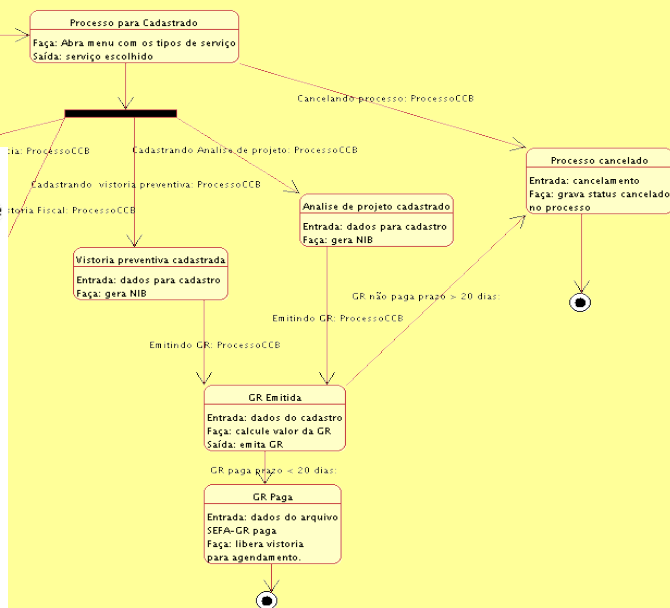
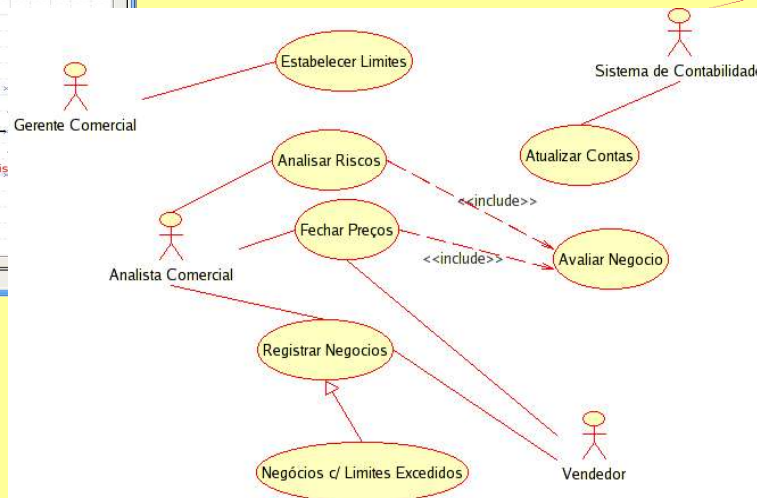
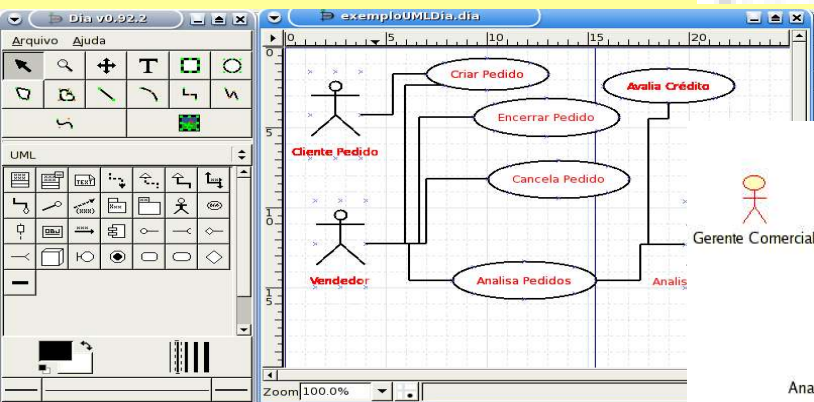


Ferramentas Livres para UML



Palestrante: Marcio Junior Vieira



www.ambientelivre.com.br

Palestrante – Marcio Junior Vieira.

- A 7 Anos Trabalha da área de Desenvolvimento e Análise de sistemas.
- A 5 anos trabalha com Software Livre.
- Formado em Tecnologia em Informática – UFPR.
- Pós-Graduação em Informática com ênfase em Software Livre – UFPR.
- Consultor de Soluções em Software Livre pela Ambiente Livre Tecnologia.
- Analista de Sistemas / Consultor – TI-Recursos Humanos pela Gci Informática.
- Participante das comunidades PSL-PR, Projava.
- Colaborador do Time de Internacionalização do KDE (Pt_BR).
-
- Site pessoal www.ambientelivre.com.br/~marcio/

Introdução

- A Linguagem UML.
- A origem da pesquisa de SL para UML.
- Os Software Livres para UML e suas características (Dia, Umbrello, ArgoUML).
- Uma breve introdução a utilização dos Softwares.
- Do que estes softwares são capazes.
- Quais os suportes a UML.
- Licenças, Desenvolvedores, Dicas, Versões, Projetos relacionados e Plataformas suportadas.
- Muitos Exemplos de Diagramas UML.



A Linguagem UML

- Surgiu no final dos anos 80 e no início dos anos 90.
- É uma linguagem de modelagem de projetos Orientados a Objetos, padrão da OMG.
- É a notação (principalmente gráfica) utilizada por métodos para expressar projetos.
- Utilizado para modelagem de sistemas na maioria das empresas que desenvolvem Software OO.

A Pesquisa.

- Origem: Acadêmicos de **Tecnologia em Informática** da UFPR precisavam de SL para Modelagem UML (2004).
- 2 Meses em pesquisas e testes de **Softwares Livres** para **Modelagem UML**.
- 5 Meses utilizando os Softwares no dia-a-dia em Projetos (acadêmicos e corporativos).
- Acompanhamento contínuo de evoluções dos Softwares e participação na tradução para Português (Umbrello).



Universidade Federal do Paraná

Ferramentas para UML

Comerciais e “Free Editions”

- MagicDraw
- Togheter
- Poseidon
- Enterprise Architect
- Rational Rose
- Jude
- Omondo Eclipse UML

Livres (BSD e GPL)

- Umbrello
- ArgoUML
- Dia

- BOUML
- Fajuba
- StarUML

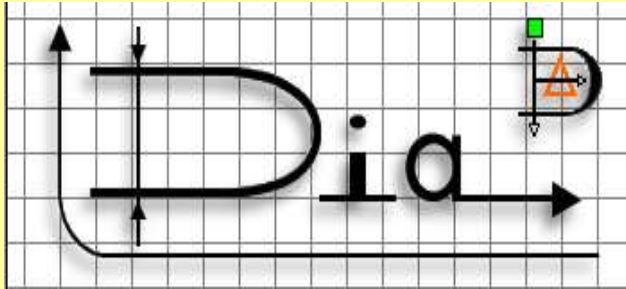
Ferramentas para UML - Tabela

Comerciais e “Free Editions”

- MagicDraw (\$ 1,599,00)
- Together (\$ 11.500,00)
- Poseidon (\$ 1.549,00)
- Enterprise Architect
(\$ 2.500,00)
- Rational Rose (\$ 5.995,00)
- Jude(\$280,00 1usuário/1ano)
- Omondo Eclipse UML
(\$ 18.136,31 / 5 usuários)

Livres (BSD e GPL)

- Umbrello +
- ArgoUML +
- Dia = (\$ 0,00 e ainda
leva os fontes)
- BOUML
- Fajuba



Dia UML

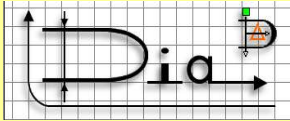
- Dia é um programa baseado em gtk+ para criação do diagrama, liberado sob a licença do GPL.
- É parte do projeto Gnome.
- Atualmente tem objetos especiais de Lógica, entidade e relacionamento, diagramas UML, fluxogramas , diagramas da rede, e circuitos simples entre outros.



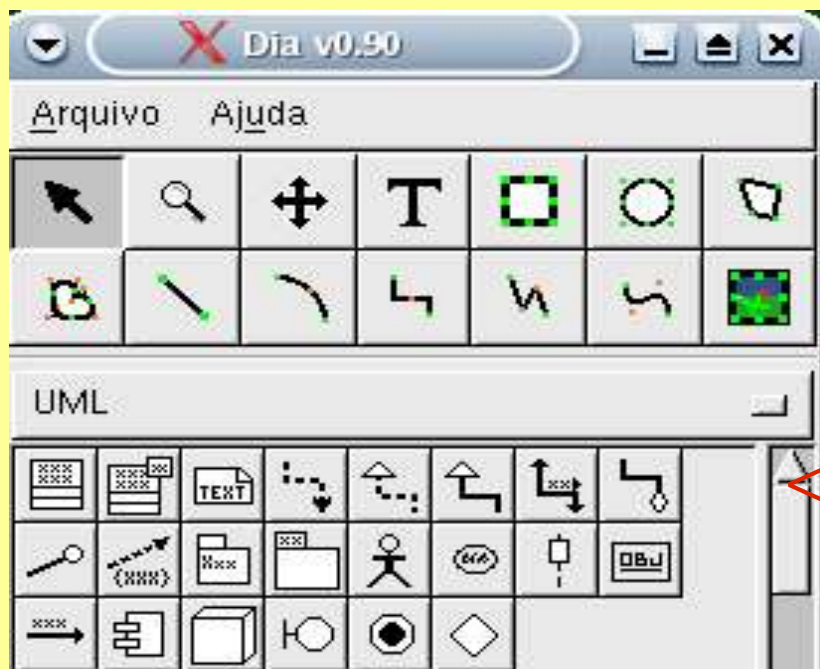
Plataformas Suportadas

- Gnu/Linux.
- Variações de Unix.
- Windows (Win32).

Do que o Dia e capaz em relação a UML?



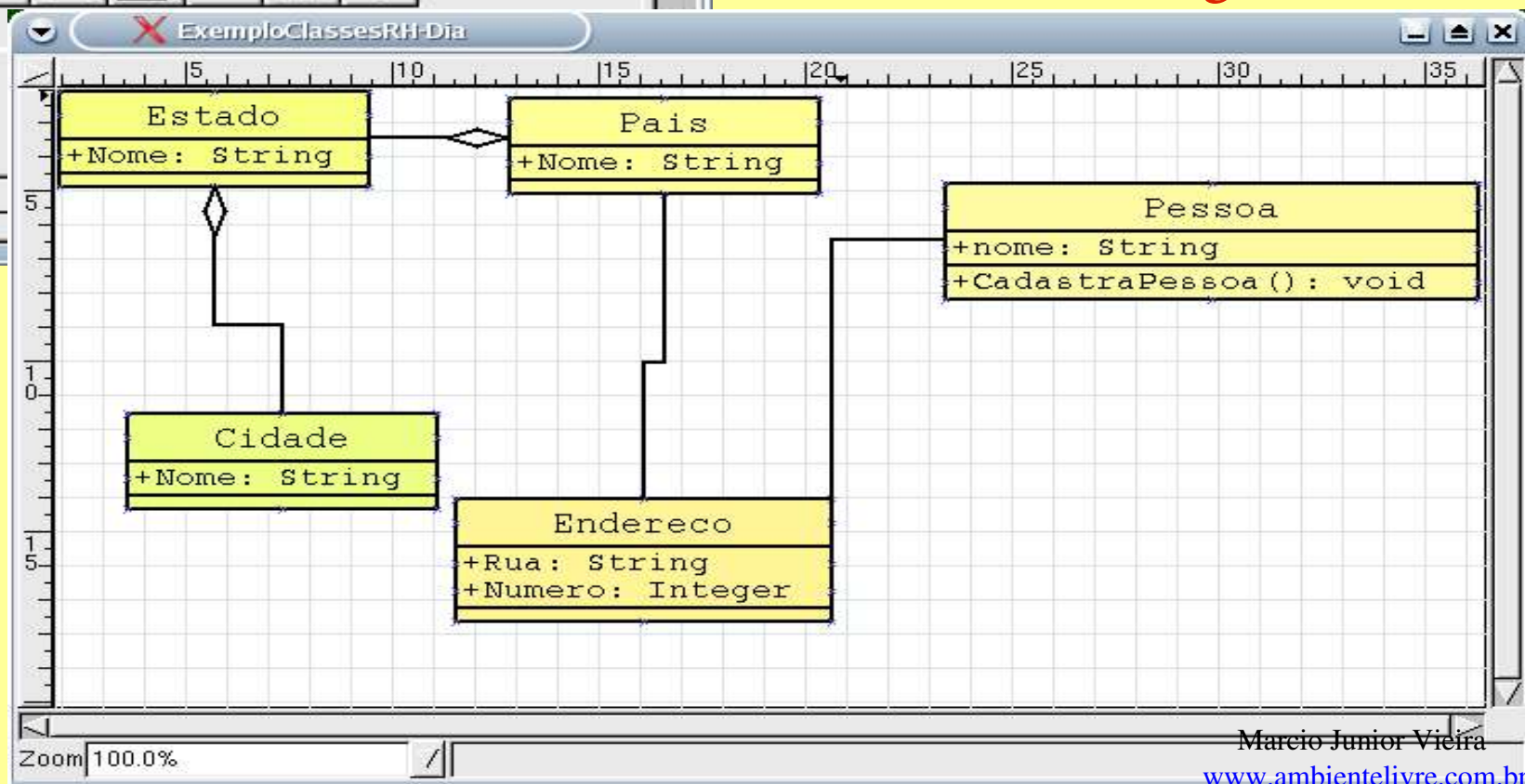
- Com o Dia podem ser criados a maioria dos Diagramas da UML , sua estrutura e simplesmente a união de componentes/objetos utilizados na Modelagem UML.
- Podem ser exportados para arquivos gráficos.
- Os diagramas podem ter componentes/Objetos de outras estruturas como ER, Fluxograma, Rede,Lógica etc.
- Personalização de Barras de Elementos.



