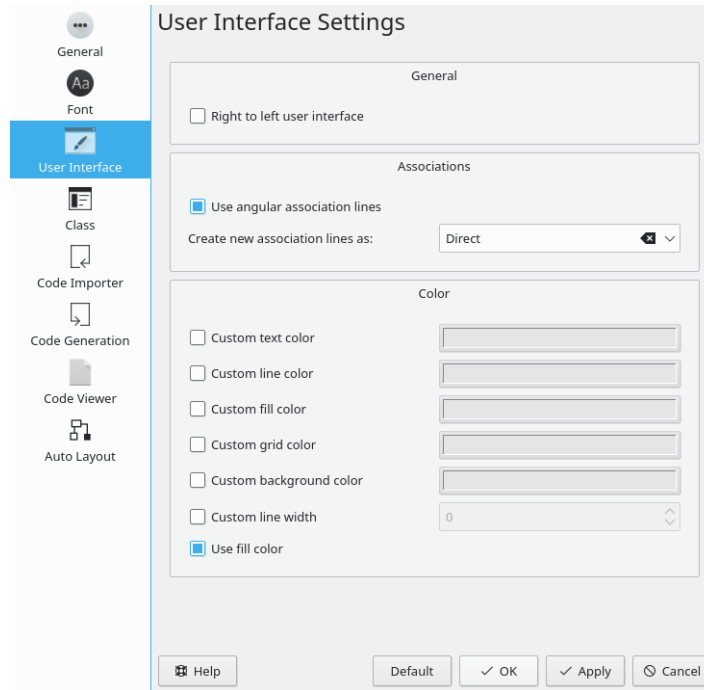
## 6.2 Configurações de fonte



### *Opções para as configurações de fonte do diagrama no Umbrello UML Modeller*

Essas configurações de fonte definem as características do texto nos diagramas. O estilo e o tamanho da fonte são as únicas opções selecionáveis.

## 6.3 Configurações da interface do usuário



*Opções para as configurações da interface do usuário no Umbrello UML Modeller*

### 6.3.1 Geral

**Interface de usuário da direita para a esquerda** configura a interface para idiomas da direita para a esquerda.

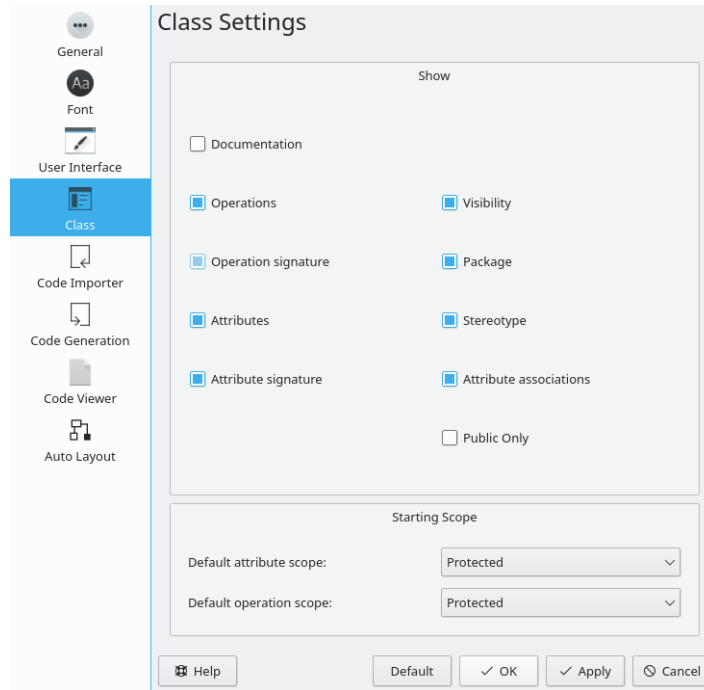
### 6.3.2 Associações

**Criar novas linhas de associação como:** permite alterar o estilo da linha de associação.

### 6.3.3 Cor

A seção de cores oferece diversas opções para alterar as cores do texto, da linha, do preenchimento, da grade e do fundo, bem como a largura da linha.