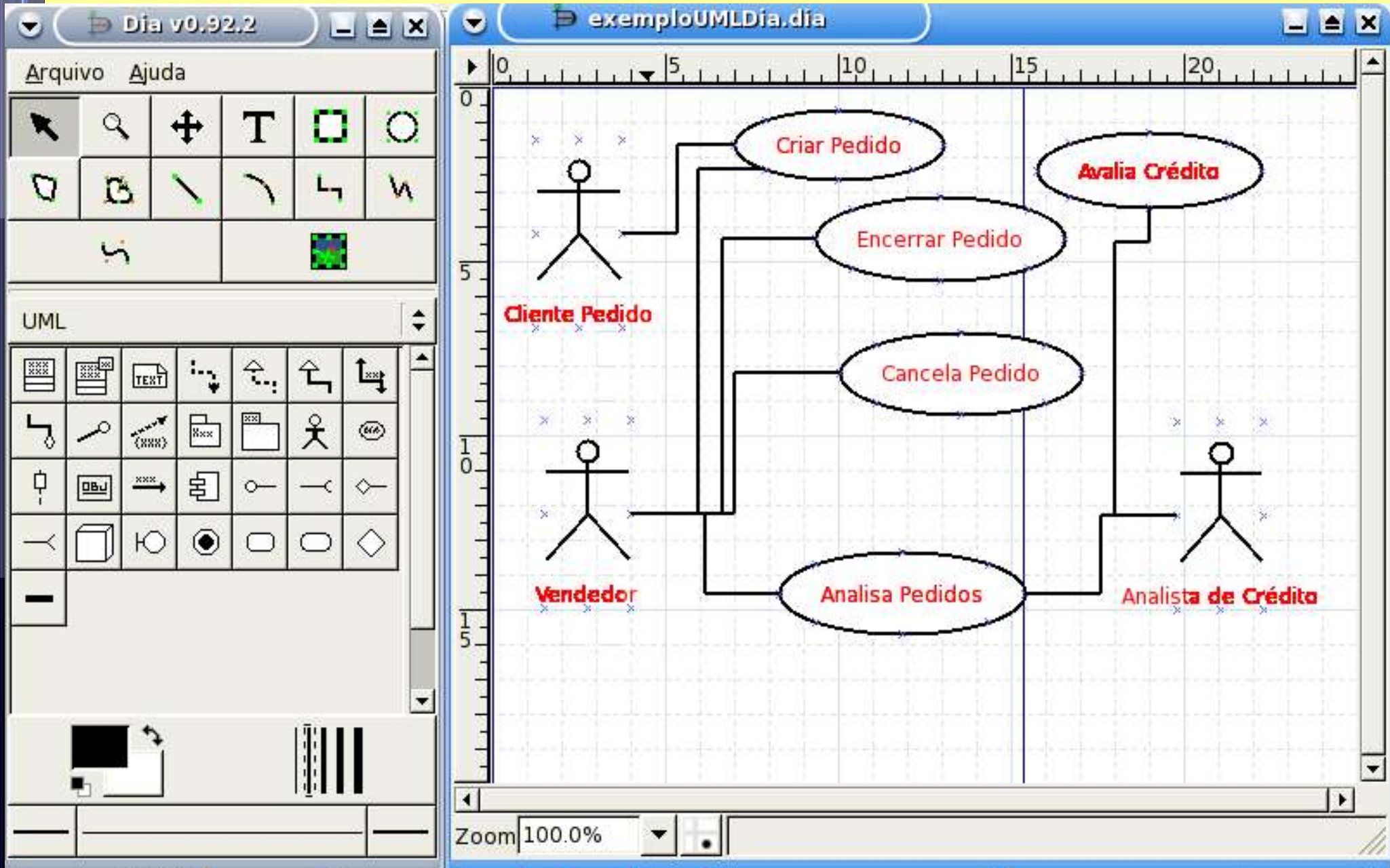
Dia UML

Tipo de Diagramas
←-- Objetos UML

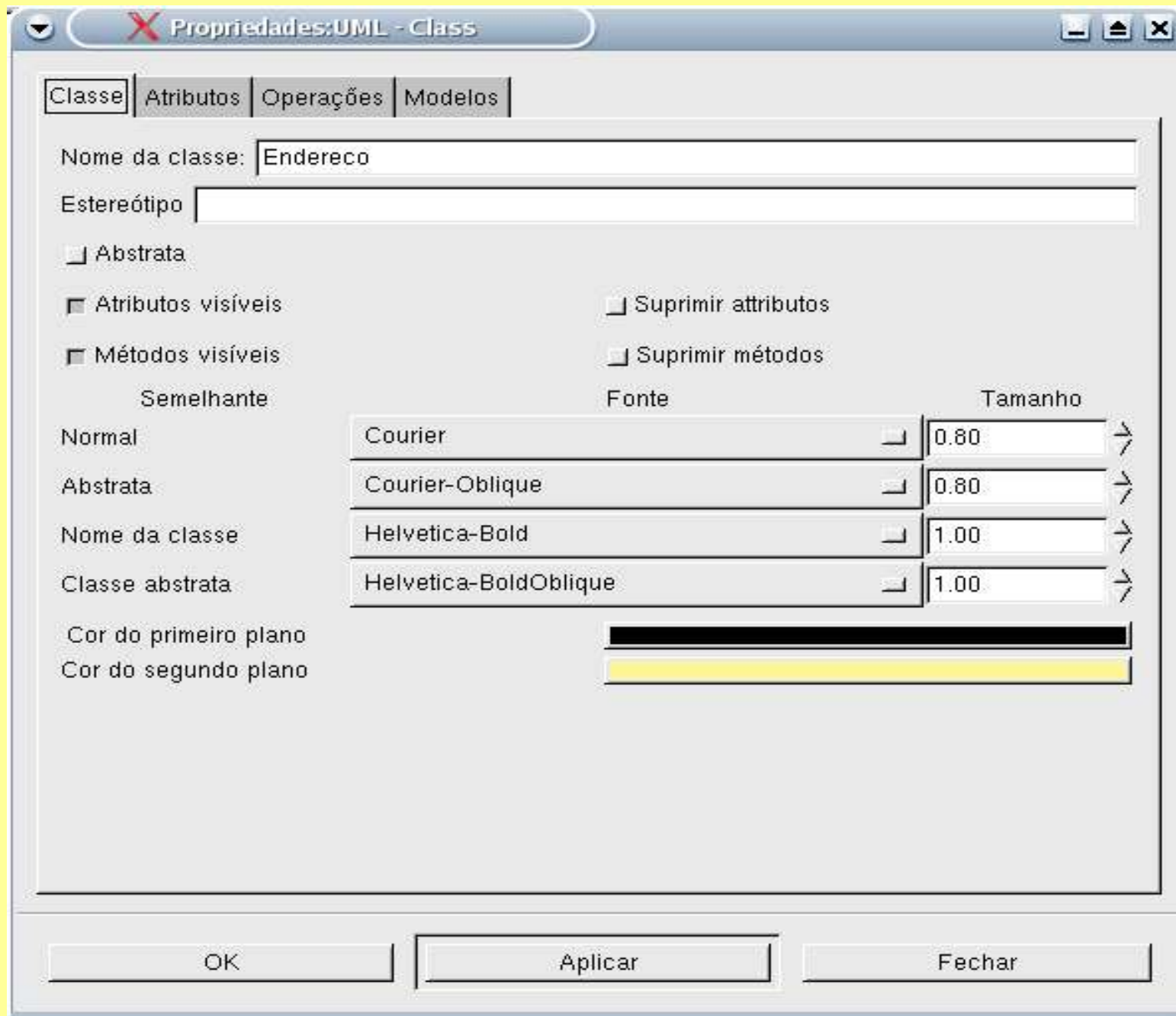
Editor de diagramas



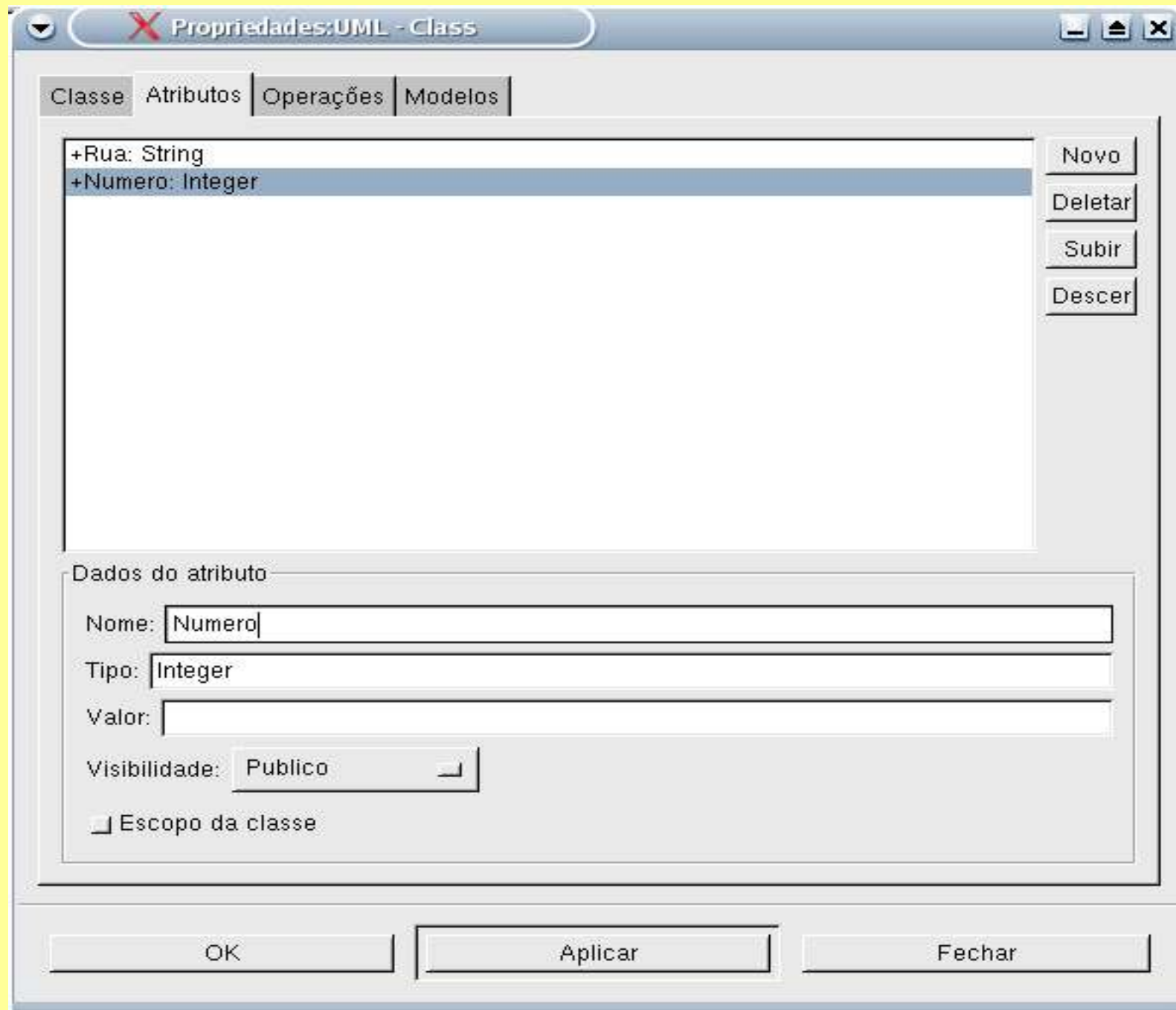
Dia UML



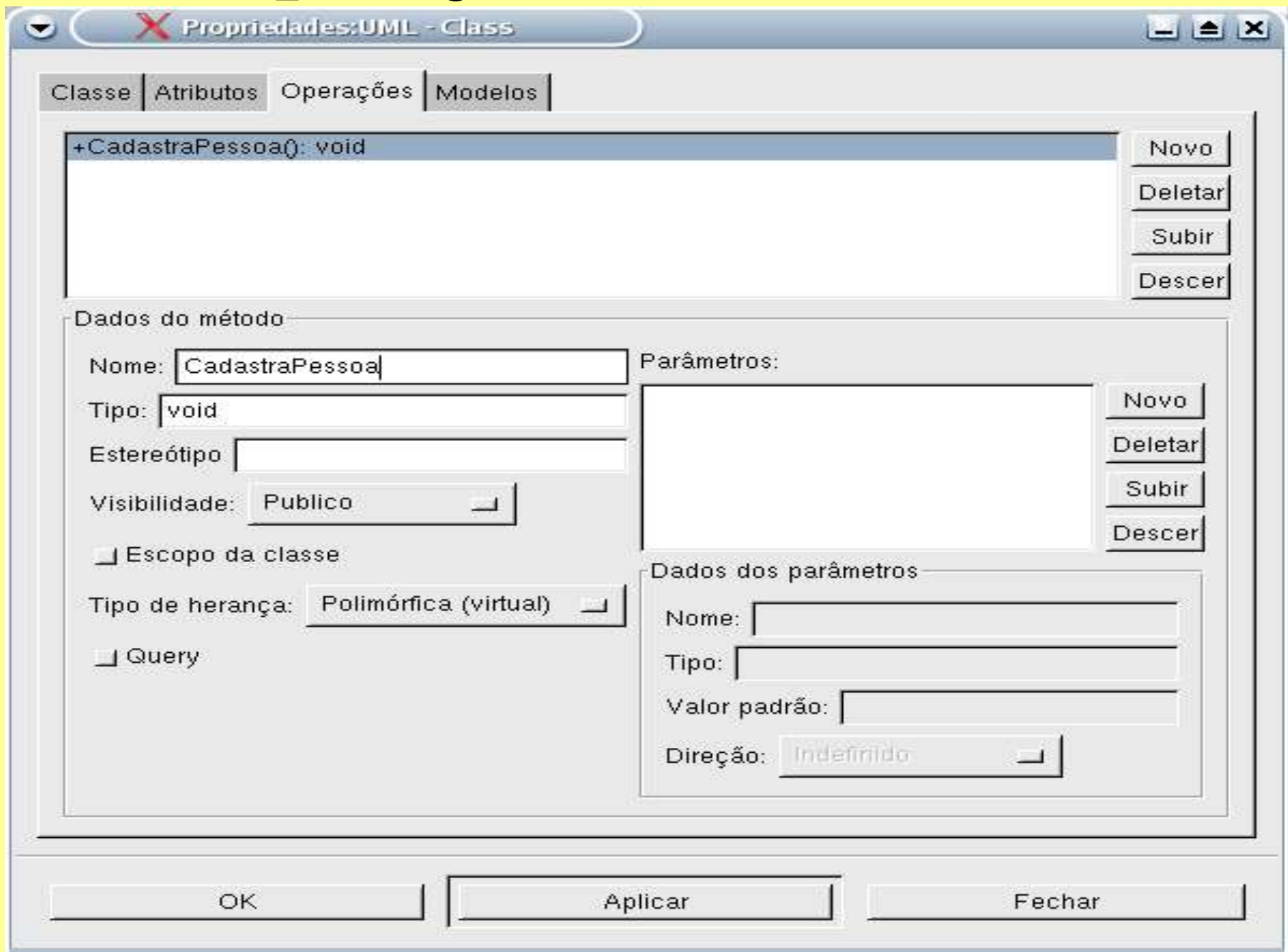
Propriedades das Classes



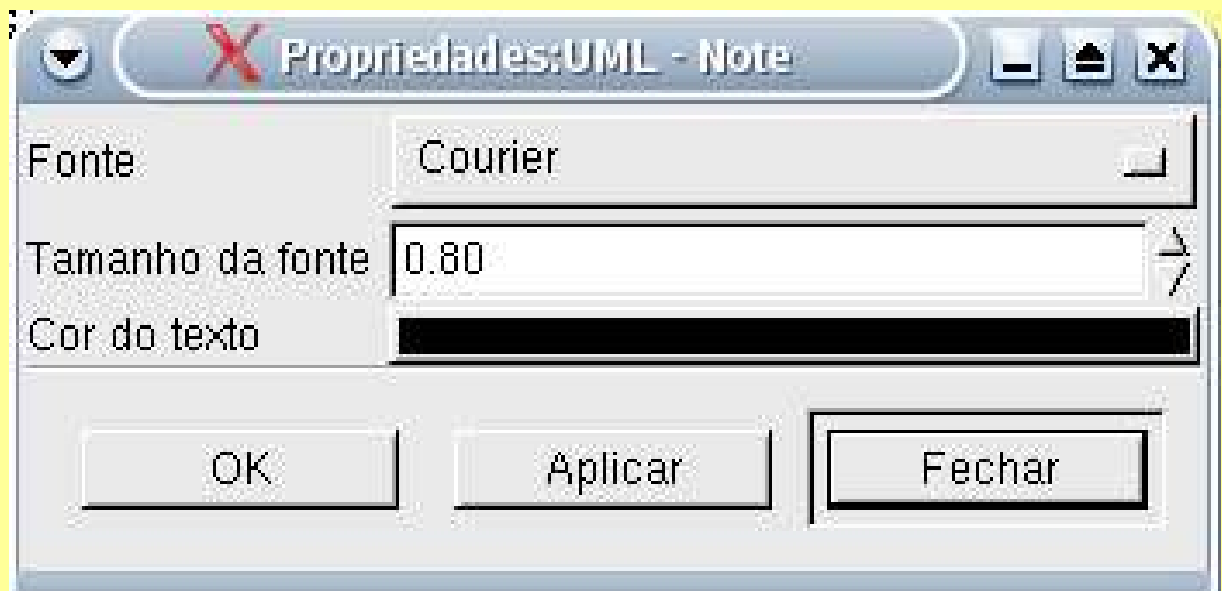
Atributos de Classes



Operações das Classes



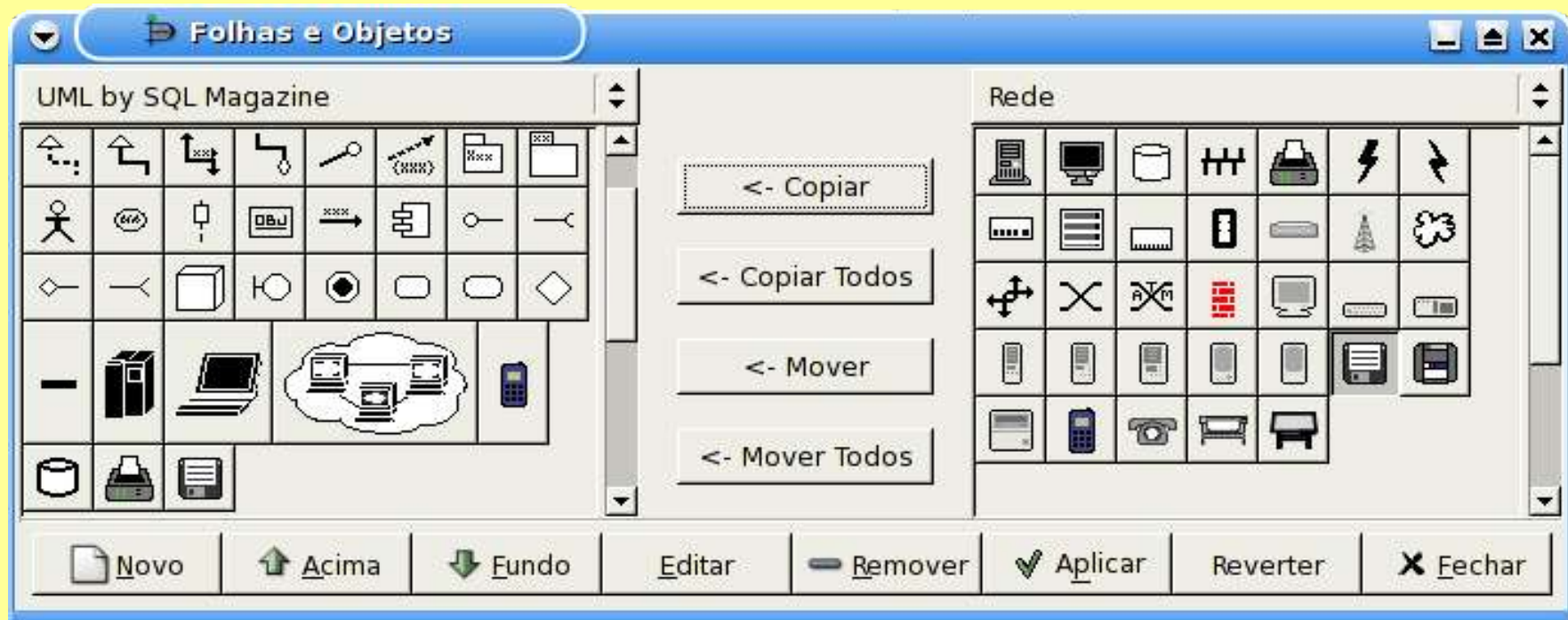
Propriedades de Mensagens e Nota





Personalização de Barra de Componentes

- Podemos fazer um mix de componentes da UML com os demais componentes suportados pela ferramenta.
- A ferramenta permite também que seja criada uma folha de objetos (UML , UML by Ambiente Livre , UML c/ ER ,etc)



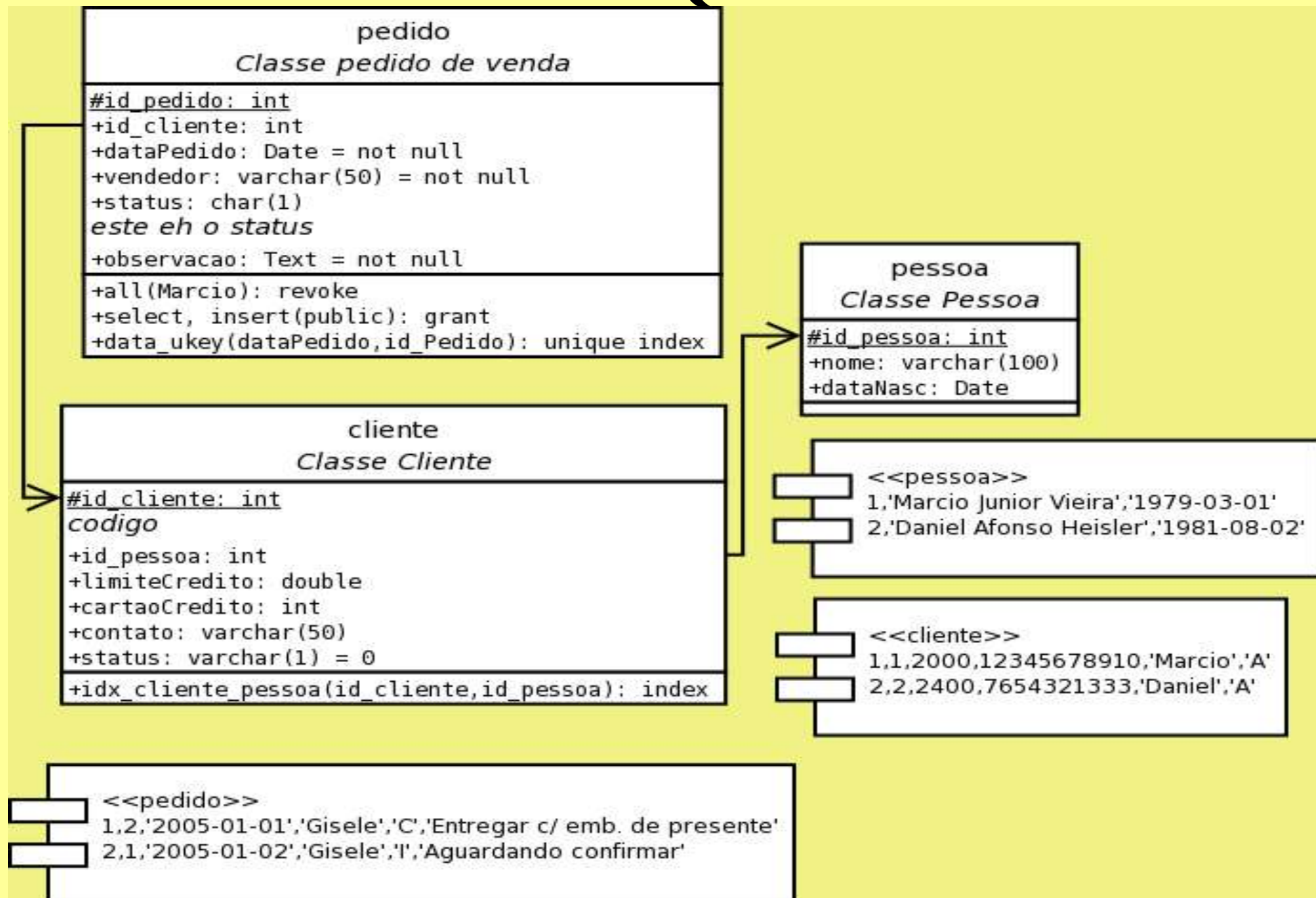
Fonte : Revista SQL Magazine



Informações Adicionais

- A versão atual do Dia é a 0.94 (em teste 0.95pre) .
- **ArgoUML2Dia** converte-se do formato de ArgoUML ao **Dia**, embora todos os tipos do objeto não são suportados ainda.
- **Agata** (gerador de relatório). Pode gerar diagramas do Dia através de Bases SQL.
- **TeDia2SQL** e **Dia2SQL** Converter diagramas do Dia para bases de dados do SQL (Sybase, PostgreSQL, oracle, DB/2, Ms-sql, MySQL)
- A maioria das distribuições Gnu/Linux já vem com o Dia nos CDs de Instalação.

Dia2SQL



Executando o Dia2SQL - Script PHP

```
php dia2sql.php -f <formato> -i <arquivoDiaUML.dia> -o meuarquivogerado.sql  
pgsql – PostgreSQL  
mysql – MySQL
```

site do dia2SQL - <http://codigolivre.org.br/projects/dia2sql/>

-- Table: cliente

-- Purpose: Classe Cliente

CREATE TABLE cliente

```
(  
  id_cliente    int PRIMARY KEY AUTO_INCREMENT,  
  id_pessoa     int,  
  limiteCredito double,  
  cartaoCredito int,  
  contato       varchar(50),  
  status        varchar(1)
```

);

ALTER TABLE cliente COMMENT = 'Classe Cliente';

ALTER TABLE cliente MODIFY id_cliente int NOT NULL;

ALTER TABLE cliente DROP PRIMARY KEY;

ALTER TABLE cliente ADD PRIMARY KEY (id_cliente);

ALTER TABLE cliente ADD INDEX idx_cliente_pessoa (id_cliente,id_pessoa);

ALTER TABLE cliente ADD FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa);

INSERT INTO cliente VALUES (1,1,2000,12345678910,'Marcio','A');

INSERT INTO cliente VALUES (2,2,2400,7654321333,'Daniel','A');

Engenharia Reversa da Bases SQL:

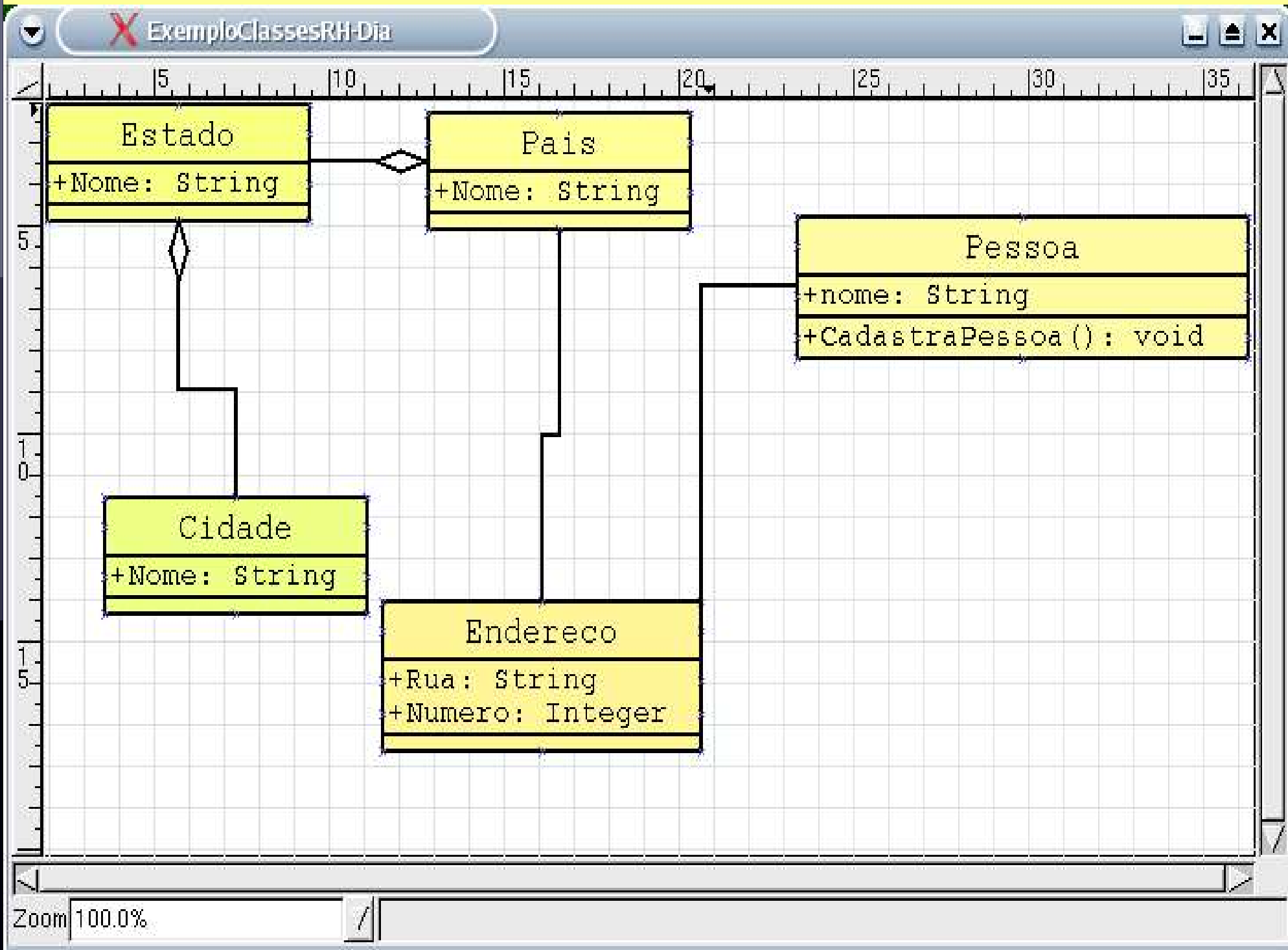


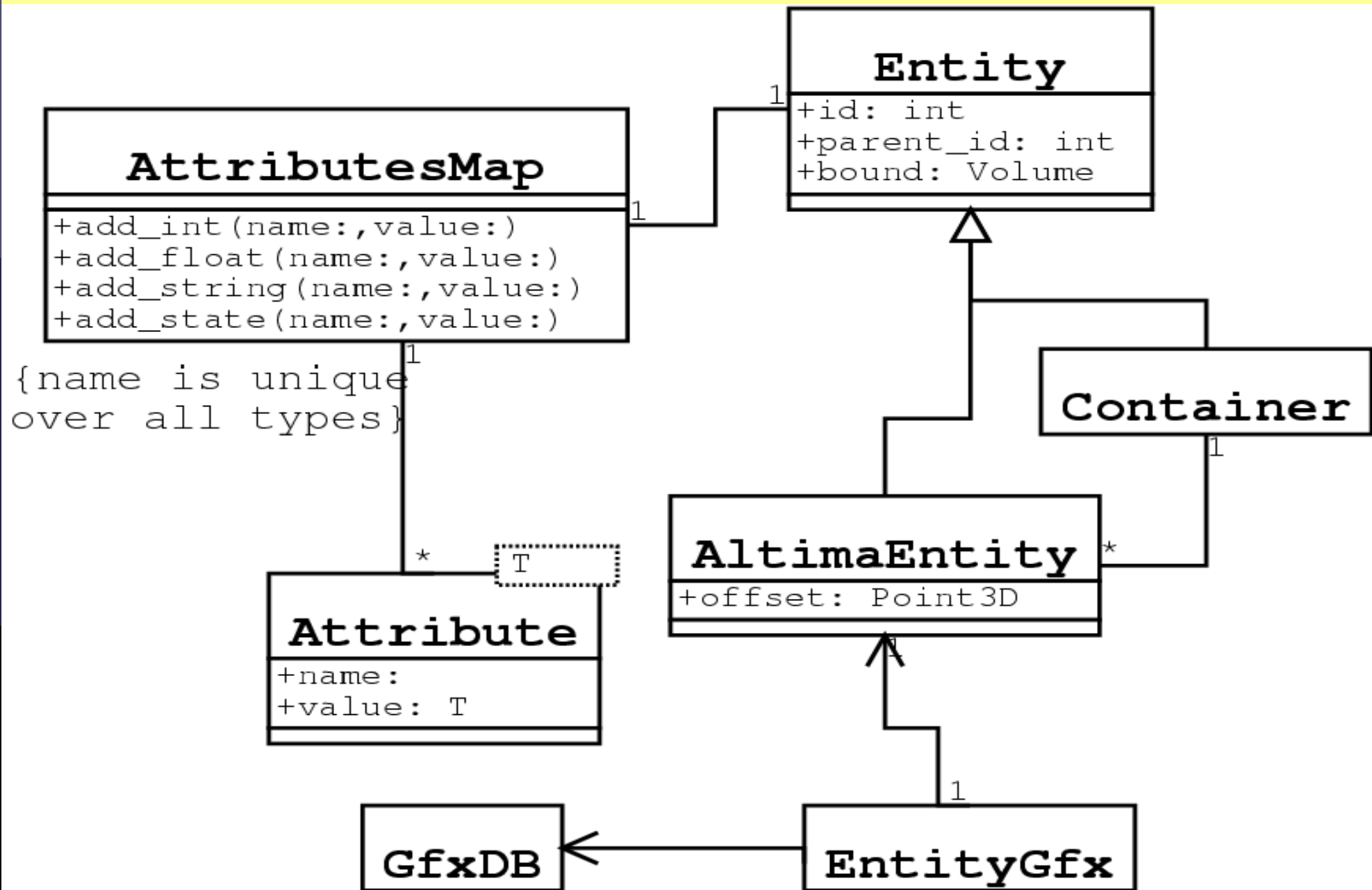
Dia+AgataReport

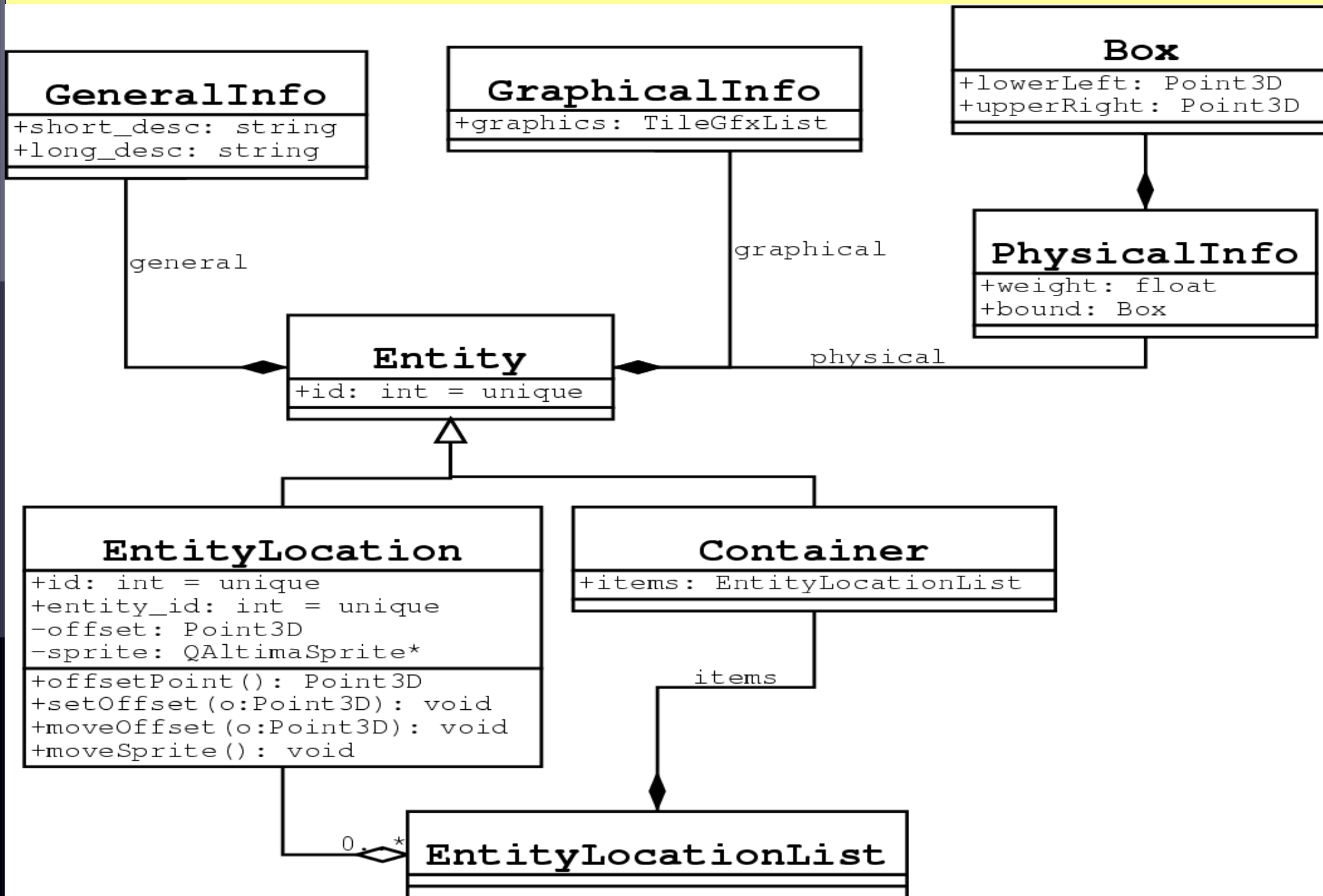
ExemploDia

The screenshot displays the Dia UML tool interface. The main workspace shows a class diagram with three classes: **pessoa**, **pedido**, and **cliente**. The **pessoa** class has attributes `+id_pessoa: int`, `+nome: string`, and `+dataNasc: date`. The **pedido** class has attributes `+id_pedido: int`, `+id_cliente: int`, `+dataPedido: date`, `+vendedor: string`, `+status: string`, and `+observacao: blob`. The **cliente** class has attributes `+id_cliente: int`, `+id_pessoa: int`, `+limiteCredito: real`, `+cartaoCredito: int`, `+contato: string`, and `+status: string`. There are associations between **pessoa** and **pedido**, and between **pedido** and **cliente**.

On the left, the 'Configuração :: dia' panel shows settings for the diagram, including 'Quantidade de linhas do diagrama' (set to 2), 'Exibir linhas de ligação' (checked), and 'Distância entre linhas' (set to 1). Below this is the 'Escolha as tabelas' panel, which lists the tables 'cliente', 'pedido', and 'pessoa' under 'Todas Tabelas'. An 'Adicionar' button is next to the list. At the bottom right, there is a 'Criar Diagrama' button and a vertical 'Agata Report' logo.

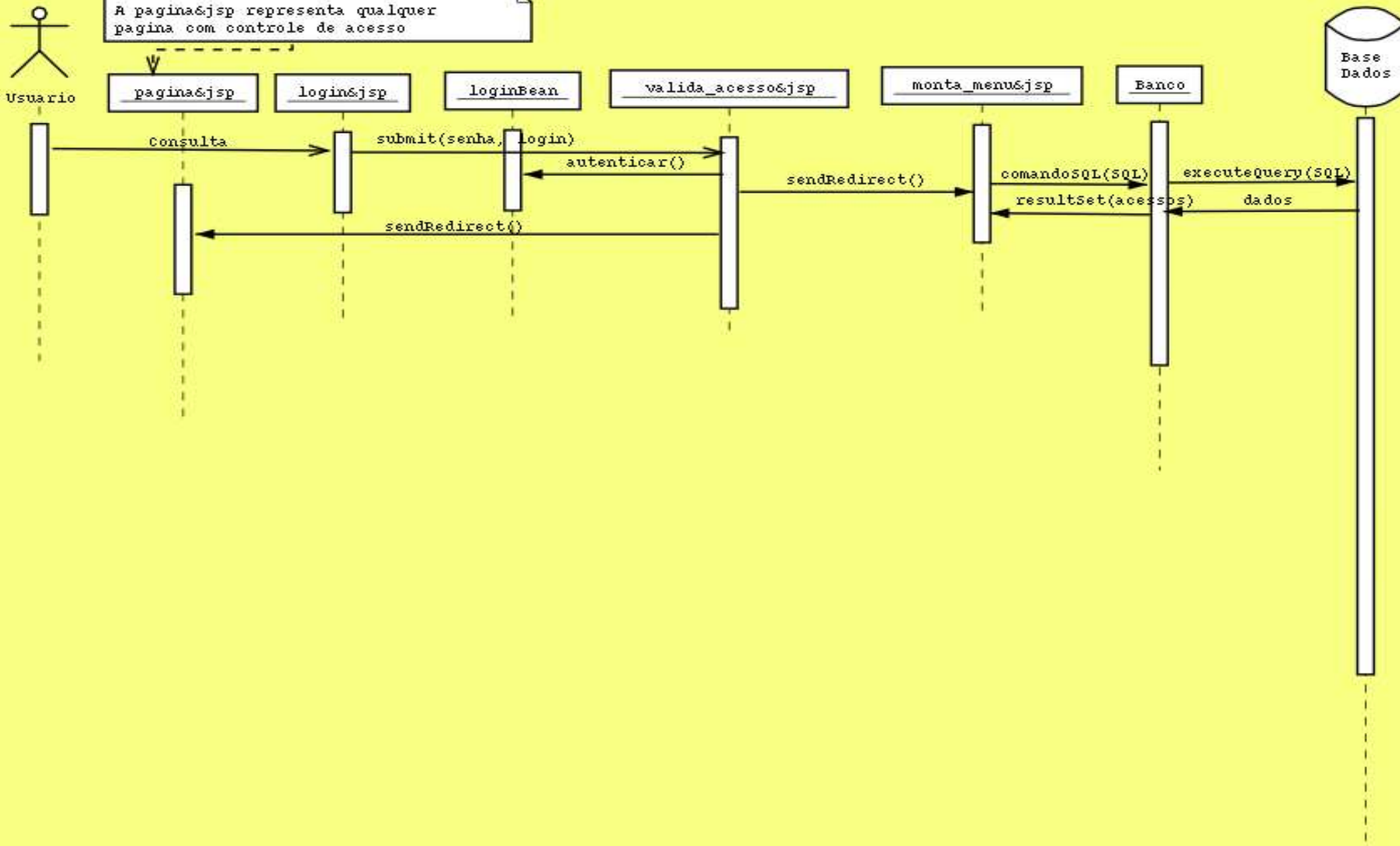


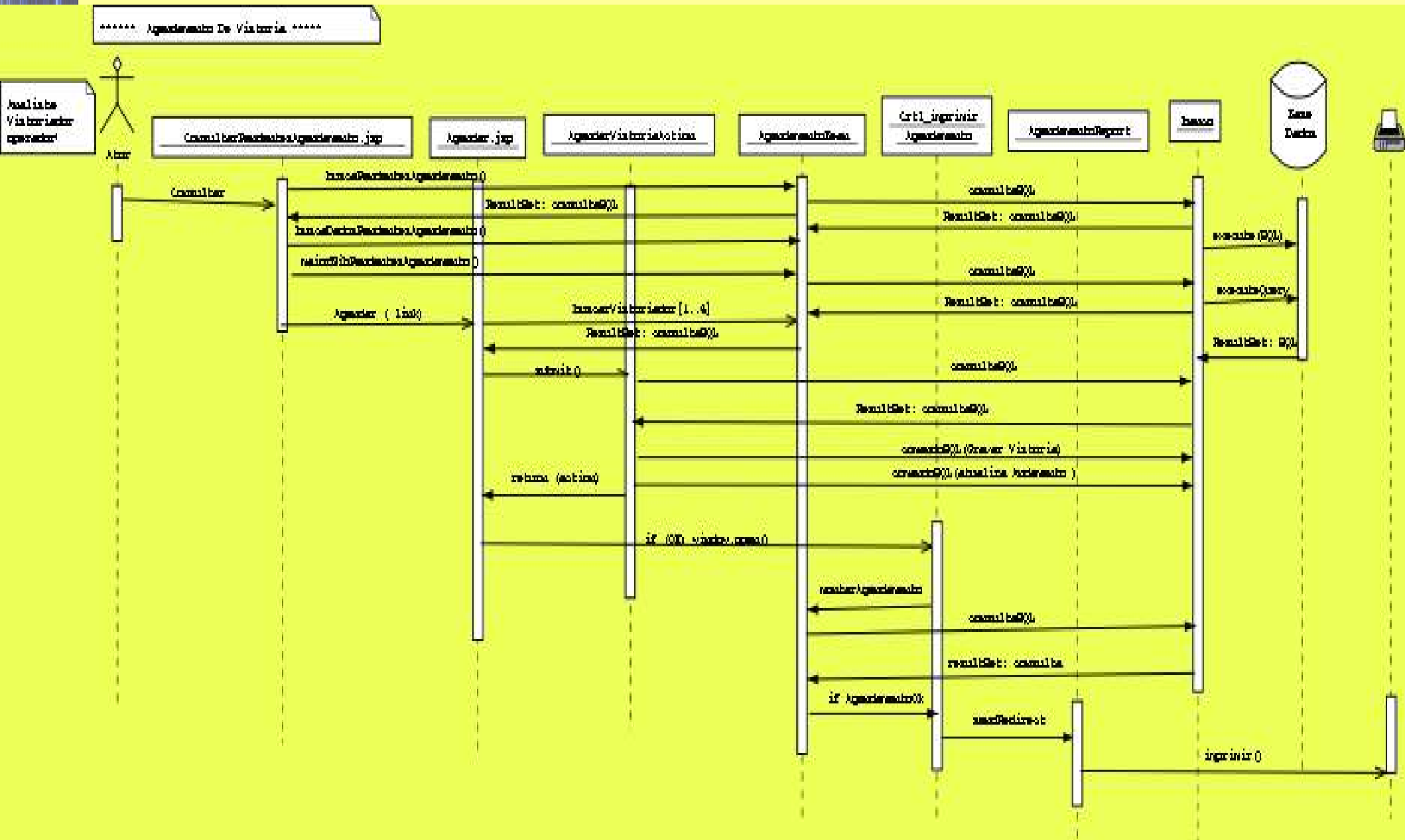


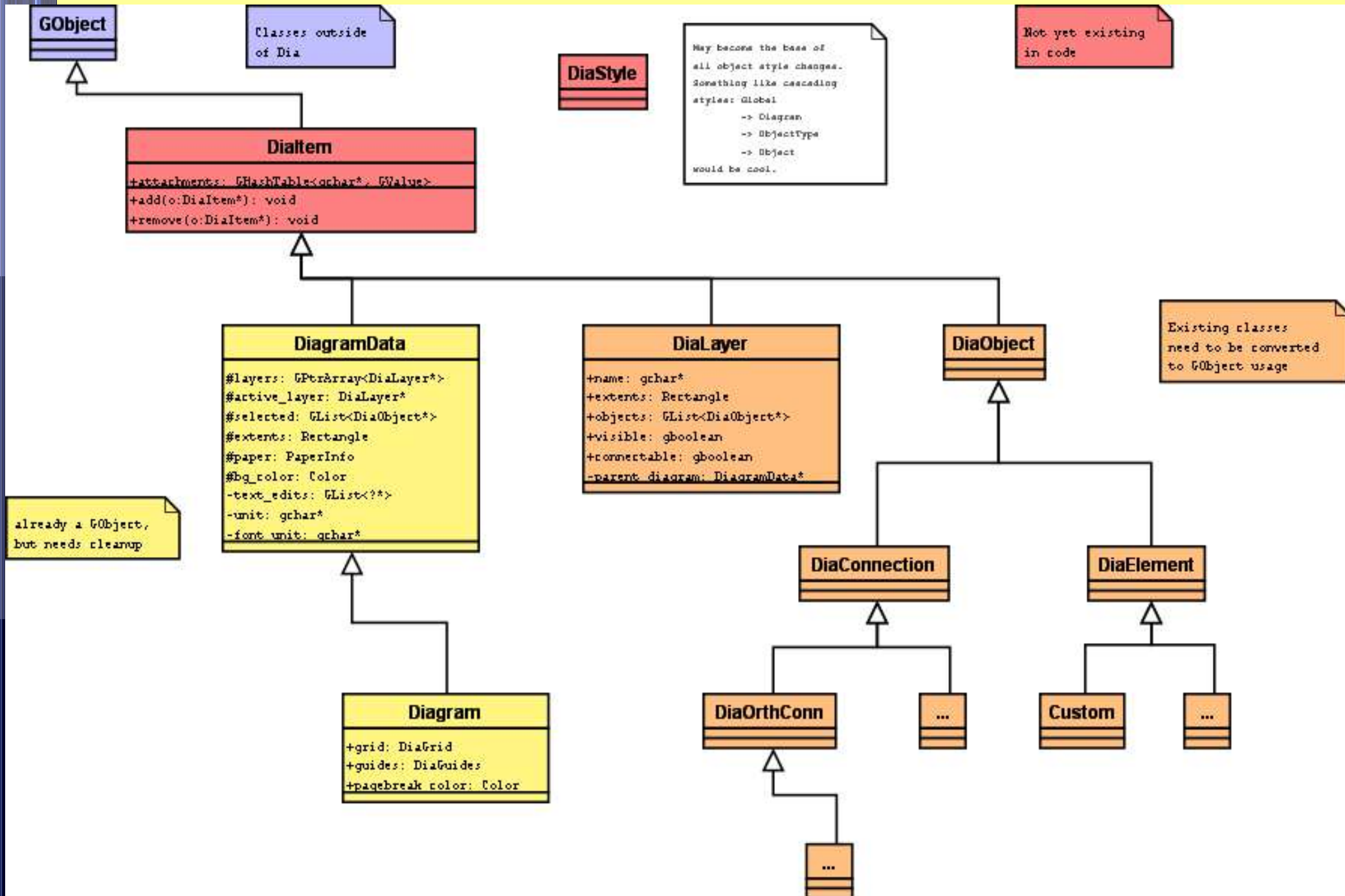


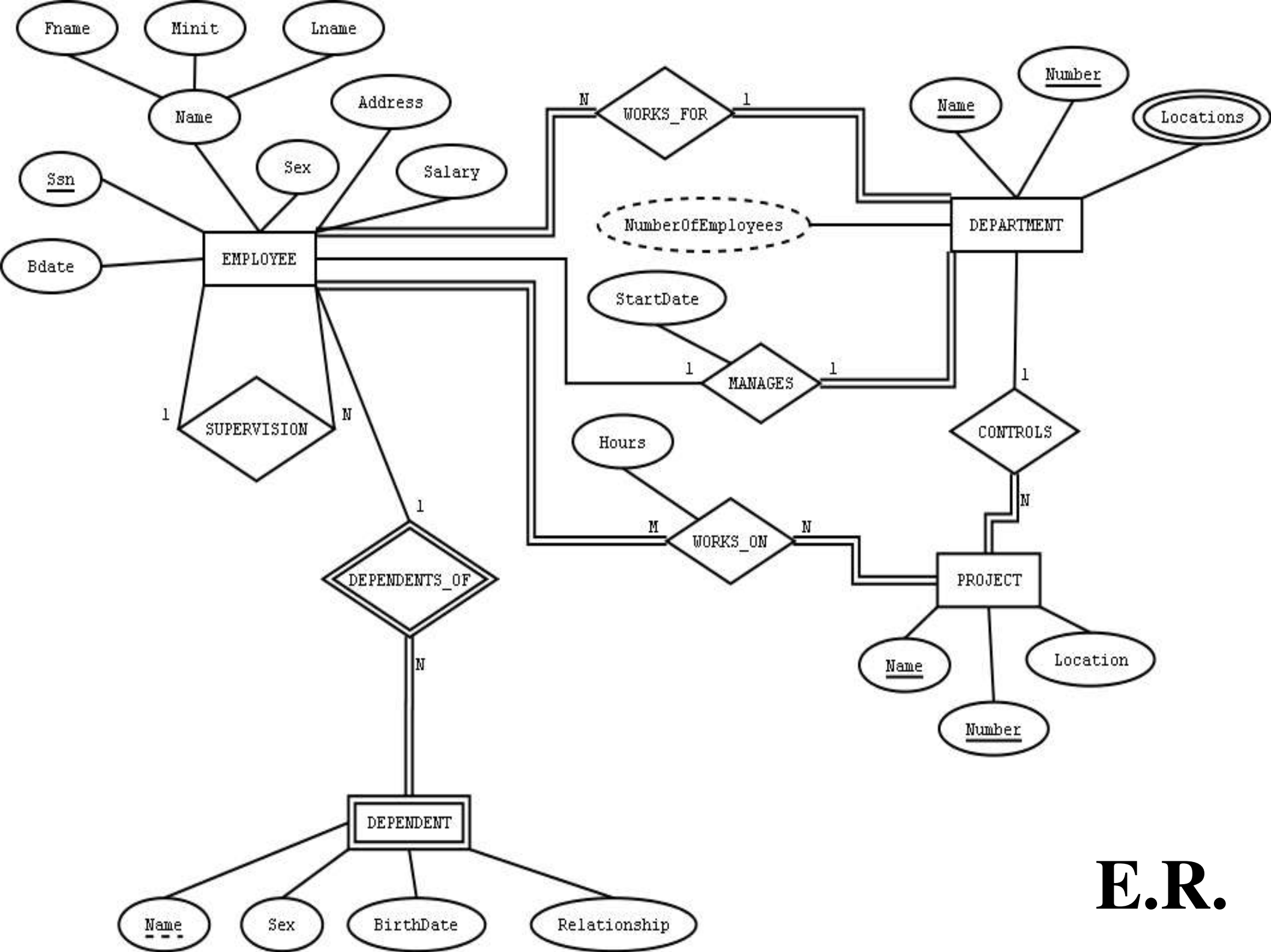
**** Acesso as Paginas JSP do PrevFogo ****
(Controle de Acesso)

A pagina&jsp representa qualquer pagina com controle de acesso

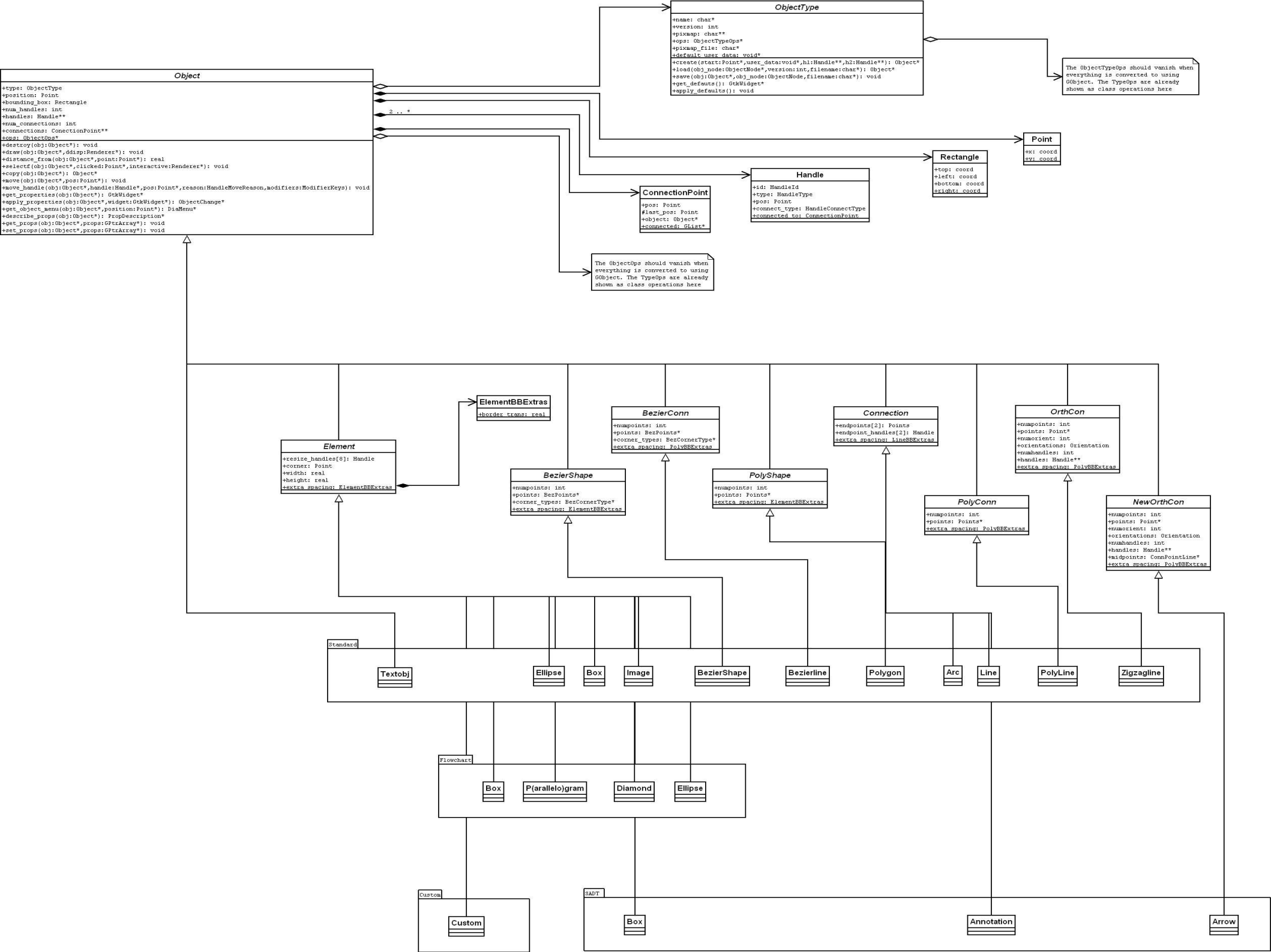


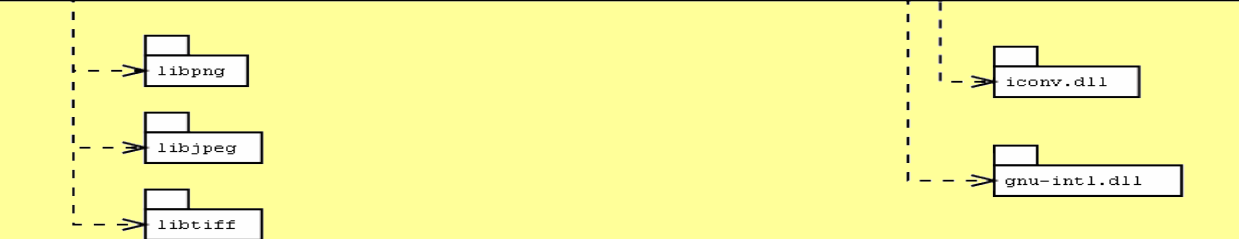
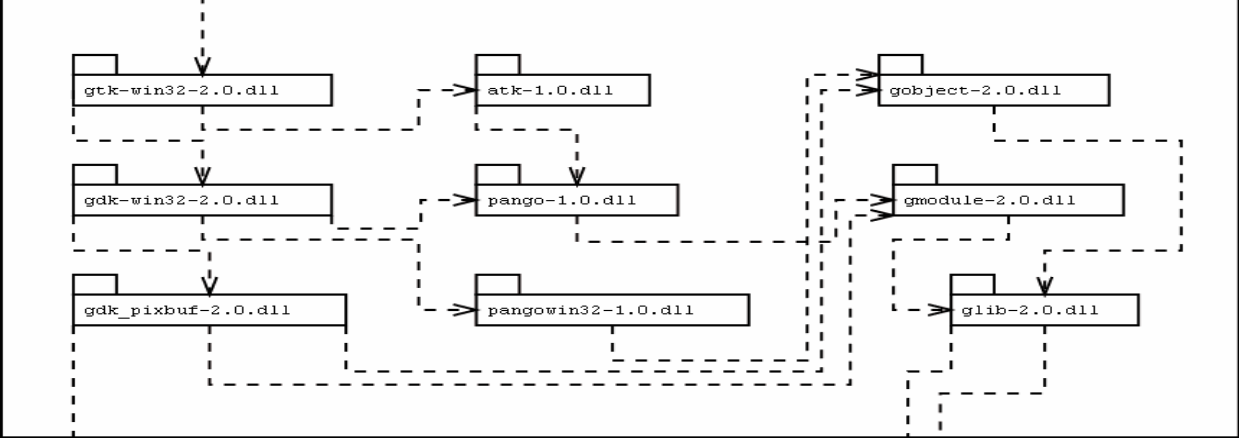
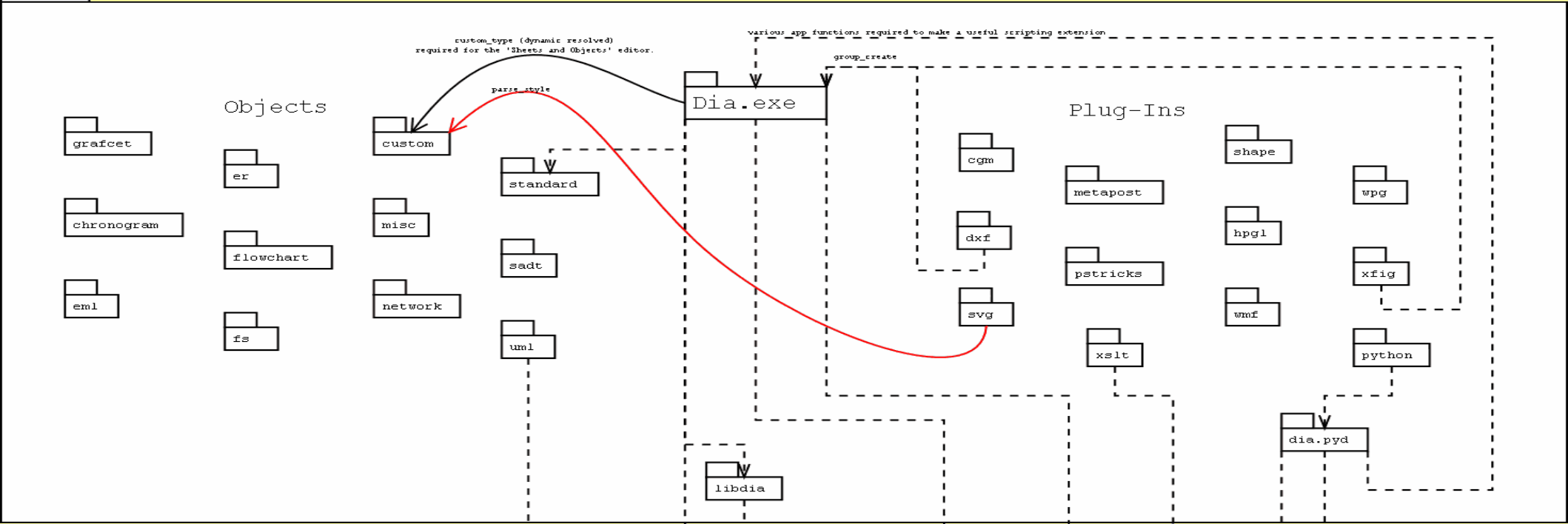






E.R.





Every Object and Plug-In is linked with libdia.
 Except 'standard' they are all not essential and can be excluded from dynamic loading.

Only uncommon dependencies are shown here. By design broken dependencies are drawn in red.

Nowadays no plug-in should need to link with gtk directly. There is an object property abstraction called StdProp available to avoid it. Direct manipulation of XML should be avoided as well. Use the data<_add>_* functions from libdia instead.



Instalação do Dia.

- **Download do Dia :**
<http://www.gnome.org/projects/dia/>
<http://dia-installer.sourceforge.net> (Para Windows)
- **Debian e outras distribuições com apt-get**
apt-get install dia
- **outras distribuições Linux.**
Deve ser feito o download e compilado.
- **Windows.**
Descompactar o dia-installer e executar o instalador.
- A maioria das distros Linux já vem com o pacote do Dia instalado
basta chama-lo com o comando **dia**



ArgoUML

ArgoUML é uma ferramenta CASE baseada na notação UML (Unified Modeling Language).

Foi desenvolvido pela comunidade de desenvolvedores de código livre Tigris vinculada a Universidade da Califórnia , Berkeley.

Sua interface é bem completa o que a torna um pouco complexa de manipular.



Do que ele é capaz ?

- Desenhar e imprimir diagramas UML.
- Gerar declarações de classes Java.
- Exportar documentação para páginas Web em Java.
- Gerar arquivos Gráficos (gif).
- Com auxílio de software de terceiros e possível gerar comandos SQL.
- Engenharia Reversa: fornece uma estrutura modular da engenharia reversa de classes Java.
- Exportar dados para o padrão XMI (baseado no formato XML)



Diagramas UML Suportados

- Diagrama de classe
- Diagrama de estados
- Diagrama de Atividades
- Diagrama de casos de uso
- Diagramas de colaboração
- Diagrama de Utilização/Componentes
- Diagrama de seqüência



Padrão UML Utilizado

- O Meta-Modelo Padrão de UML implementado é O UML 1.4 da OMG (Object management Group)
- Atualmente a OMG está em processo de especificação da UML 2.0.



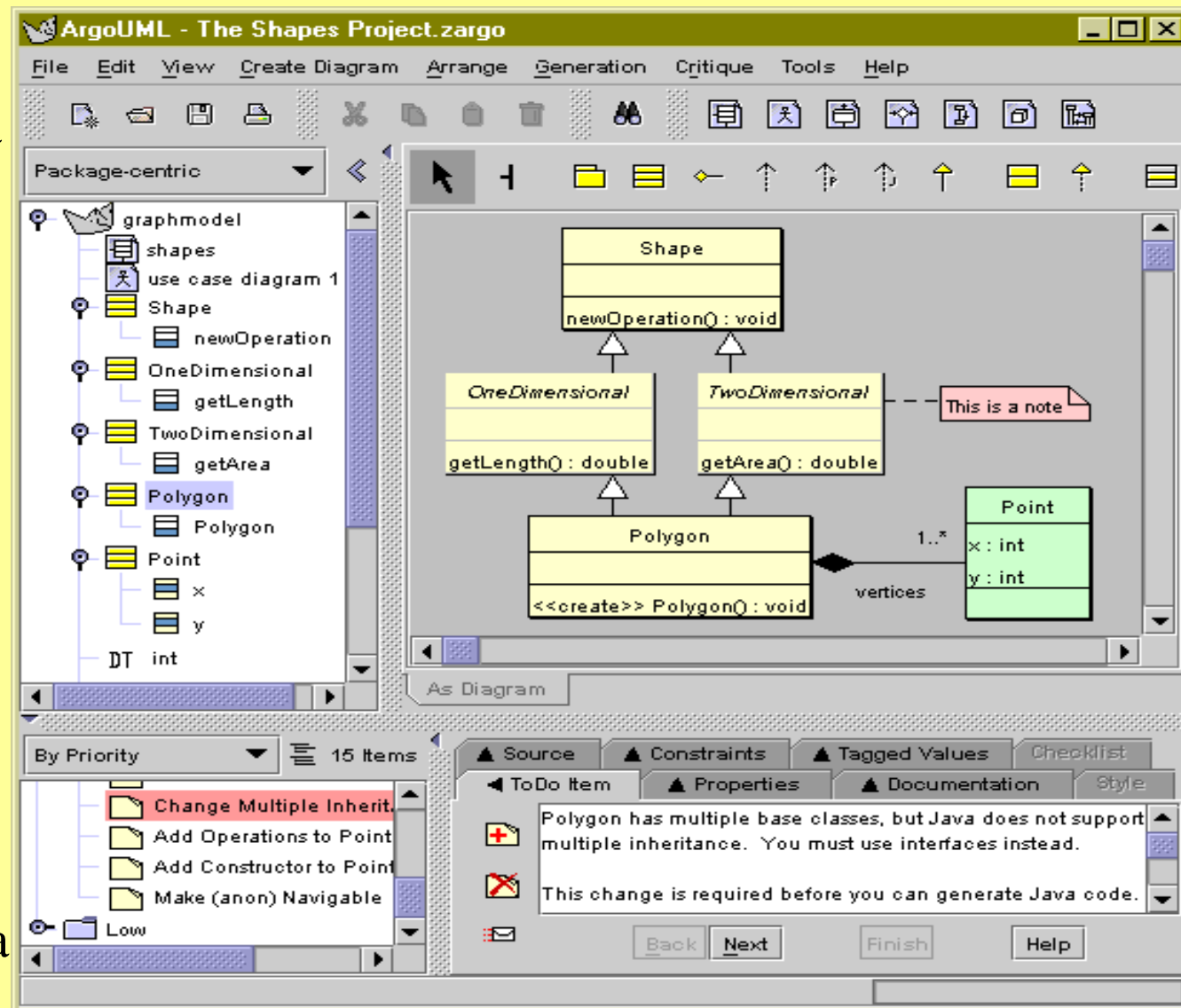


Requisitos para instalação

- Java 1.4 ou superior (JavaWebStart)
- ArgoUML é escrito inteiramente em Java e usa as classes padrões do Java . Isto permite que ArgoUML funcione virtualmente em toda a plataforma que suporte uma JVM (Java Virtual Machine).
- Internacionalização: inglês americano, inglês britânico, francês, alemão, espanhol, Russo, Norueguês, chinês.

ArgoUML

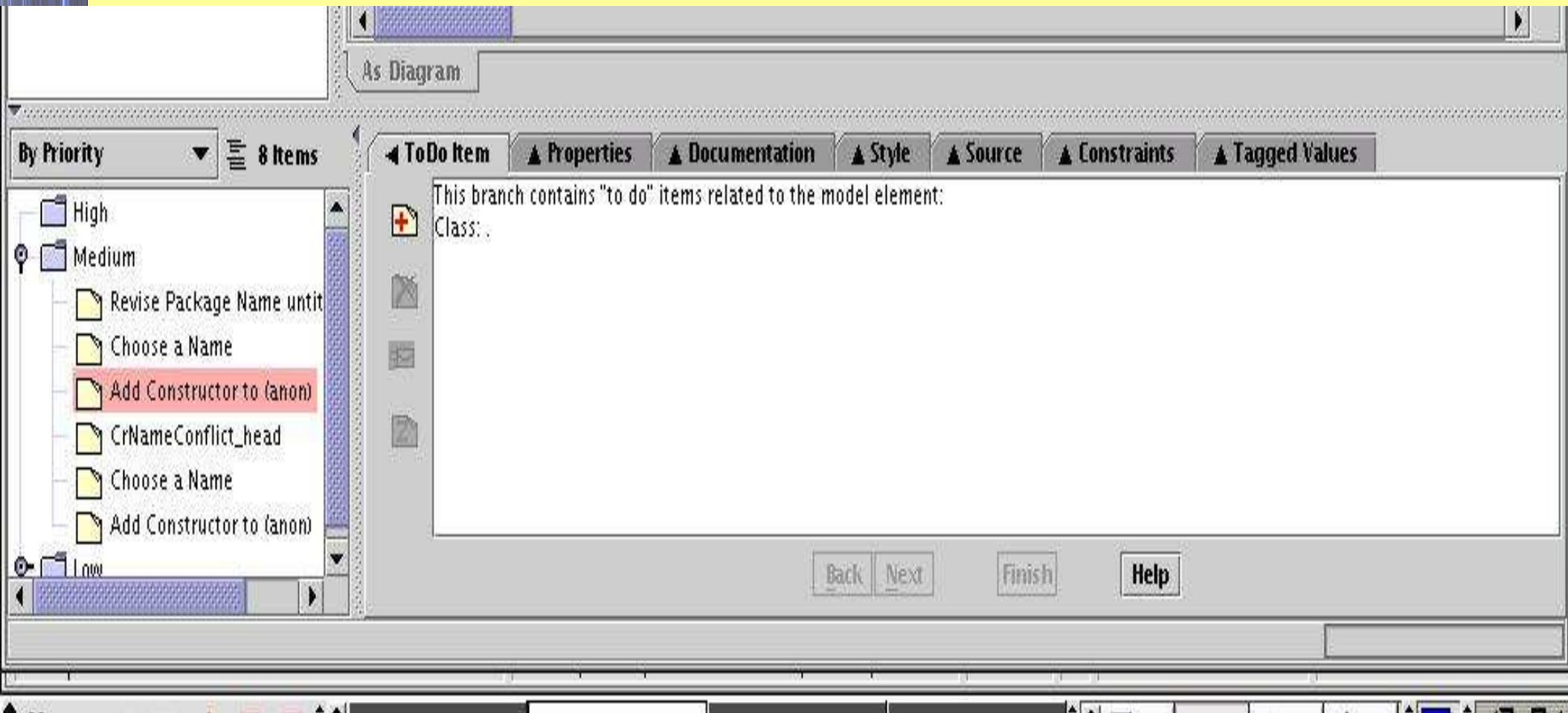
- A tela de ArgoUML é dividida em 4 painéis:
- **Esquerda superior :** uma vista hierárquica da do projeto atual.
- **Direita superior :** editor para a parte selecionada do projeto, neste caso um diagrama da classe.
- **Esquerda inferior :** desenhador de Lista **ToDo** (para fazer) .
- **Direita inferior :** detalhes do objeto selecionado no diagrama **ToDo** (Para fazer) .





Placa de Detalhes

- Permite que você edite detalhes do elemento selecionado do projeto.





ToDoItem

- O software interage um o usuário através de uma lista de procedimentos que ele deve executar, criando pendências para o termino da elaboração de um diagrama. Mostra a descrição do item selecionado, consistindo com 3 parágrafos curtos sobre prováveis problemas e etapas de como resolver. (Ex. Método Construtor)
- Diferencial de qualquer outra ferramenta do mercado.

The screenshot displays the 'ToDo Item' dialog box in the Ferramentas Livres para UML software. The dialog has a title bar with 'ToDo Item' and several tabs: 'Properties', 'Documentation', 'Style', 'Source', 'Constraints', and 'Tagged Values'. The 'Documentation' tab is active, showing a list of items with icons and text. The first item is a red plus sign icon followed by the text: 'You have not yet defined a constructor for class Municipio. Constructors initialize new instances such that their attributes have valid values. This class probably needs a constructor because not all of its attributes have initial values.' The second item is a red X icon followed by: 'Defining good constructors is key to establishing class invariants, and class invariants are a powerful aid in writing solid code.' The third item is an envelope icon followed by: 'To fix this, add a constructor manually by clicking on Municipio in the navigator pane and adding an operation using the context sensitive pop-up menu in the property tab, or select Municipio where it appears on a class diagram and use the "Add Operation" tool. Then use the Properties tab on the Detail Pane to select the <<create>> stereotype on the operation.' The fourth item is a document icon followed by: 'A constructor is an operation with the stereotype <<create>>.' At the bottom of the dialog, there are four buttons: 'Back', 'Next', 'Finish', and 'Help'. In the background, a navigator pane is visible with a tree view under 'Medium' containing items like 'CrNameConflict_head', 'Add Constructor to Municip', 'Choose a Name', etc.



Propriedades

- Mostra a propriedade do elemento selecionado do projeto, as propriedades variam de acordo com o elemento.
- propriedades como: nome da classe, visibilidade, generalizações, especializações, atributos, métodos e ainda alguns botões que auxiliam a excluir a classe ou criar novos atributos, métodos ou classes.

The screenshot shows the 'Properties' tab of a UML tool's dialog box for an 'Operation' element. The 'Name' field contains 'gravar'. The 'Stereotype' field is empty. The 'Owner' field shows 'projetoexemplo.Municipio'. The 'Visibility' section has radio buttons for 'public' (selected), 'protected', and 'private'. The 'Modifiers' section includes checkboxes for 'abstract', 'leaf', 'root', 'query', and 'static'. The 'Concurrency' section has radio buttons for 'sequential' (selected), 'guarded', and 'concurrent'. The 'Parameters' field contains 'return'. The 'Raised Signals' field contains 'none'. The dialog also features a 'ToDo Item' tab and several icons for actions like 'Add', 'Remove', and 'Cancel'.



Documentação

- Documentação do elementos selecionados : autor , versão e descritivo etc.

Documentation

Author: Marcio Junior Vieira

Version: V1.0

Since: 10/09/2003

Deprecated:

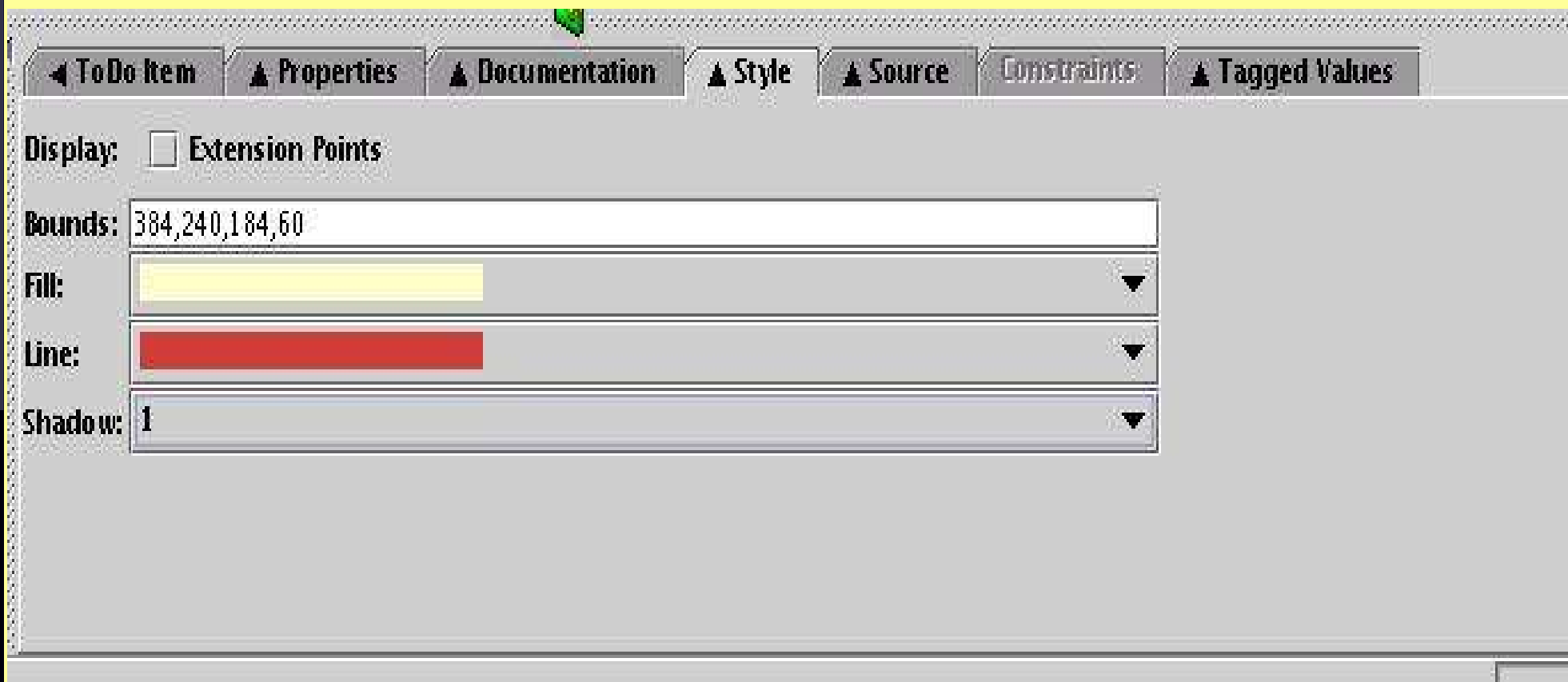
See:

Documentation: Somente o Super usuário terá o direito de eliminar dados do CEP.