## 6.4 Configurações da classe



*Opções para as configurações de classe no Umbrello UML Modeller*

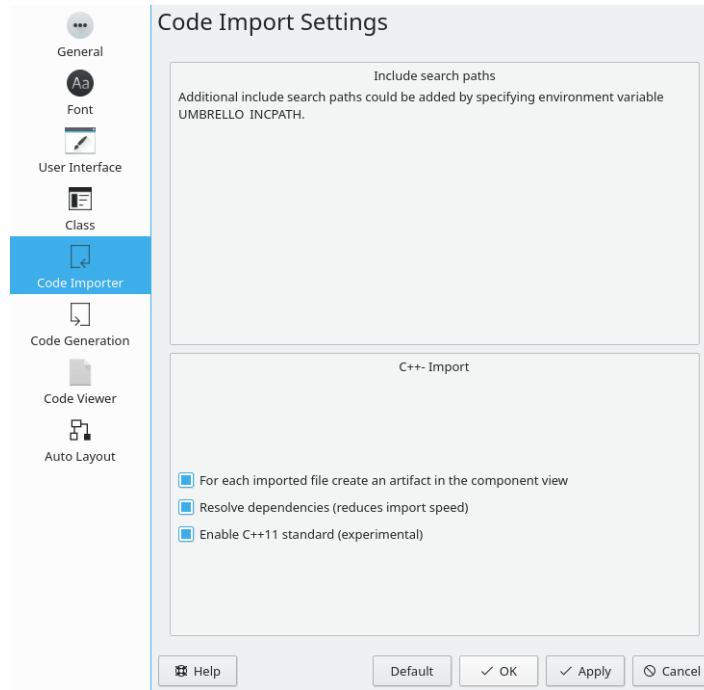
### 6.4.1 Mostrar

A seção Mostrar possui diversas configurações que determinam quais características de classe são mostradas no Diagrama de Classes.

### 6.4.2 Escopo inicial

Opções para configurações padrão de atributos e operações: público, privado ou protegido.

## 6.5 Configurações do importador de código



*Opções para as configurações de importação de código no Umbrello UML Modeller*

### 6.5.1 Incluir os caminhos de pesquisa

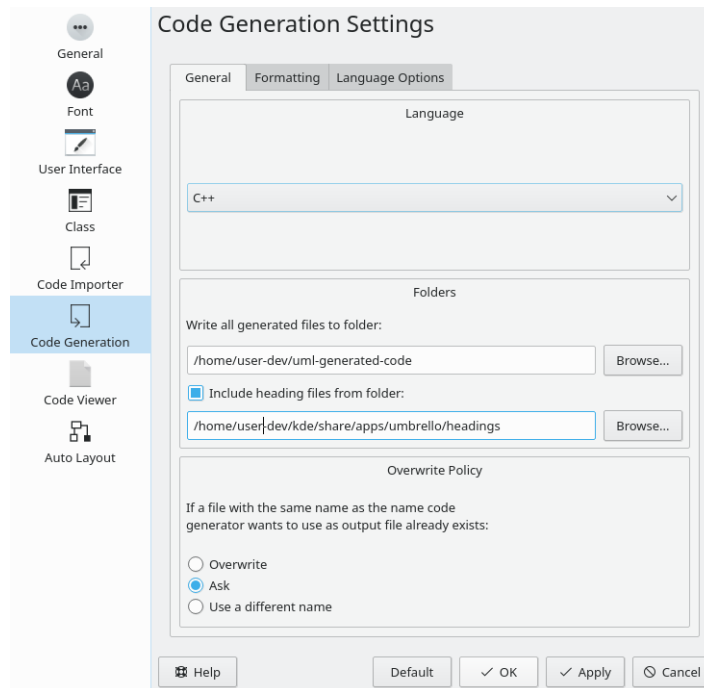
Uma recomendação geral é dada para melhorar a pesquisa incluindo UMBRELLO\_INCPATH como uma variável de ambiente.

### 6.5.2 Importação de C++

- **Para cada arquivo importado, crie um artefato na visualização do componente** O artefato criado pode então ser arrastado para a visualização do Diagrama de Classes, onde as dependências podem ser facilmente visualizadas, juntamente com os atributos e funções de cada arquivo.
- **Resolver dependências (reduz a velocidade de importação)** Garante que todas as dependências de arquivos sejam resolvidas, o que então aparece nas dependências de classe no Diagrama de Classes.
- **Habilitar o padrão C++11 (experimental)** Um recurso experimental para estar em conformidade com o C++11; desabilite-o se não for necessário.