Estilo


- Propriedades visuais do elementos selecionados no projeto (cor , tamanho, formas gráficas)





Fontes

- Visualização do código fonte Java que será gerado.
- A ferramenta não permite que seja alterado o código gerado, apenas após sua exportação.



```
public class Estado {  
  
    public String uf;  
    public Municipio myMunicipio;  
    public Municipio myMunicipio;  
    public Cep myCep;  
  
    public void gravar() { }  
    public void atualizar() { }  
    public void excluir() { }  
}
```



Chave-Valor (TaggedValues)

- Permite a entrada de uma chave no elemento e um valor correspondente , pode ser utilizado como histórico, ou discriminar dados úteis para o elemento.

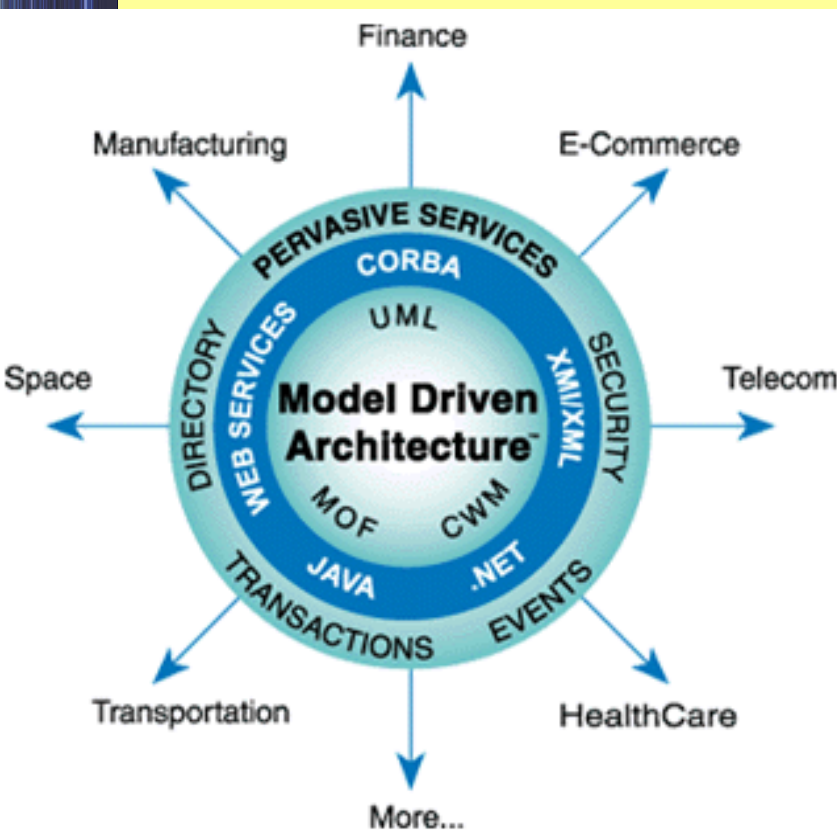
Tag	Value
homologado	NAD
desenvolvido	SIM
disponivel para instalacao	NAD



Integração com AndroMDA



- O AndroMDA é um framework open source baseado em MDA (Model Driven Architecture). Ele utiliza modelos UML gerados por ferramentas CASE (padrão XMI) e uma série de plugins, chamados de cartuchos (cartridges) para realizar a geração de componentes customizados, ou seja, o código-fonte do sistema.



Os componentes podem ser gerados para as plataformas padrão de mercado: J2EE (Java), .net (C#), PHP, HTML, etc. Aos plugins já disponíveis soma-se a capacidade de customização do AndroMDA. Basta escrever um novo plugin (ou modificar um já existente) para que a uma nova linguagem ou plataforma possa ser suportada.



Informações adicionais

- Atualmente a versão estável é a 0.20
- Suporte e compatibilidade ao AndroMDA.
- Os comandos “Undo” e “Redo” (desfazer e refazer) ainda não foram implementados na versão atual (apenas reverter para arq. salvo).
- O início do projeto foi no ano de 1998 na Universidade da Califórnia, Berkeley
- Licença BSD Open Source.
- Mais de 1.000.000 downloads do site da Tigris.
- Software comercial baseado (Poseidon for UML)

Poseidon UML (comercial)

Untitled - Poseidon for UML Professional Edition

File Edit View Create Diagram Align Generation Plug-Ins Help

Diagram Centric

Class Diagram_1

Class Diagram_1

Class Diagram_1

Location

-Name:String

<< Finder >>+All():Collection

User

-Number: Int

-FirstName: String

-LastName: String

Rates

Receipt

+member expenses *

TravelReport

+reservation

+travelReport 0..1

Reservation

-Number:String

-ReservationBegin Date :Date

-ReservationEnd Date:Date

<< Finder >>+Criteria(Vehicle Number:String):void

<< Finder >>+Member(memberNumber:String):void

+getNumber():String

+setNumber_ Number:String):void

Member

-Number: Int

-FirstName: String

-LastName: String

+reserving member

+member's account

+account holder

Account

-Number: Long

+generated bill

+billed account

Transaction

+transaction

+cleared pay

Bill

0..1

+bill

0..1

+vehicle location

+home location

<< EntityObject >>

Vehicle

-License Number:String

-Description:String

-Number: Int

-Vehicle Class:String

+Criteria(inout VehicleClass:String_Location:String):void

↑+reserved vehicle

Collaboration Diagrams

Deployment Diagrams

Sequence Diagrams

Sequence Diagrams

Birdview

Properties

Name Reservation

Namespace model 3

Visibility public protected package private

Modifiers abstract static final root active

<< Stereotypes ...

Owned Elements

accessor getNumber()

Attributes

-Number : String

-ReservationBeginDate : Date

-ReservationEnd(Date): Date

Operations

+Criteria(inout VehicleNumber:Strin

+Member(inout memberNumber:Str

Associations

placed reservation - reserving memb

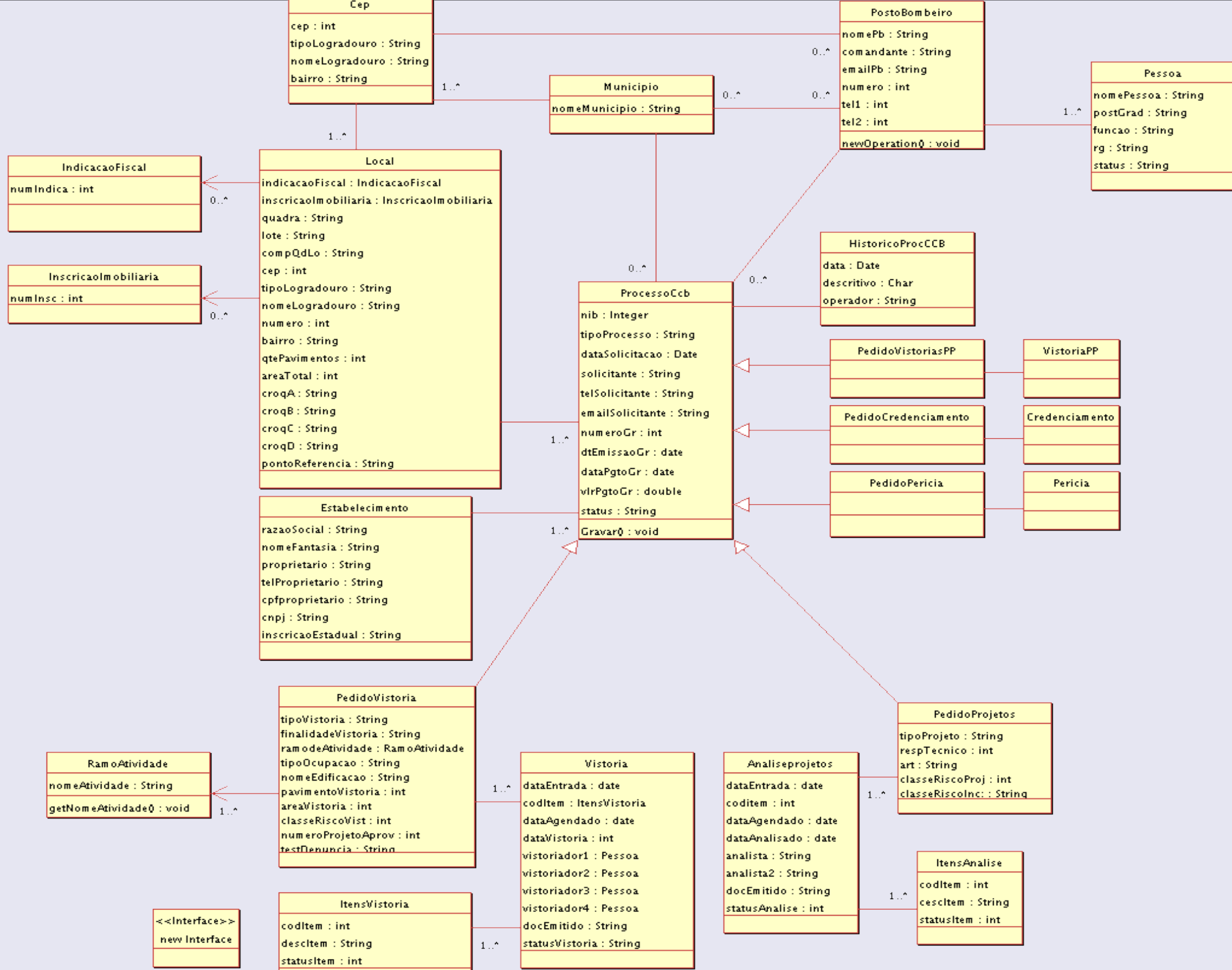
Reservation -> reserved vehicle

Reservation <-> TravelReport

↑ Extends

↑ Implements

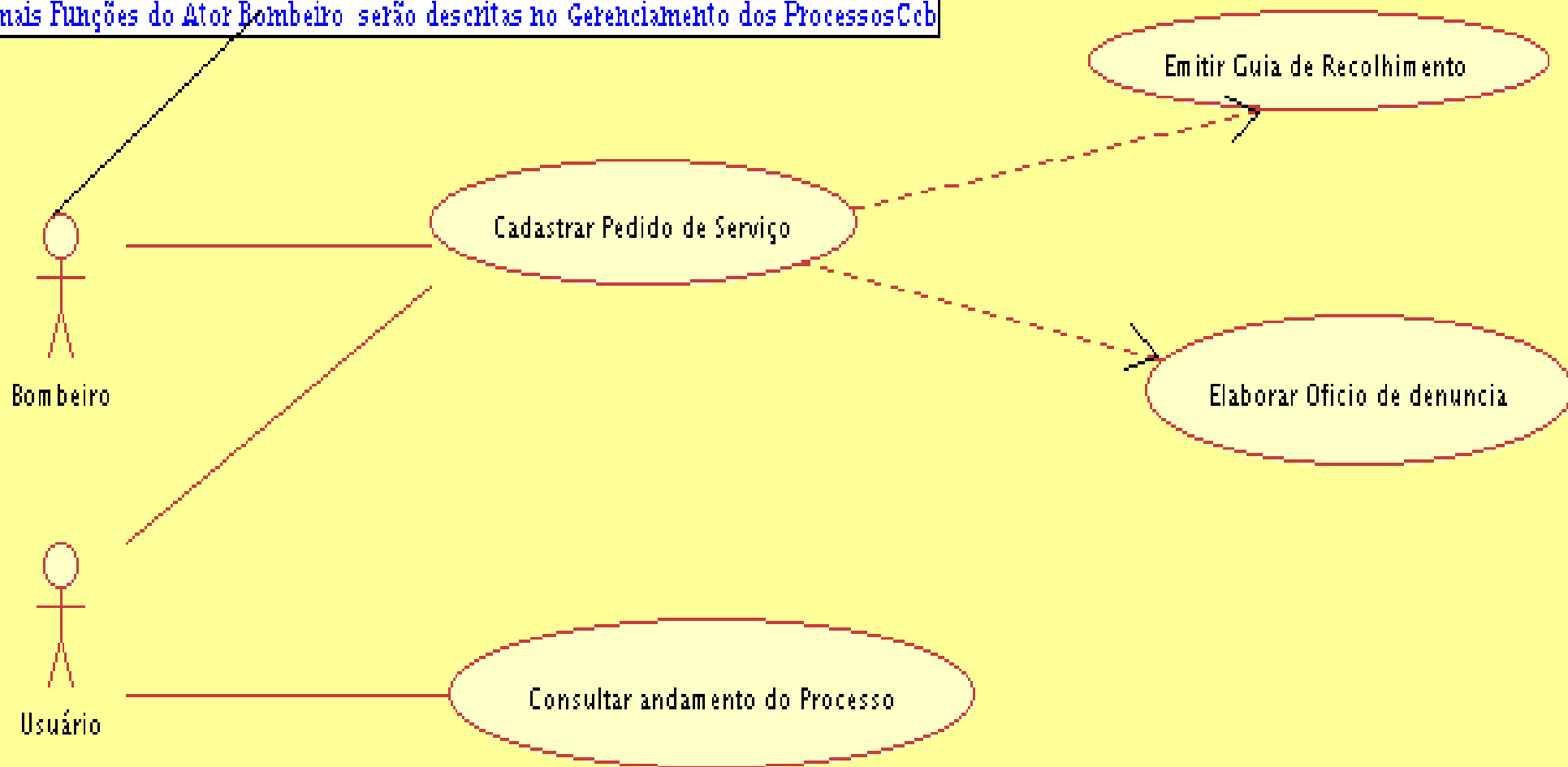
80%

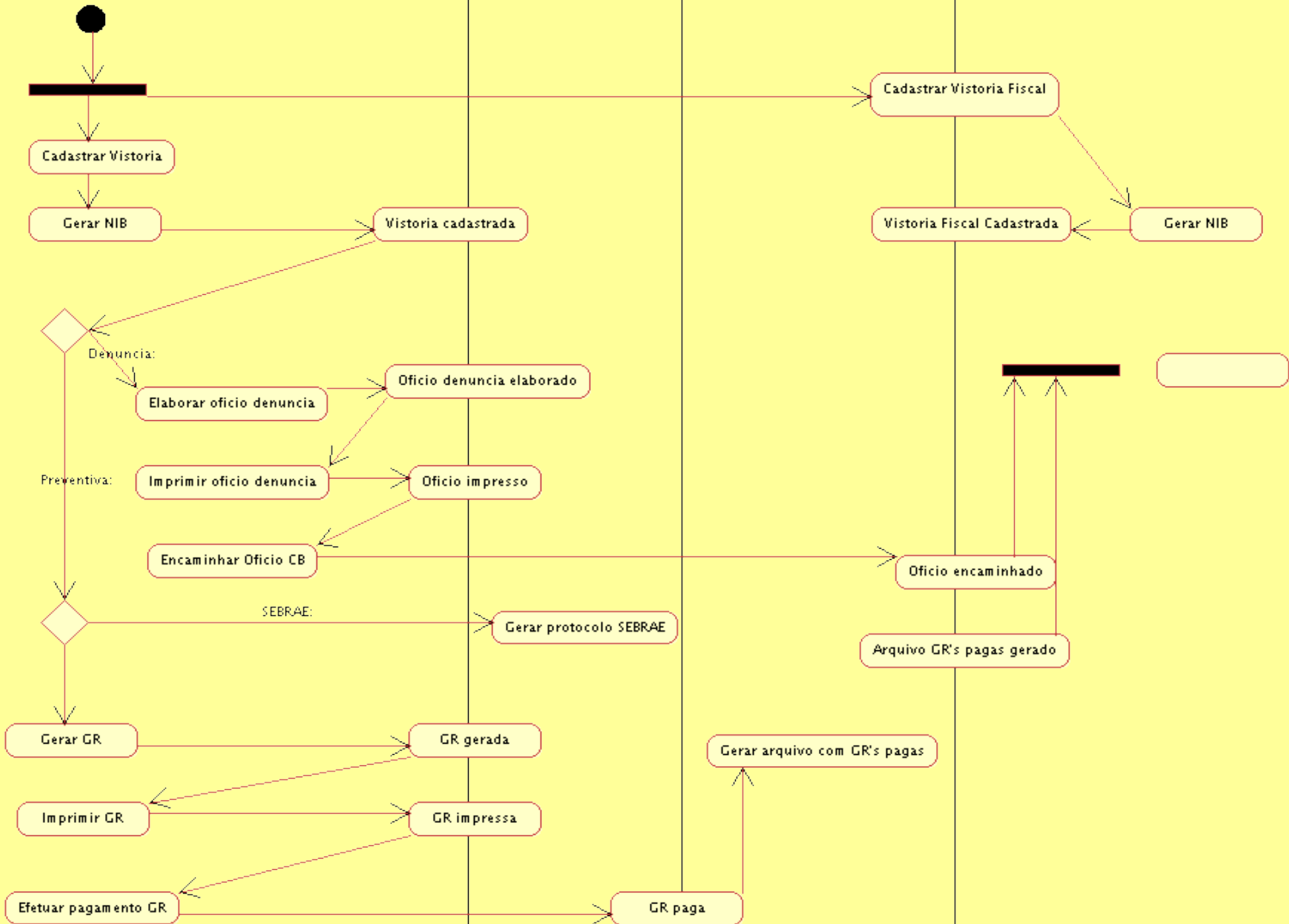


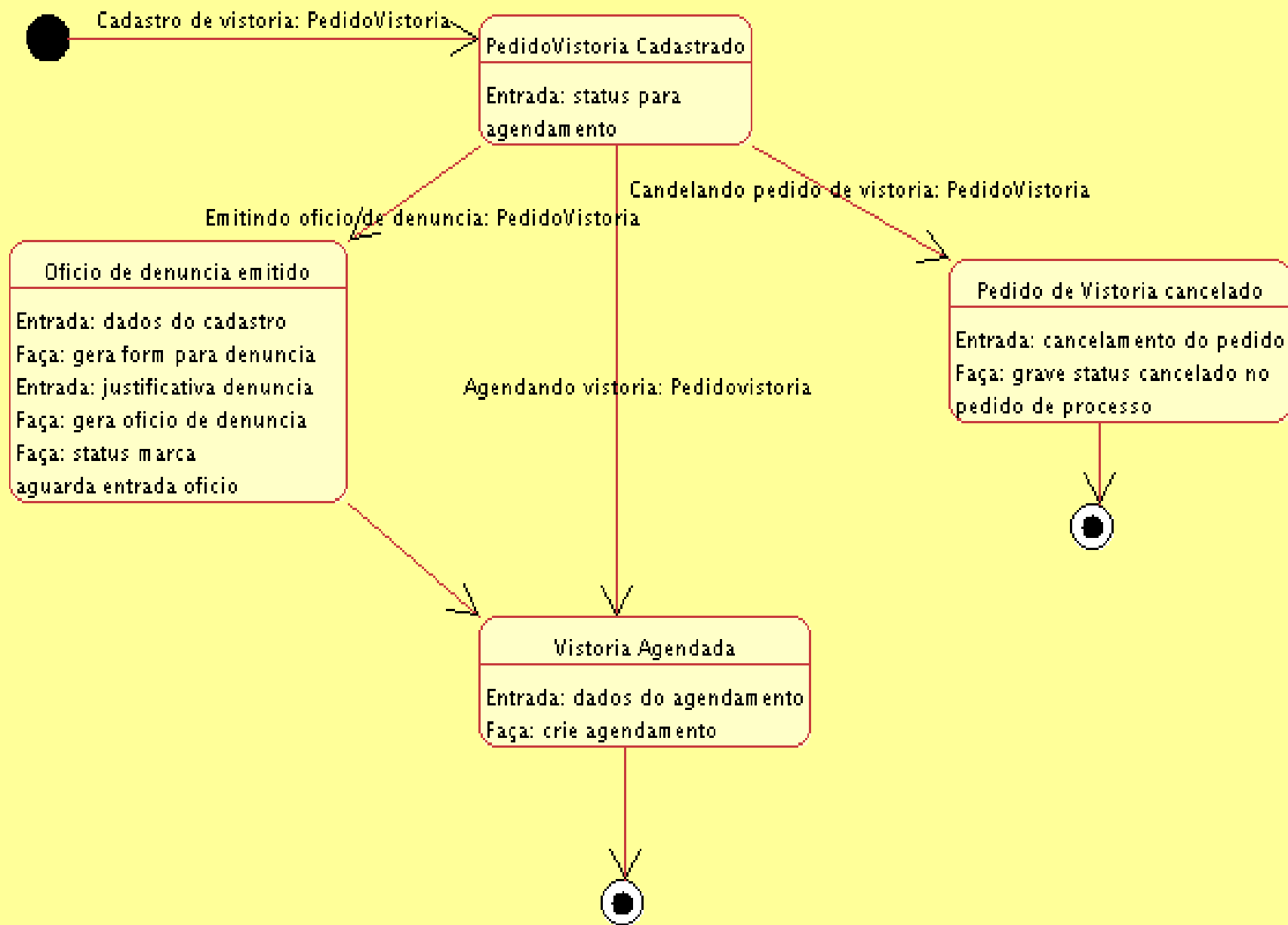


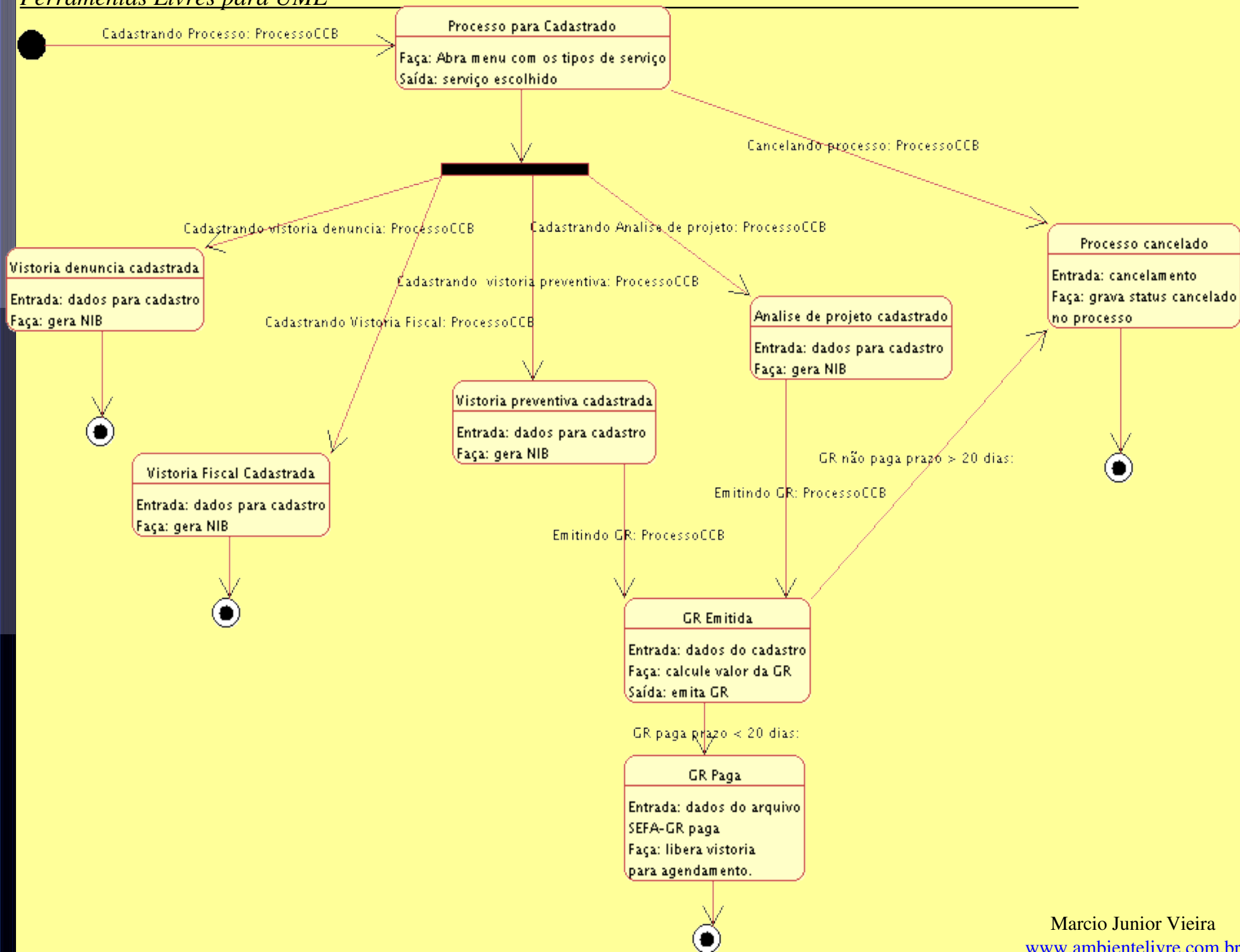
SOLICITAÇÃO DE SERVIÇO CCB

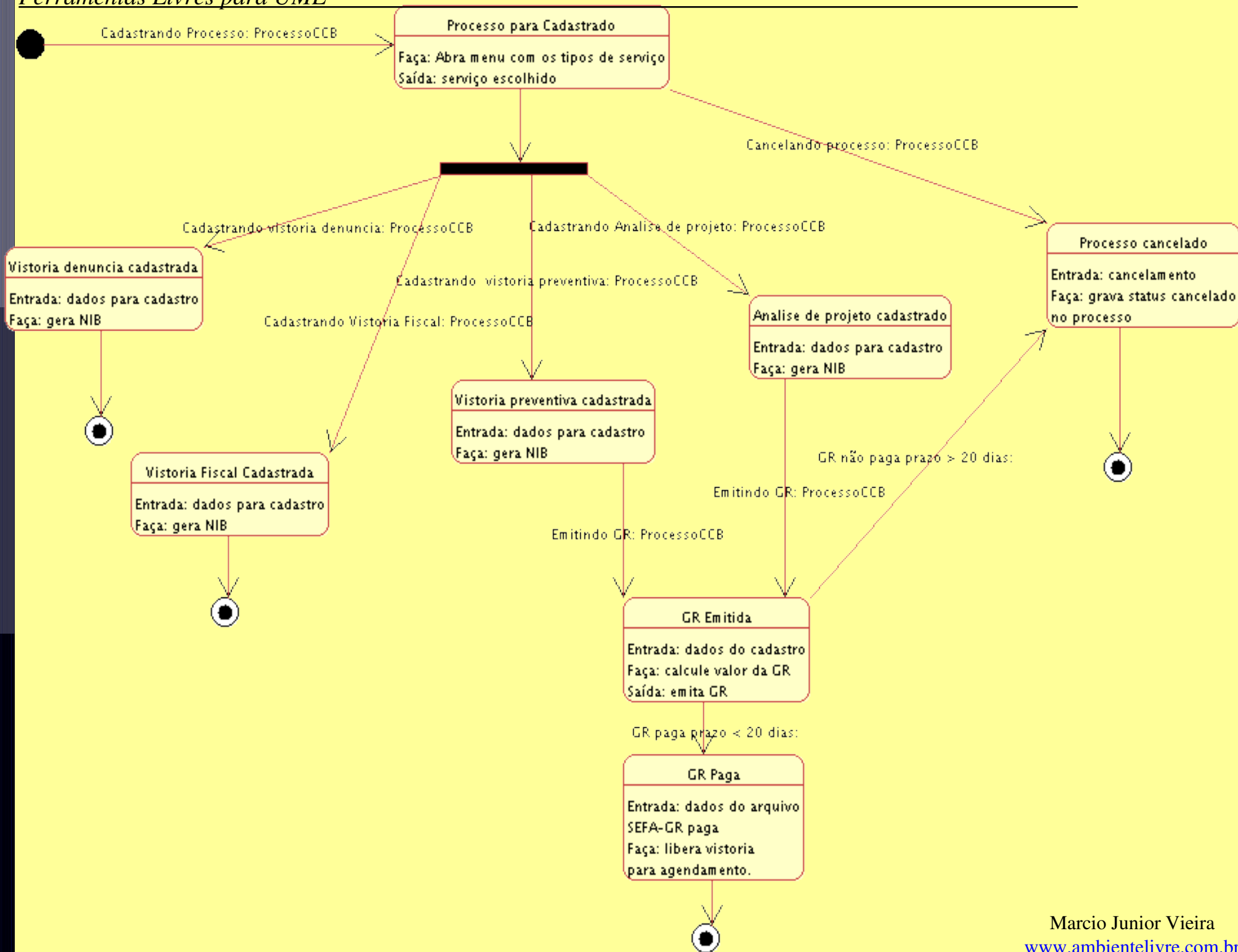
As Demais Funções do Ator Bombeiro serão descritas no Gerenciamento dos Processos Ccb













Instalação da ArgoUML.

- Download do ArgoUML. <http://argouml.tigris.org>
- **Download da JVM. : <http://www.java.sun.com>**
- Descompacte a ArgoUML no local de instalação (/usr/local/argo ou c:/Arquivos de Programas/argo).
- Crie um link com o seguinte comando:
- Unix, Linux FreeBSD
java -jar /usr/local/argo/argouml.jar
- Windows
java -jar c:/Arquivos de Programas/argo/argouml.jar
- Se o Java estiver instalado corretamente a ArgoUML estará pronto para a criação de diagramas.



Umbrello

Umbrello é um Software de Modelagem UML, que é integrado ao projeto KDE.

Este Software é utilizado para modelar o próprio projeto do KDE por a grande de seus desenvolvedores que utilizam UML.



Plataformas Suportadas

- Gnu/Linux.
- FreeBSD
- Solaris



Do que o Umbrello e Capaz

- Desenhar e imprimir diagramas UML.
- Gerar declarações de classes Java, PHP, javaScript, ActionScript, C++, SQL, Ada, IDL, XMLSchema, Python, Perl e Ruby.
- Gerar arquivos Gráficos (png).
- Engenharia Reversa de Classes .
- Arquivos no padrão XMI (baseado no formato XML)
- Wizard para Criação de Classes e geração de Código Fonte e configuração de cabeçalhos do Código de Fonte.
- Visualizador de Código Fonte.
- Refatoração (refactoring).



Diagramas suportados

- Diagramas de classe.
- Diagrama de seqüência.
- Diagramas de Colaboração.
- Diagramas de caso de Uso.
- Diagrama de Estado.
- Diagrama de Atividade.
- Diagrama de componentes.
- Diagrama de Utilização.



Suporte para criação de diagramas.

- Multiplicidade.
- Atributos.
- Operações.
- Modelos.
- Associações.
- Generalização.
- Dependência.
- Composição.
- Agregação.
- Refatoração.



O Umbrello e sua utilização!

- O Umbrello atualmente é recomendado para pequenos projetos.
- Diversos desenvolvedores de Softwares que trabalham no projeto KDE utilizam o Umbrello para Modelagem UML.
- Seus desenvolvedores vêm trabalhando para que o software tenha um suporte de desenvolvimentos para médios e grandes projetos.
- Alguns Usuário :
- *Projeto Quanta* : <http://quanta.kdewebdev.org/>
- *GnuMed* - <http://www.gnumed.org/>
- *OMG* - <http://www.omg.org/>
- *OpenCity* - <http://www.opencity.info/>
- Mais usuários em www.umbrello.org (users)



Umbrello UML Modeller

VisualizacaoUmbrello.xmi [modificado] - Modelador UML Umbrello

Arquivo Editar Diagrama Código Configurações Ajuda

Novo Abrir Salvar Imprimir Desfazer Refazer Cortar Copiar Colar Índice Deslizante do Zoom Zoom para 100%

Componentes

Diagramas UML

Visões

- Visão Lógica
 - diagrama de classe
 - Tipos de dados
- Visão Caso de Uso
 - Sistema Integrado
 - Analista Comercial
 - Sistema de Contabilidade
 - Vendedor
 - gerente Comercial
 - Analisar Risco
 - Atualizar Contas
 - Avaliar Negocio
 - Estabelecer Limites
 - Fechar Preço
 - Negocios c/ Limites exce...
 - Registrar Negocio

Documentação

Negocios com limites excedidos devem ter tratamento especial.

Documentação

```
graph TD
    G[gerente Comercial] --- E([Estabelecer Limites])
    A[Analista Comercial] --- AR([Analisar Risco])
    A --- FP([Fechar Preço])
    A --- RN([Registrar Negocio])
    SC[Sistema de Contabilidade] --- AC([Atualizar Contas])
    V[Vendedor] --- RN
    AR -.->|<<include>>| AN([Avaliar Negocio])
    FP -.->|<<include>>| AN
    RN --- NL([Negocios c/ Limites excedidos])
```

Visão Hierárquica

Visão dos Diagramas



Visões dos diagramas UML.

Visão Lógica

- Diagramas de Classes.
- Diagramas de Colaboração.
- Diagramas de Estado.
- Diagramas de Atividades.
- Diagramas de Seqüência.

Visão Caso de Uso.

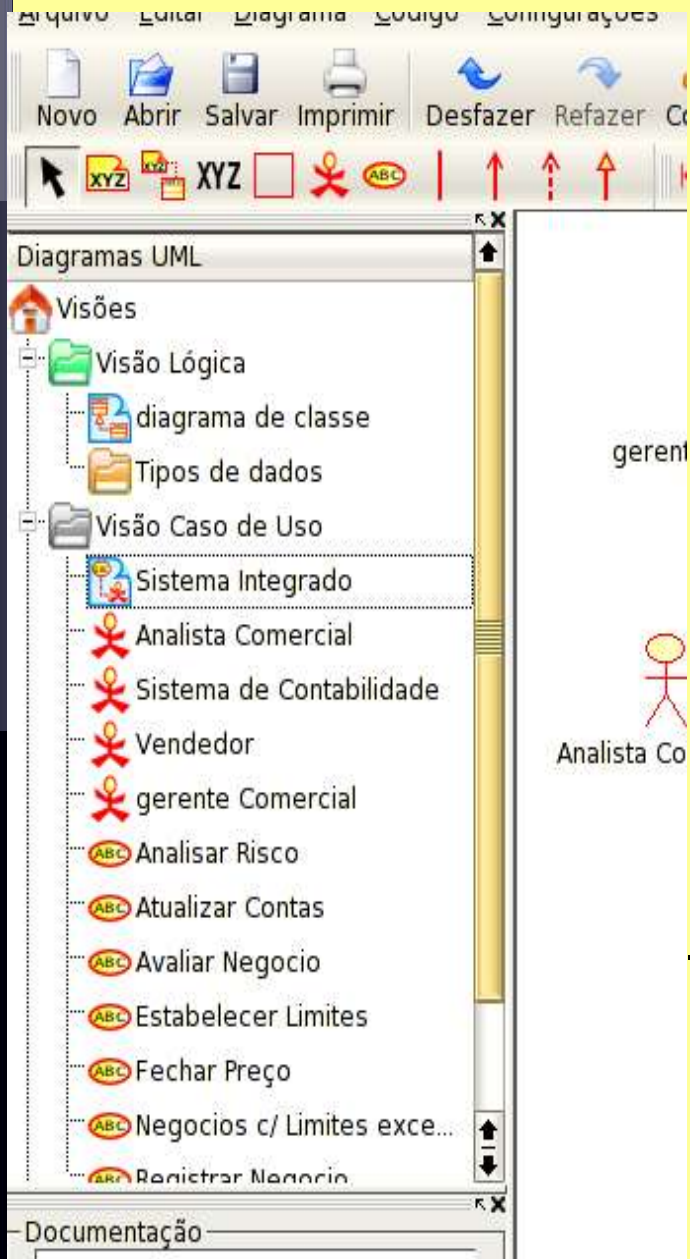
- Diagramas de Caso de Uso.

Visão do componente.

- Diagramas de Componentes.

Visão de Distribuição.

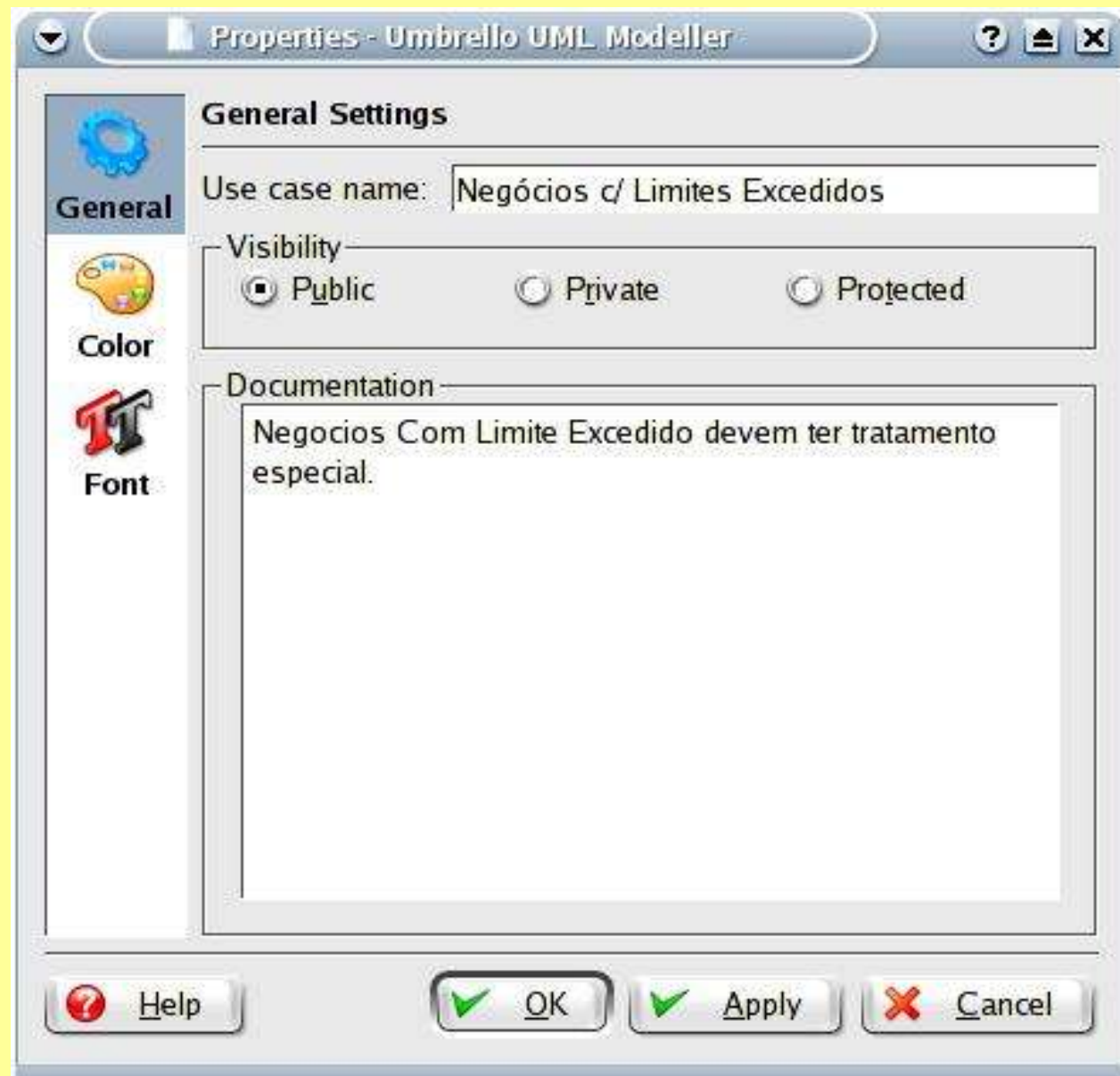
- Diagramas de Distribuição.





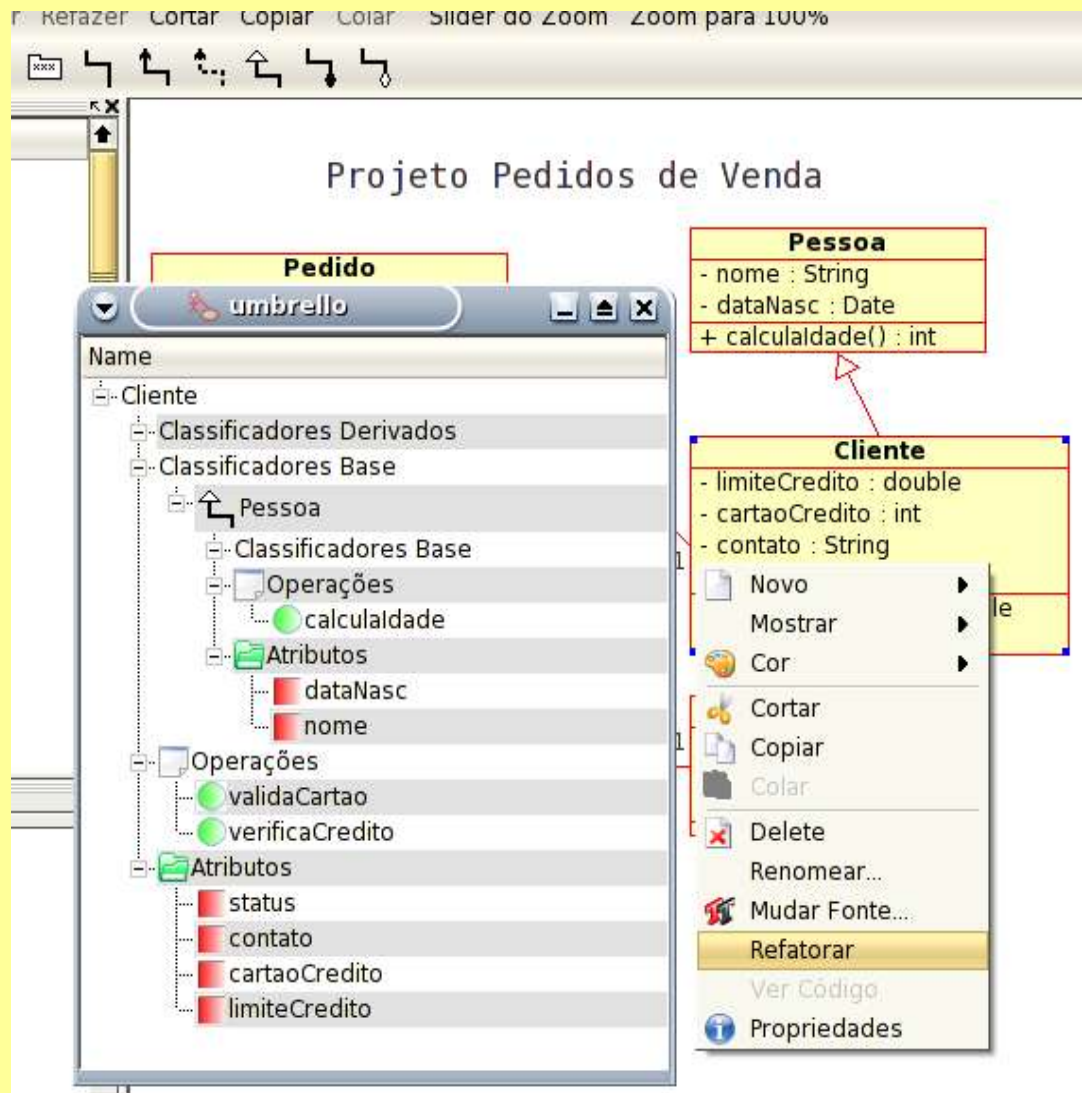
Propriedades - Caso de Uso

- Nomes
- Visibilidade Obj.
- Documentação
- Cores
- Fontes





Refactoring

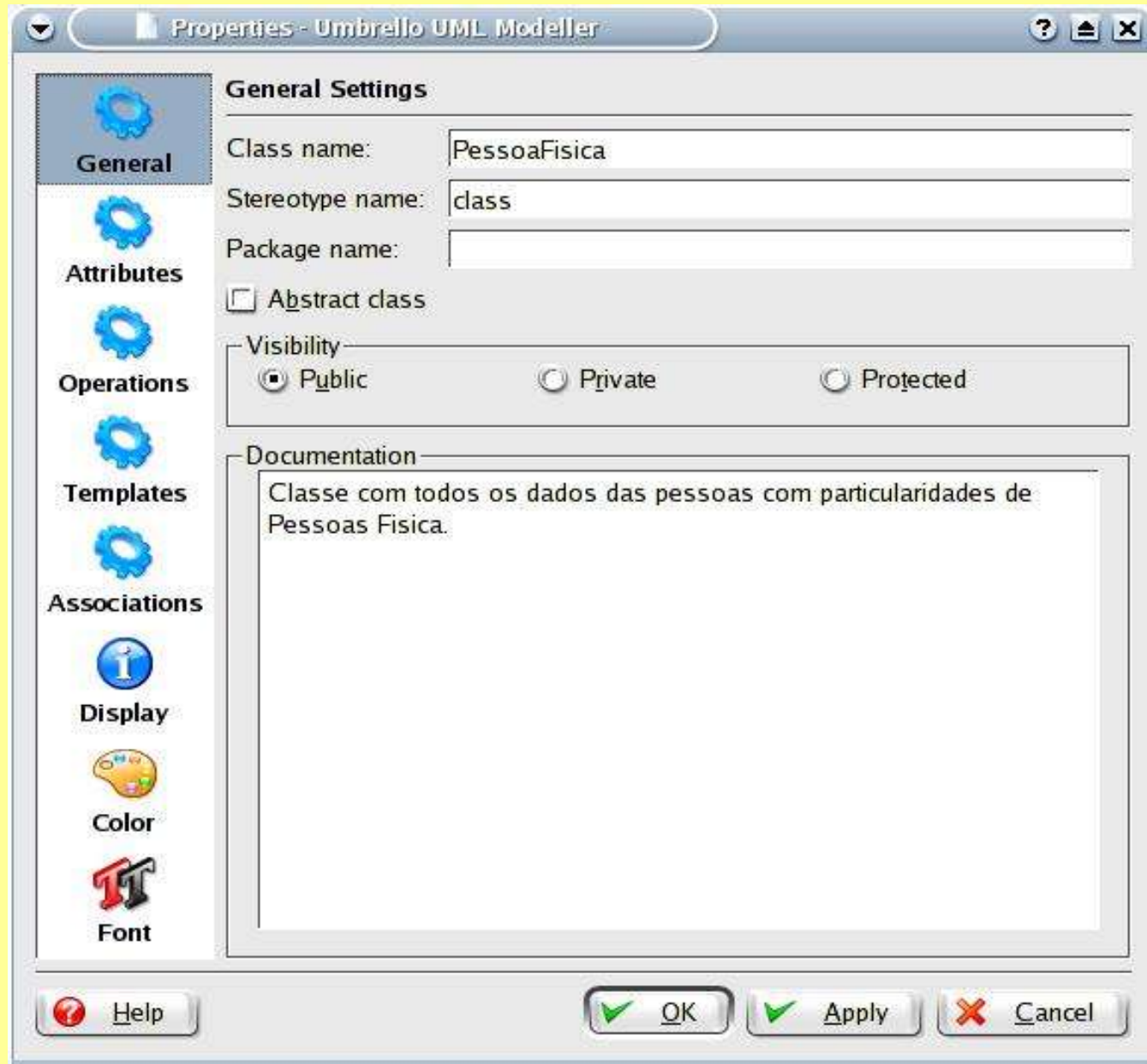


- Segundo Martin Fowler significa: “Mudança interna de estrutura de um software, que o torna mais fácil de se entender e mais barato de modificar, sem causar alterações em seu comportamento observável”.