## 6.6 Configurações de geração de código

### 6.6.1 Aba Geral das configurações de geração de código



#### *Opções para as configurações gerais de geração de código no Umbrello UML Modeller*

O Umbrello UML Modeller pode gerar código-fonte para várias linguagens de programação com base no seu modelo UML para ajudar você a começar a implementação do seu projeto. O código gerado consiste nas declarações de classe, com seus métodos e atributos, para que você possa “preencher as lacunas” fornecendo a funcionalidade das operações de suas classes.

#### 6.6.1.1 Linguagem

Escolha a linguagem de programação a ser usada nos projetos. As opções oferecidas são: ActionScript, Ada, C++, C#, D, IDL, Java, JavaScript, MySQL, Pascal, Perl, PHP, PHP5, PostgreSQL, Python, Ruby, SQL, Tcl, Vala e XMLSchema.

#### 6.6.1.2 Pastas

**Salvar todos os arquivos gerados na pasta:** possui um campo editável para o caminho desejado para os arquivos gerados ou, opcionalmente, um botão de navegação para selecionar o caminho.

**Incluir arquivos de cabeçalho da pasta:** Se marcado, permite que o usuário especifique um caminho em um campo editável ou o escolha com um botão de navegação.

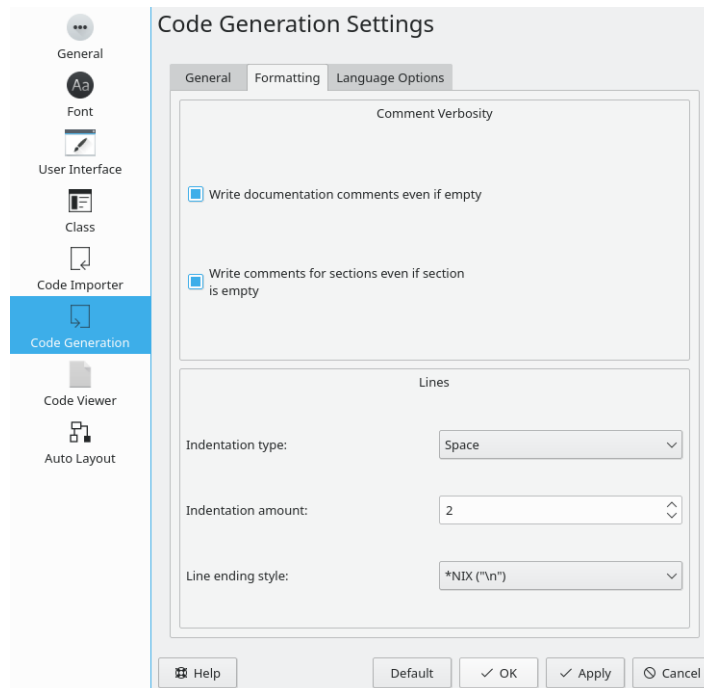
#### 6.6.1.3 Política de sobrescrita

Quando o código é gerado na pasta especificada, esta configuração determina o que acontece se um arquivo com o mesmo nome for encontrado.

- **Sobrescreve** o arquivo sem aviso ou opção.

- **Pergunta** se deseja sobrescrever o arquivo ou renomeá-lo.
- **Usa um nome diferente** quando um arquivo já existir, renomeando-o usando um sufixo.

## 6.6.2 Aba de formatação das configurações de geração de código



*Opções para as configurações de formatação de geração de código no Umbrello UML Modeller*

### 6.6.2.1 Detalhes dos comentários

**Escrever comentários de documentação mesmo que estejam vazios** Gera comentários para classes e funções, mesmo que estejam vazias.

**Escrever comentários da seção** permite que os usuários escolham se desejam escrever comentários na seção. Os valores possíveis são: Nenhum, Quando a seção não estiver vazia, Sempre.