Propriedades - Classes

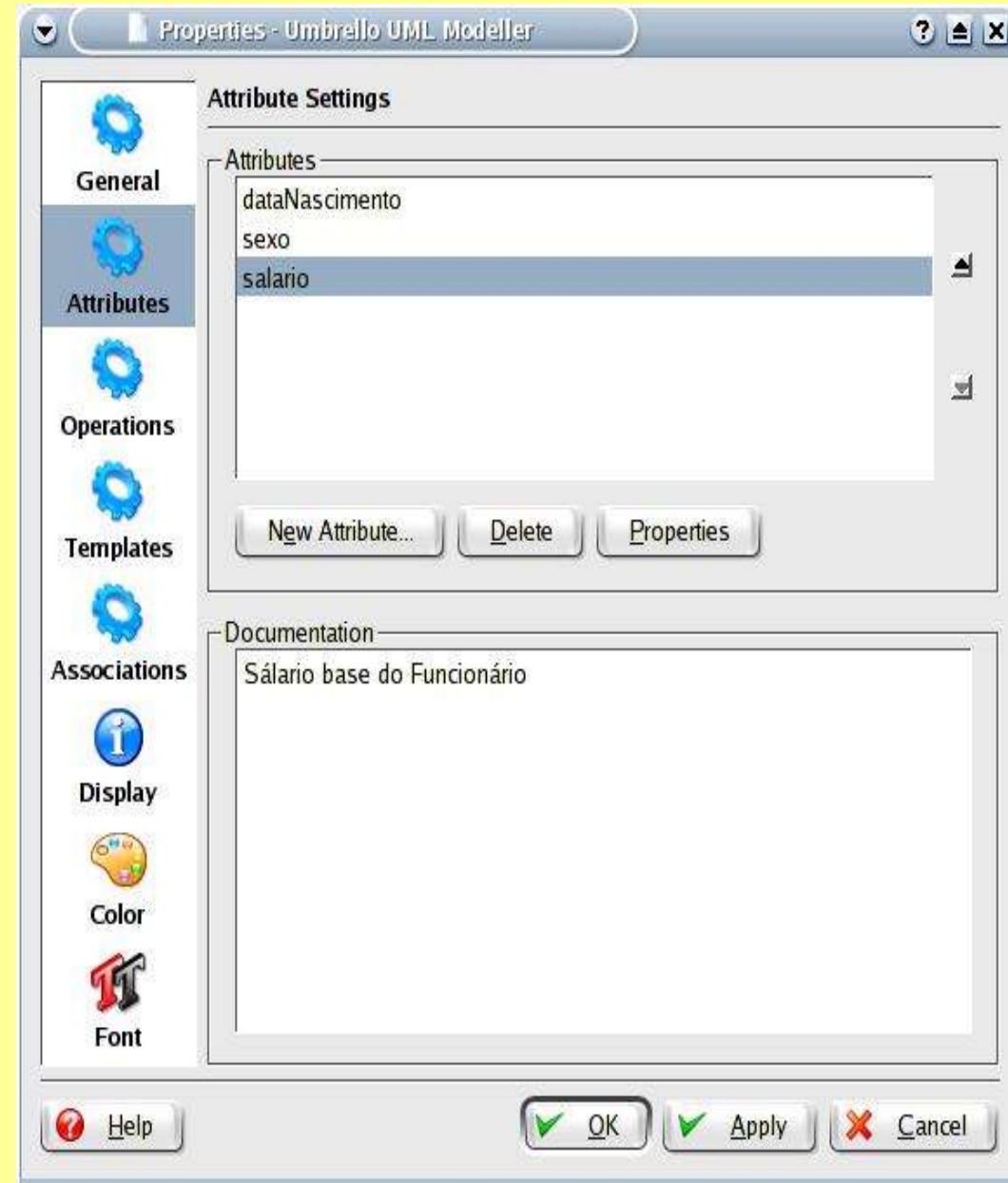
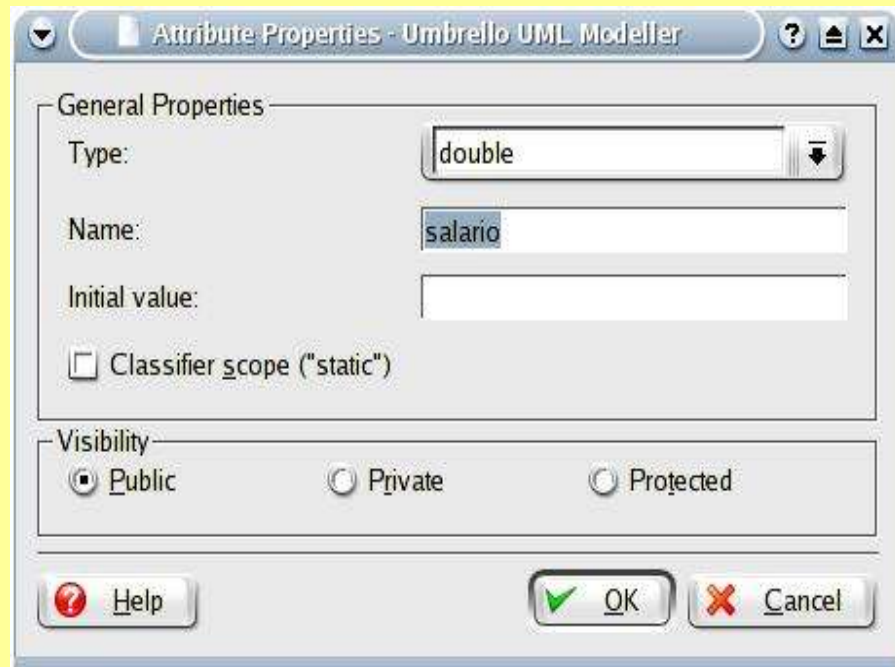
- Atributos
- Operações
- Associações
- Templates
- Cores
- Fontes





Propriedades - Atributos de Classes

- Documentação
- Propriedades





Visualizador de Fontes

The screenshot shows the "Visualizador de Código" window in Umbrello. The window title is "Projeto-LightCMS.xmi [modificado] - Modelador UML Umbrello". The window has two checked checkboxes: "Mostrar tipo de bloco" and "Mostrar blocos ocultos". A "Fechar" button is in the top right. The code editor shows the source code for "Module.java".

```
Module.java

import DataBase;
import int;
// Class Module
//
public class Module {
// Fields
//
private int codeModule;
// Methods
// Constructors
// Empty Constructor
public Module ( ) { }
// Accessor Methods
/**
 * Get the value of codeModule
 *
 * @return the value of codeModule
 */
private int getCodeModule ( ) {
    return codeModule;
}
/**
 * Set the value of codeModule
 *
 *
 */
private void setCodeModule ( int value ) {
    codeModule = value;
}
// Operations
}
```

Código-fonte Java gerado e Headings

```
/******
```

```
    Pedido.java - Copyleft marcio
```

```
GNU Public License .... Aqui você pode escrever a licença para seu código, comentários e outras informações para isso  
basta configurar os "headings"
```

```
os arquivos estão alocados em /usr/lib/kde3/share/apps/umbrello/headings/heading.java
```

```
Este arquivo foi gerado : Sáb Jan 29 2005 as 13:29:43
```

```
*****/
```

```
import java.util.Date;
```

```
import java.lang.String;
```

```
import Boolean;
```

```
public class Pedido {
```

```
    // Fields
```

```
    private int id_pedido;
```

```
    private java.util.Date dataPedido;
```

```
    private java.lang.String observacoes;
```

```
    // Methods
```

```
    // Constructors
```

```
    // Accessor Methods
```

```
    /**
```

```
     * Get the value of id_pedido
```

```
     *
```

```
     * @return the value of id_pedido
```

```
     */
```

```
    private int getId_pedido ( ) {
```

```
        return id_pedido;
```

```
    }
```

```
    // Operations
```

```
    public void encerrarPedido ( ) {
```

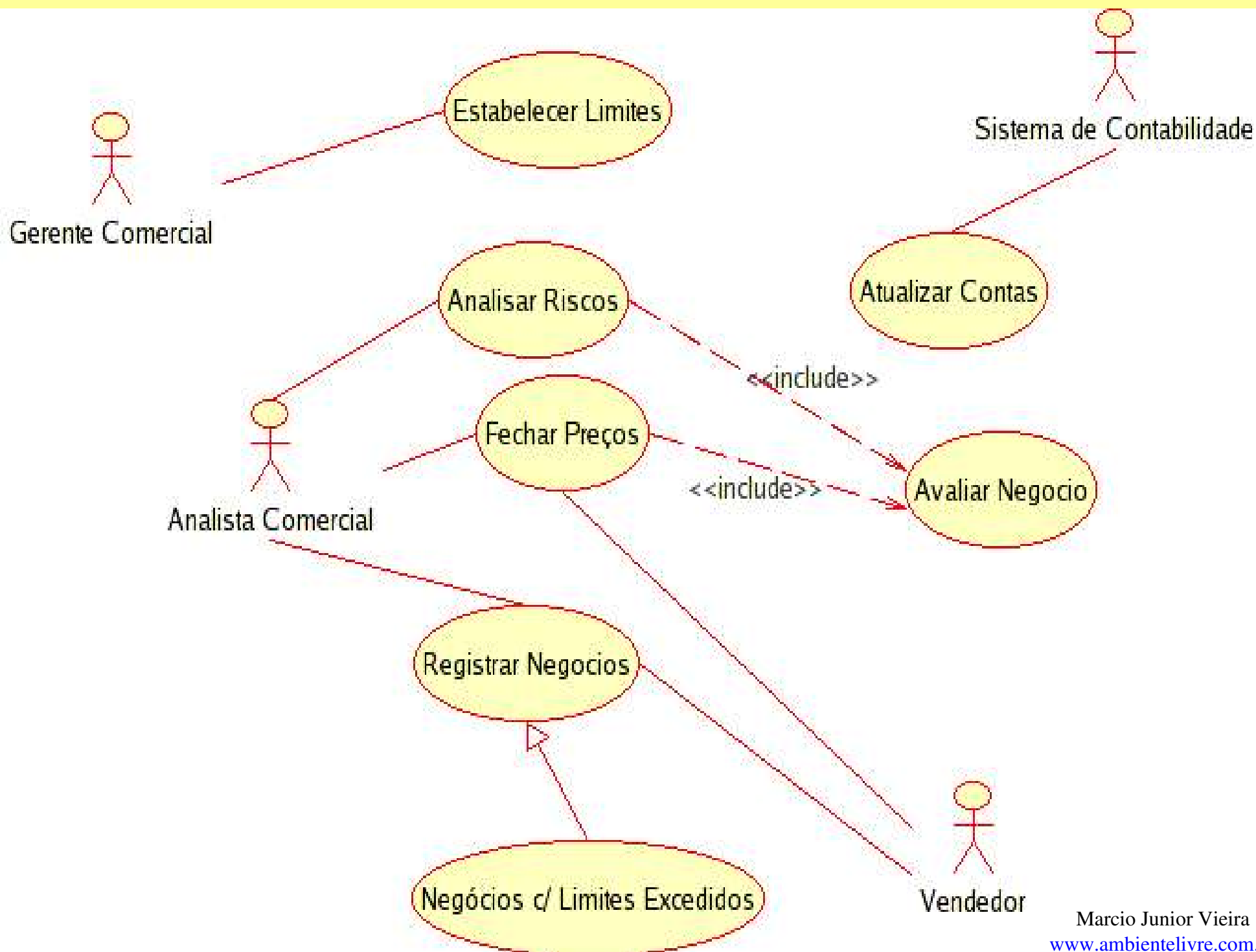
```
    }
```

```
}
```

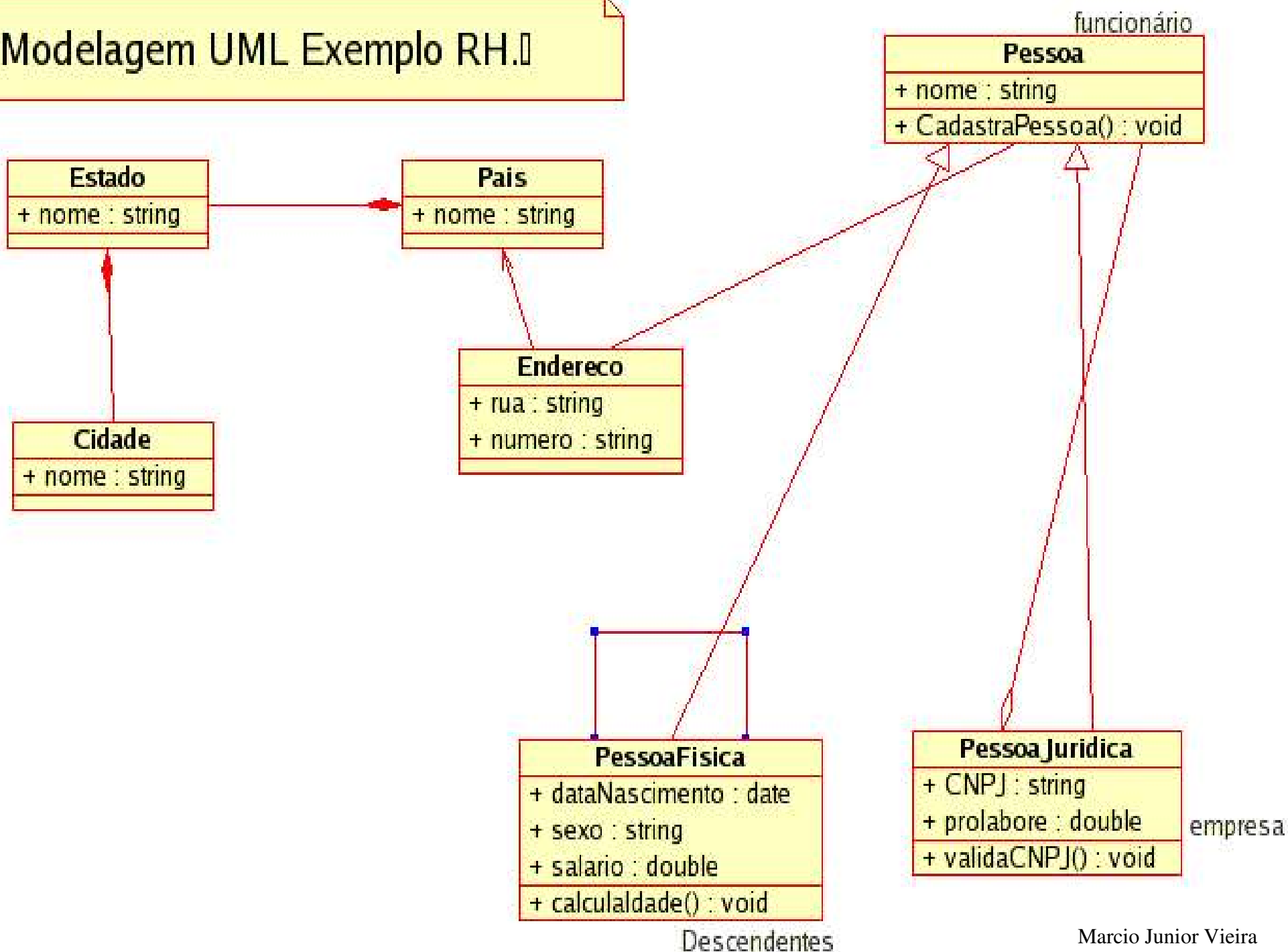


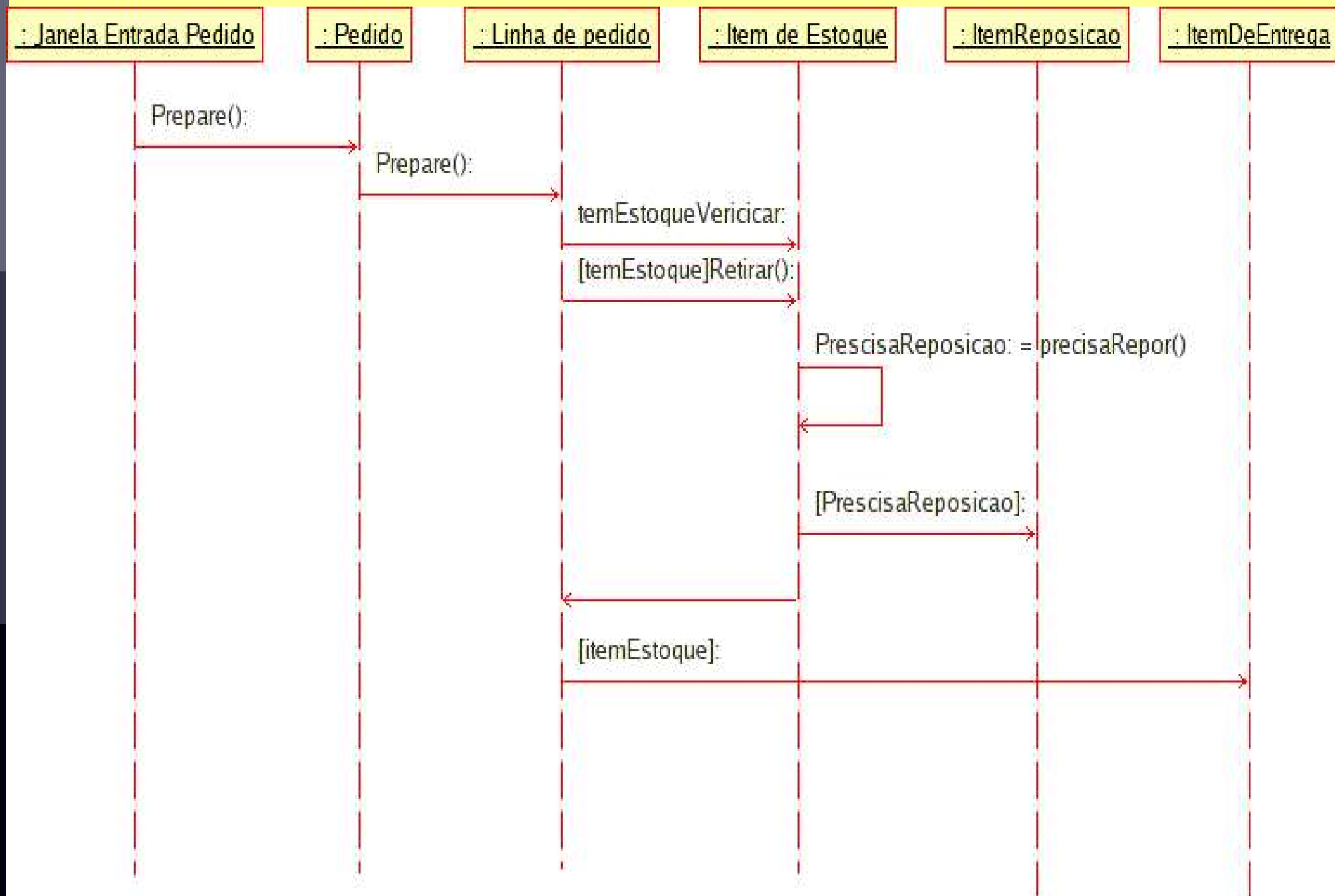
SQL gerado a partir de uma classe

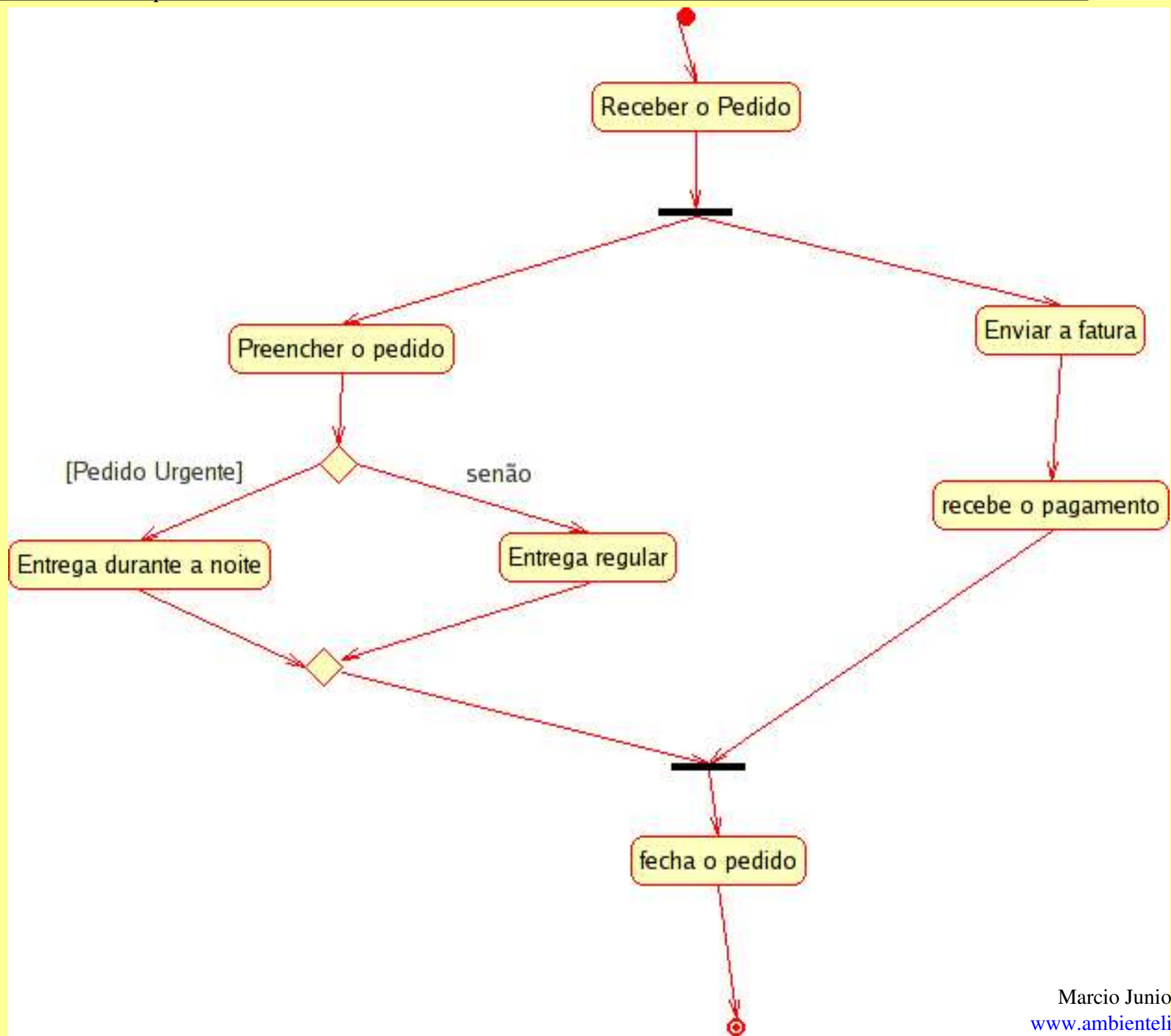
```
--          pedido - Copyleft marcio
--
-- GNU Public License .... Aqui você pode escrever a licença para seu código, comentários e
outras
-- informações para isso basta configurar os "headings"
--
-- os arquivos estão alocados em /usr/lib/kde3/share/apps/umbrello/headings/heading.sql
--
-- Este arquivo foi gerado : Sáb Jan 29 2005 as 13:33:43
--
-- TABLE: Pedido
--
CREATE TABLE Pedido (
    id_pedido int ,
    dataPedido Date ,
    vendedor Char(50) ,
    status Char(1) ,
    observacoes Text
);
```