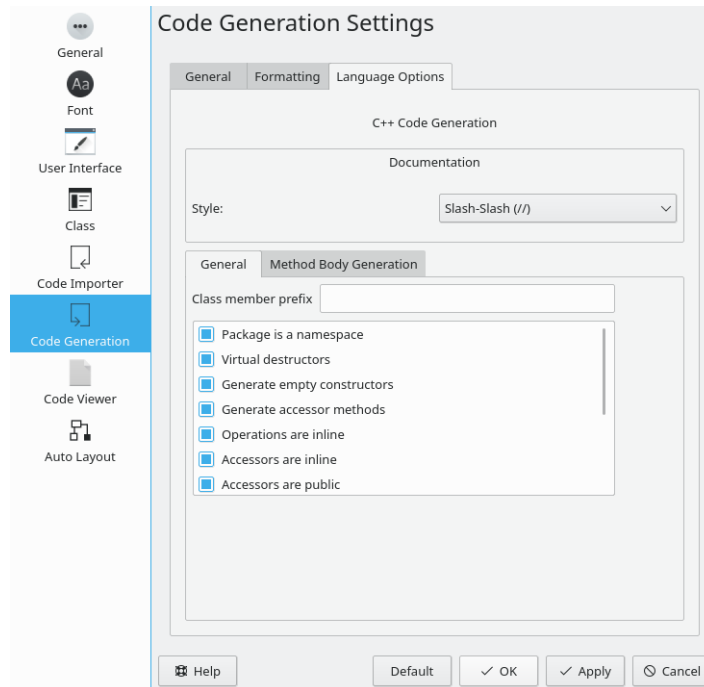
### 6.6.2.2 Linhas

**Tipo de recuo:** oferece uma escolha entre nenhum recuo, tabulação ou espaço.

**Quantidade de recuo:** permite ao usuário especificar o número de espaços para a opção de recuo com tabulação ou espaço.

**Estilo de fim de linha:** é uma escolha entre os estilos de fim de linha do \*NIX, Windows e Mac.

## 6.6.3 Opções de linguagem



*Opções para as configurações de linguagem de geração de código no Umbrello UML Modeller*

Esta página muda para cada linguagem de programação selecionada na aba Geral. Atualmente, as únicas opções disponíveis são para a linguagem C++.

### 6.6.3.1 Geração de código C++

#### 6.6.3.1.1 Documentação

**Estilo:** oferece a opção de usar `/** */` ou `///` como estilo de documentação.

#### 6.6.3.1.2 Geral

Na aba **Geral** da aba **Opções de linguagem**, são listadas diversas opções de geração de código.

- **Prefixo do membro da classe**

Uma opção que permite que um prefixo determinado pelo usuário seja adicionado aos membros da classe quando o código for gerado.

- **Pacote é um espaço de nome**

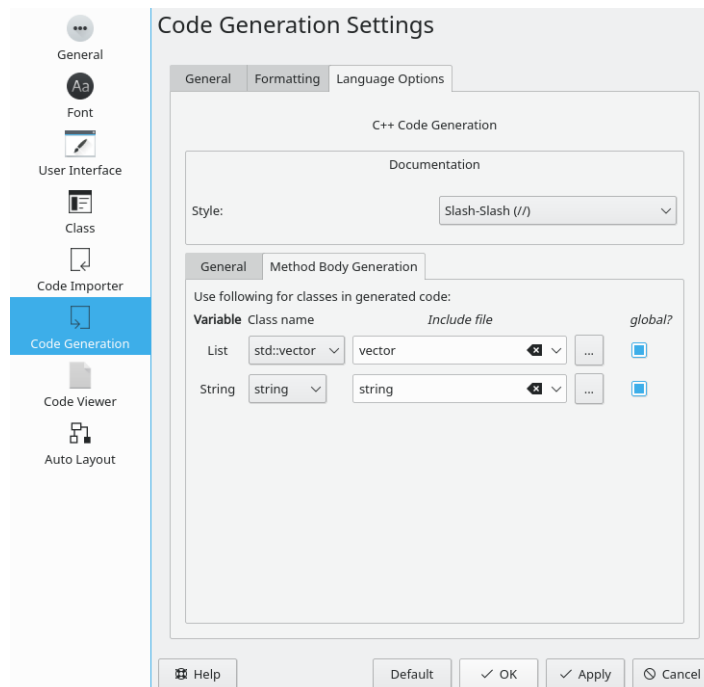
Os espaços de nome fornecem um método para evitar conflitos de nomes em projetos grandes. Os símbolos declarados dentro de um bloco de espaço de nome são colocados em um escopo nomeado que impede que sejam confundidos com símbolos de nome idêntico em outros escopos.

- **Destrutores virtuais**

Embora os destrutores não sejam herdados, se uma classe base declarar seu destrutor como virtual, o destrutor da classe derivada sempre o sobrescreverá. Isso torna possível excluir objetos alocados dinamicamente de tipo polimórfico por meio de ponteiros para a classe base.

- **Construtores vazios gerados**  
Isso irá gerar construtores que possuem chaves vazias.
- **Métodos assessores gerados**  
Irá gerar métodos para acessar tipos de dados.
- **As operações são incorporadas**  
Gero os métodos como incorporados (inline), mas os compiladores são livres para optar por não incorporar o método.
- **Os assessores são incorporados**  
Os métodos que acessam os dados da classe serão gerados inline, mas os compiladores podem optar por não incluir o método em linha.
- **Assessores são públicos**  
Os métodos gerados como públicos estarão disponíveis para qualquer instanciação da classe.
- **Criar getters com o prefixo 'get'**  
Isso adicionará o prefixo "get" aos métodos que obtêm/retornam os dados da classe.
- **Remover prefixo '[a-zA-Z]\_' dos nomes dos métodos assessores**  
Se um prefixo foi inserido em **Prefixo do membro da classe**, isto irá removê-lo.
- **Métodos assessores iniciam com letra maiúscula**  
Isso coloca em maiúscula a primeira letra do nome do método.
- **Usar '\' como etiqueta da documentação ao invés de '@'**  
Uma opção de etiqueta para usar ao documentar os parâmetros de um método.

### 6.6.3.1.3 Método de geração de código



Opções para as configurações do corpo do método da linguagem de geração de código no Umbrello UML Modeller

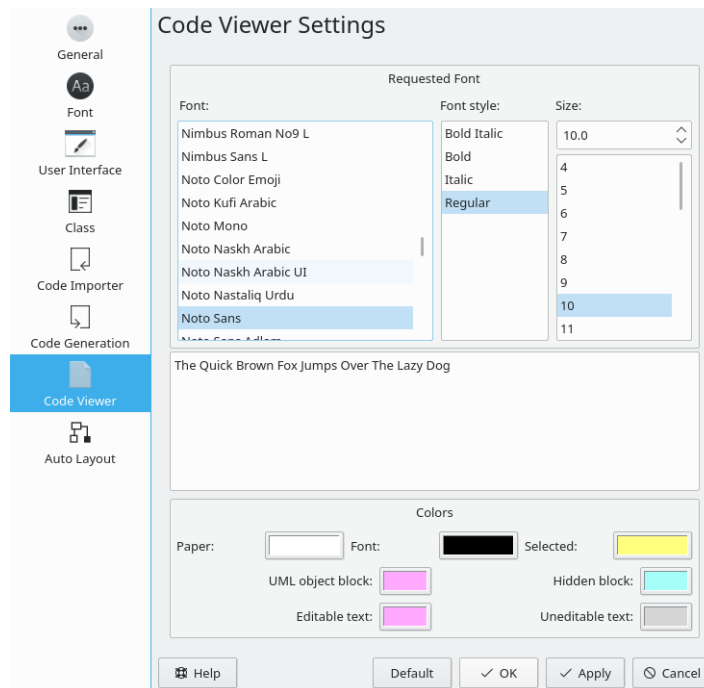
### Lista

Possui opções de QPtrList, vector e std::vector para o tipo de lista. Um campo editável ou selecionável segue para especificar o arquivo de inclusão, juntamente com um botão de navegação para encontrar e selecionar o arquivo de inclusão. Há também uma opção para tornar a lista global.

### String

Opções de string ou QString para o tipo de string. Um campo editável ou selecionável segue para especificar o arquivo de inclusão, juntamente com um botão de navegação para encontrar e selecionar o arquivo de inclusão. Há também uma opção para tornar a string global.

## 6.7 Configurações do visualizador de código

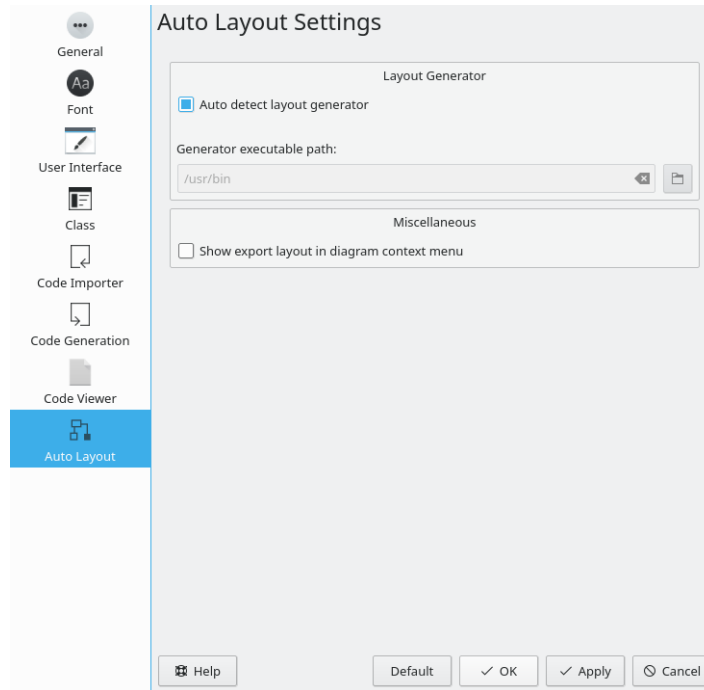


### *Opções para as configurações do visualizador de código no Umbrello UML Modeller*

Permite a personalização do visualizador de código. A seção **Fonte solicitada** permite a seleção da fonte, estilo da fonte e tamanho da fonte. Uma representação das suas escolhas é mostrada abaixo das opções.

Na seção **Cores**, é possível alterar as opções Papel, Fonte, Selecionado, Bloco de objeto UML, Bloco oculto, Texto editável e Texto não editável. As alterações de cor podem ser feitas clicando na caixa de cor ao lado do rótulo correspondente.

## 6.8 Configurações de layout automático



*Opções para as configurações de layout automático no Umbrello UML Modeller*

### **Detectar automaticamente o gerador de layout**

O recurso de layout automático depende dos geradores de layout fornecidos pelo pacote GraphViz, que normalmente é instalado junto com o Umbrello por um gerenciador de pacotes. O Umbrello possui suporte integrado para detectar o local de instalação desses geradores de layout. Nos casos em que essa dependência não estiver disponível ou não for adequada, um caminho de instalação diferente poderá ser selecionado.

### **Mostrar a exportação do layout no menu de contexto do diagrama**

A exportação de arquivos DOT é realizada usando o layout de exportação. Com esta opção marcada, o layout de exportação é adicionado à lista de layouts de diagrama disponíveis e permite uma pré-visualização rápida da exportação DOT.

## Capítulo 7

# Autores e histórico

Este projeto foi iniciado por Paul Hensgen como um de seus projetos Universitários. O nome original do aplicativo era Modelador UML. Paul fez todo o desenvolvimento até o final de 2001 quando o programa atingiu sua versão 1.0.

A versão 1.0 já oferecia muitas funcionalidades, mas após o projeto ter sido revisado pela Universidade de Paul, outros desenvolvedores puderam juntar-se e iniciaram fazendo contribuições importantes para o Modelador UML, como alternar de um formato binário de arquivo para um arquivo XML, suporte para mais tipos de Diagramas UML, Geração de Código e Importação de Código apenas para citar algumas.

Paul retirou-se da equipe de desenvolvimento no Verão de 2002 mas, como um Software Livre e de Código Aberto, o programa continua a melhorar e evoluir, e está sendo mantido por um grupo de desenvolvedores de diferentes partes do mundo. Em Setembro de 2002, o projeto mudou seu nome de Modelador UML, para Umbrello UML Modeller. Existiram várias razões para a mudança do nome, a mais importante sendo que apenas 'uml' — como é comumente conhecido — é um nome muito genérico e causa problemas com alguns distribuidores. A outra razão importante é que os desenvolvedores acham Umbrello um nome mais legal.

O desenvolvimento do Umbrello UML Modeller assim como as discussões sobre para onde o programa deve progredir nas versões futuras é aberto e ocorre na Internet. Se você gostaria de contribuir para o projeto, por favor não hesite em contactar os desenvolvedores. Existem muitas maneiras nas quais você pode auxiliar o Umbrello UML Modeller:

- Relatando erros ou sugerindo melhorias
- Corrigindo erros e adicionando recursos
- Escrevendo uma boa documentação ou traduzindo para outros idiomas
- E é claro...codificando conosco!

Como você viu, existem muitas maneiras de contribuir. Todas elas são muito importantes e todos são bem-vindos a participar.

Os desenvolvedores do Umbrello UML Modeller podem ser contactados em [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org).

## Capítulo 8

# Direitos autorais

Direitos autorais 2001, de Paul Hensgen

Direitos autorais 2002-2020, dos autores do Umbrello UML Modeller.

Esta documentação é licenciada sob os termos da [Licença de Documentação Livre GNU](#).

Este programa é licenciado sob os termos da [Licença Pública Geral GNU](#).