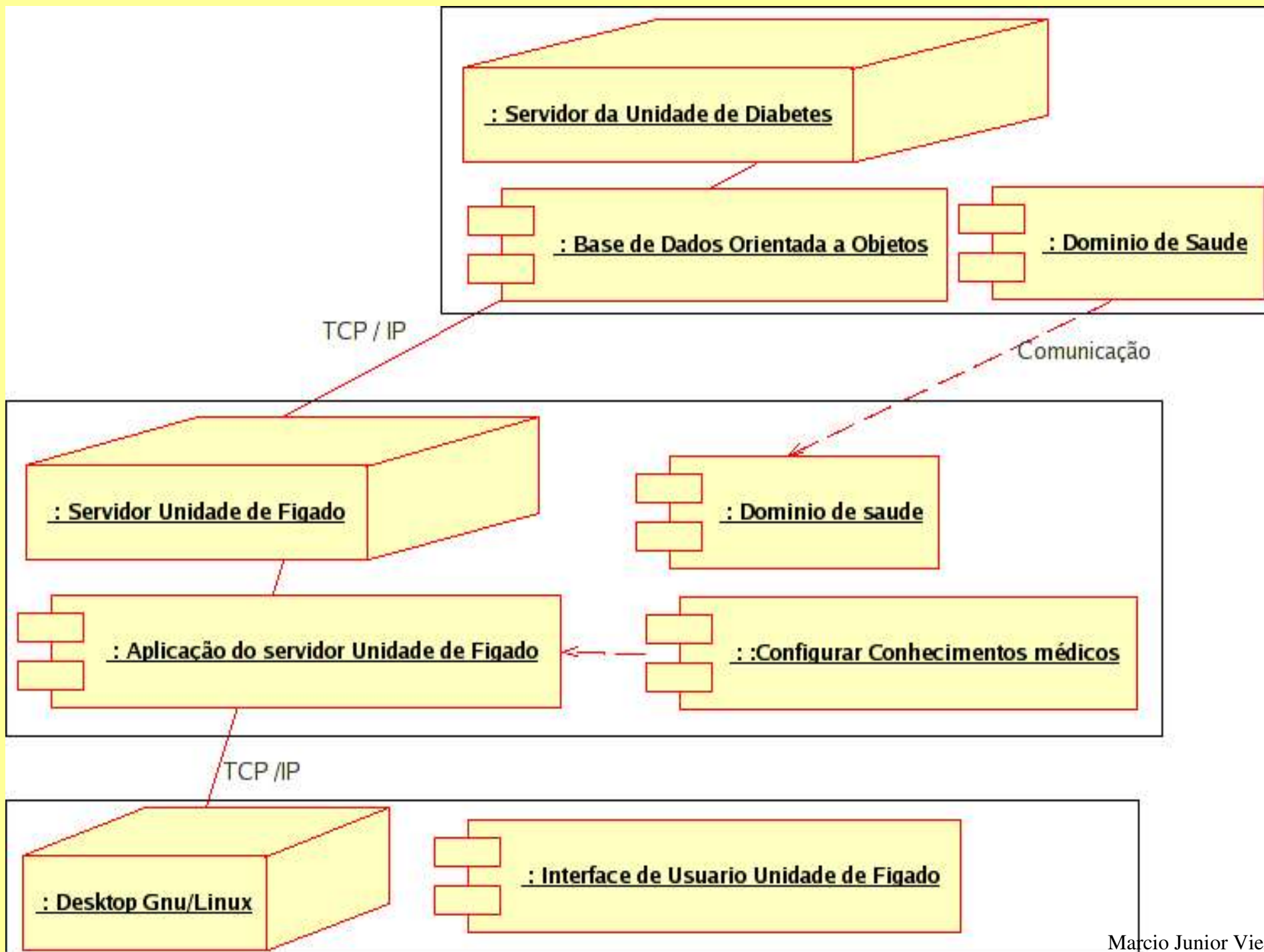


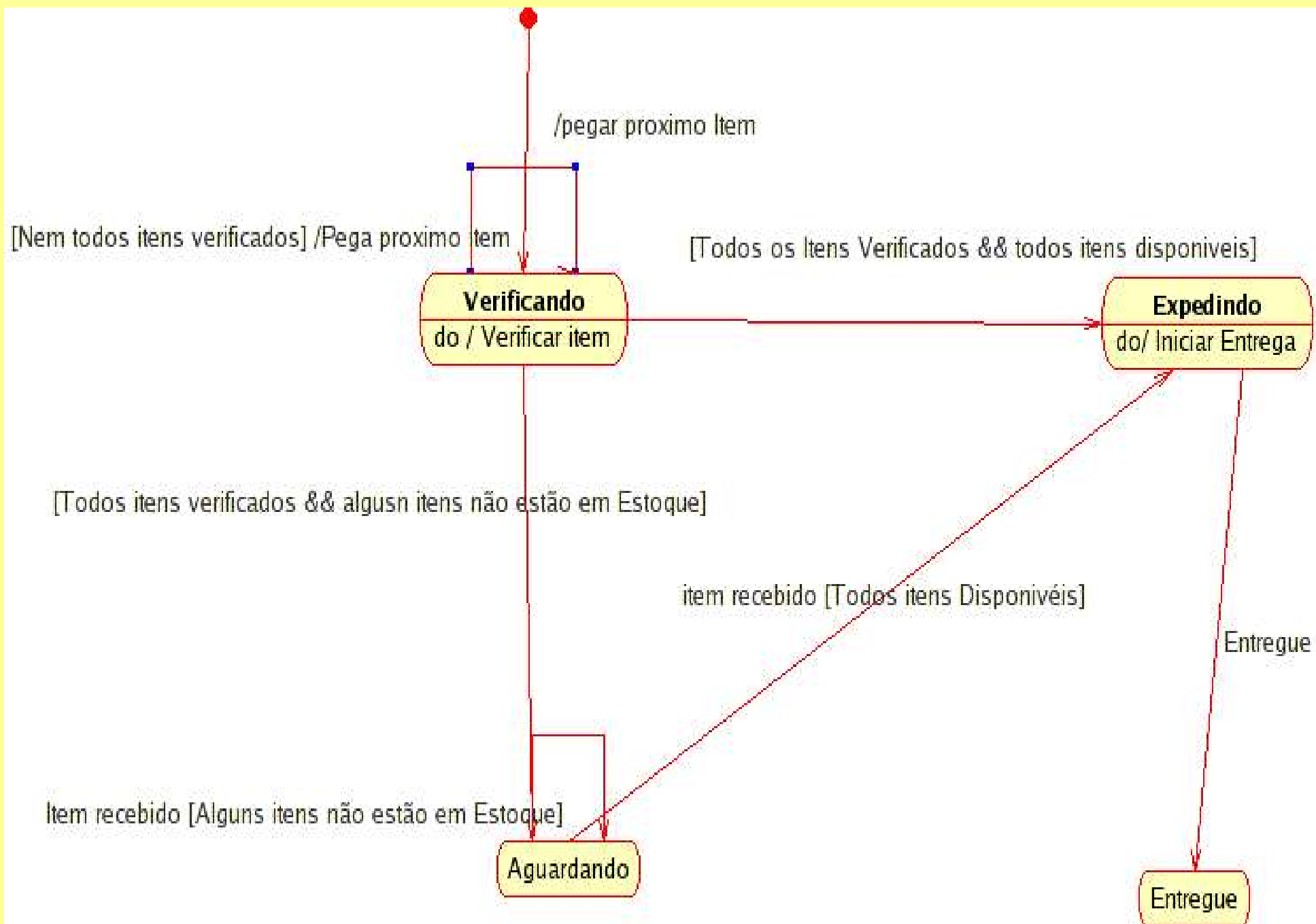
Modelagem UML Exemplo RH.▯

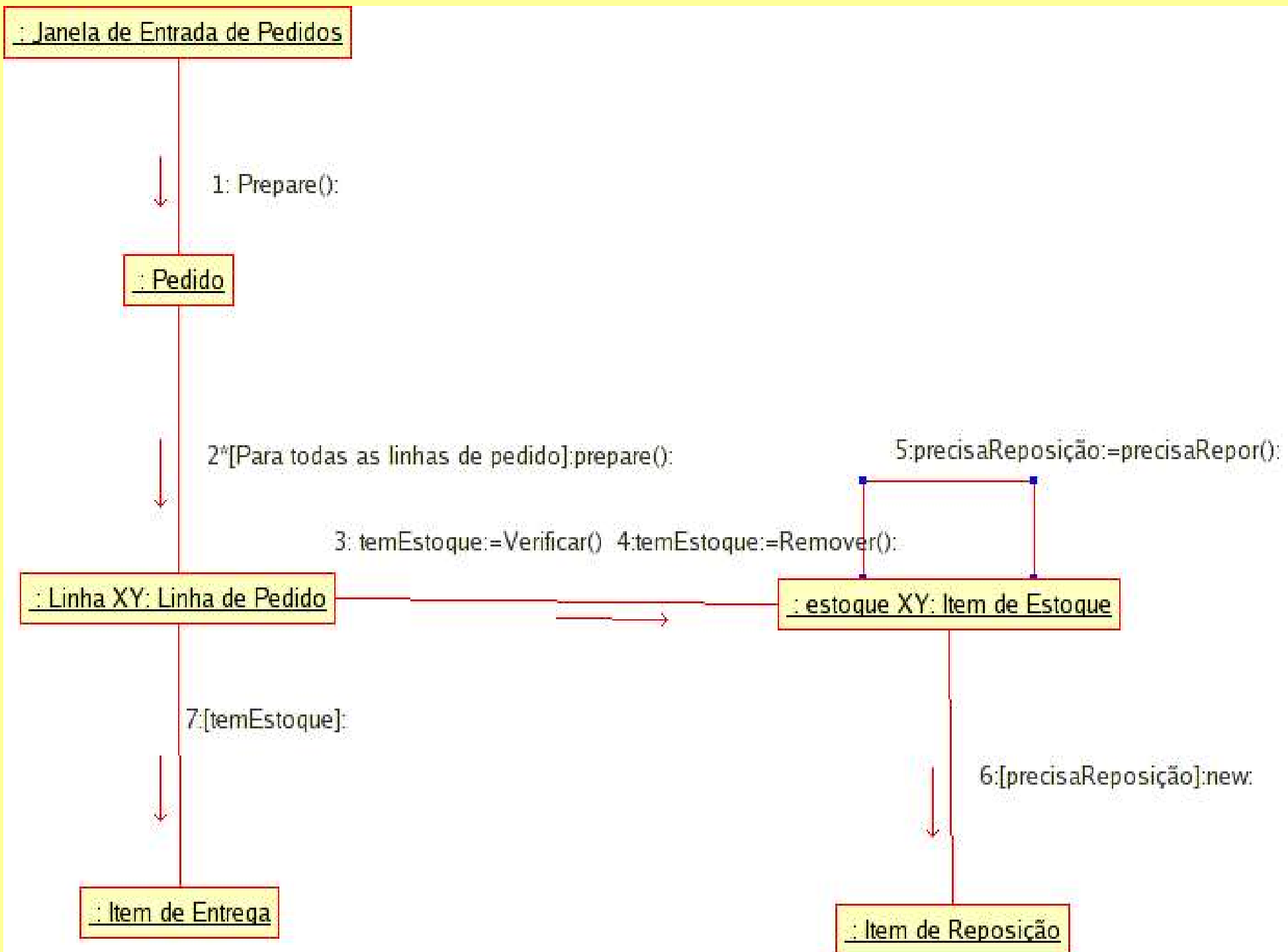


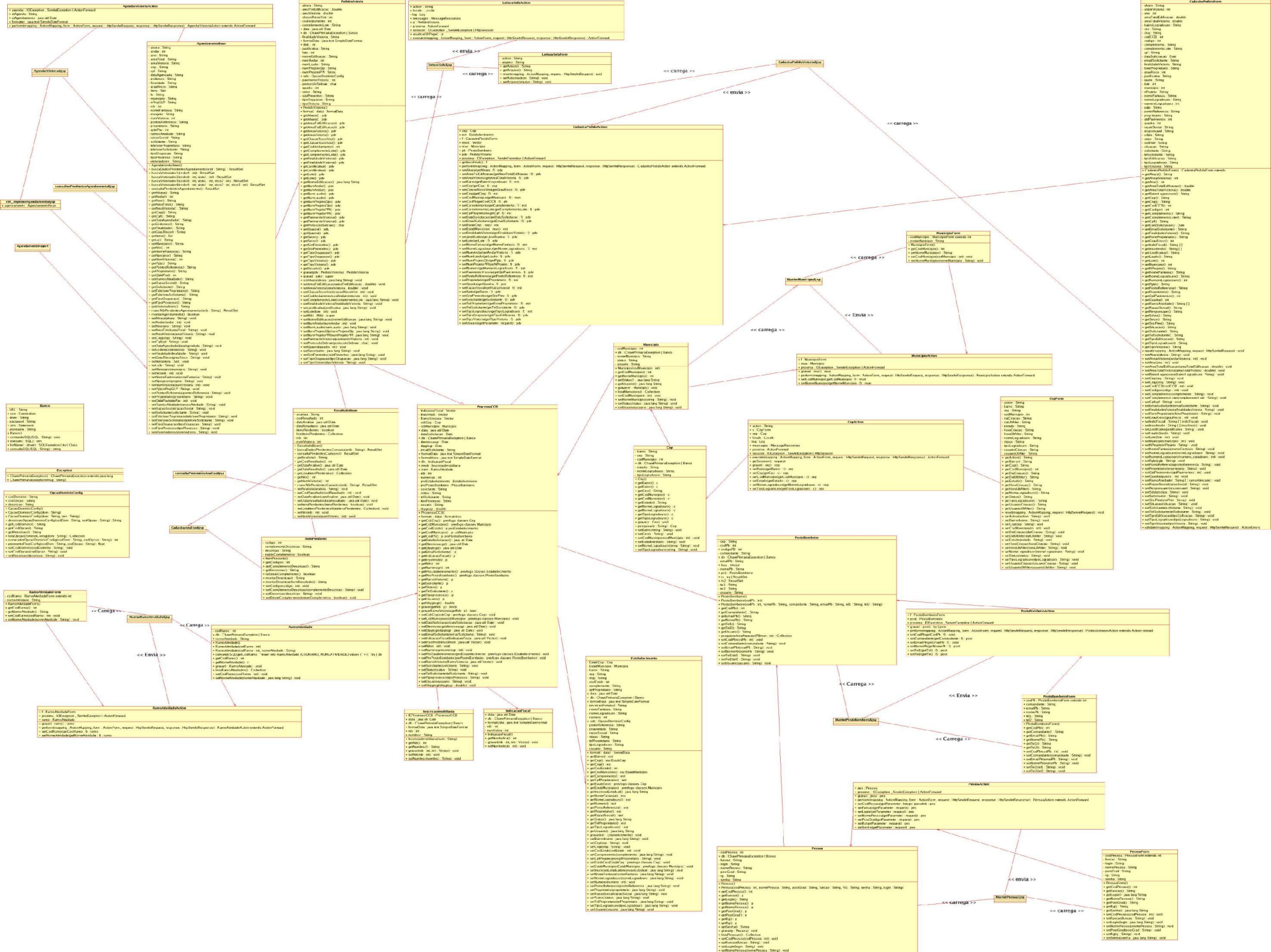












Umbrello UML Modeller - 1.5

The screenshot displays the Umbrello UML Modeller 1.5 interface. The main workspace shows a class diagram with the following classes and relationships:

- SWH_manager**: + setLevelTeam(), + setHeadProgrammer(), + setMarketingTeam().
- cellular company**: - name : string, - address : string.
- product**: - release : int, - headProgrammer : lead_programmer, - productManager : product_manager.
- product_manager**: (empty class).
- marketing_staff**: + sell().
- release**: - start_reselling, - end_reselling.
- order**: - orderStart : date, - guaranteeTime : date, - cellular company, - guaranteed_start : date, - contact : contact_man.
- malfunction**: - description : string, - open : date.
- contact_man**: - cellular company, - name : string, - address : string, - job : int, + problem().
- programmer**: + install(o : order) : installation_report, + close_malfunction(m : malfunction) : problem_report.
- lead_programmer**: (empty class).
- head_programmer**: + separate(), + setLeadProgrammer().
- installation_report**: - installer : programmer, - d : date, - c : contact_man.
- problem_report**: - start_fix : date, - end_fix : date, - release : int, - description : string, - p : programmer.
- function**: (empty class).

The interface includes a menu bar (File, Edit, Diagram, Code, Settings, Help), a toolbar with various icons, and a sidebar with "UML Diagrams" and "Documentation" panes. The status bar at the bottom shows "Ready." and the system tray with a clock displaying 23:52.

Please send contributions of screenshots to jr@jriddell.org.



Instalação do Umbrello UML Modeller

- Download do Umbrello : <http://www.umbrello.org>

Suse e outras distribuições com pacotes RPM

`rpm -Uvh umbrello-1.3.0-0-suse90.i586.rpm`

- **Debian e outras distribuições com apt-get**

`apt-get install umbrello`

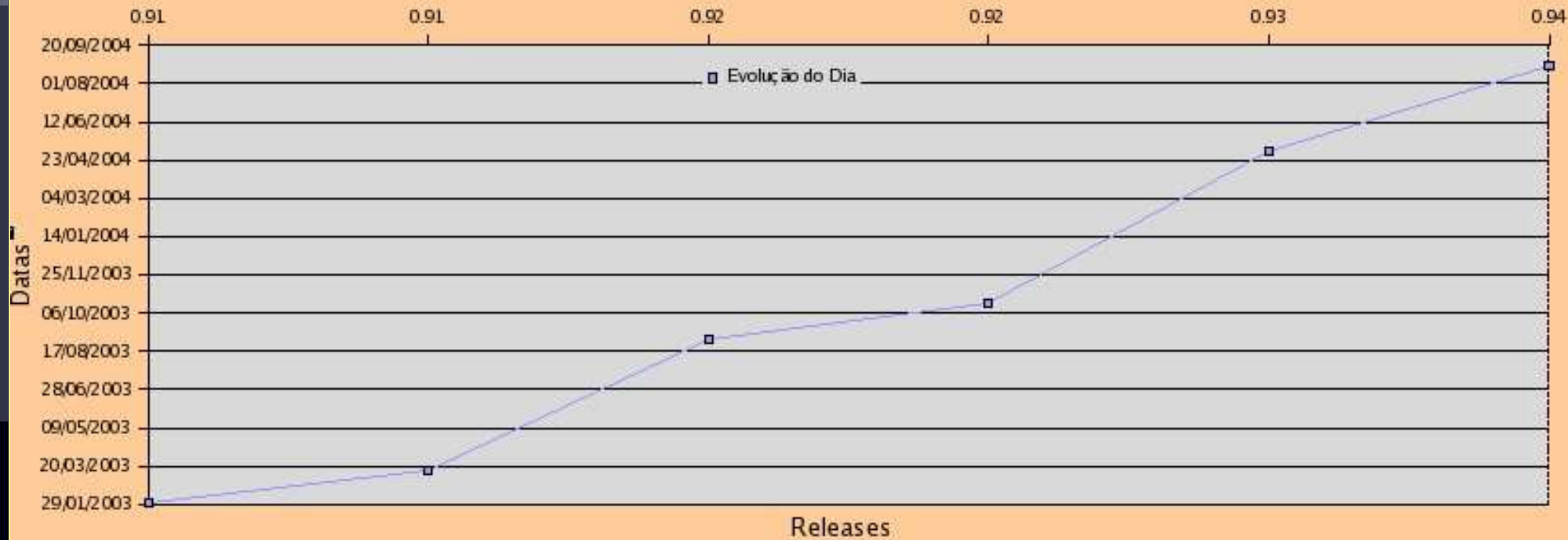
- **Gentoo**

`emerge kdesdk`

- Para demais distribuições deve ser compilado o código-fonte, para isso é necessário o pacote de desenvolvimento do KDE 3 e o Qt3, chamados de `kdelibs-dev` e `qt3-dev` ou similar, após instalado basta executá-lo pelo comando `umbrello`.

Evolução do Dia

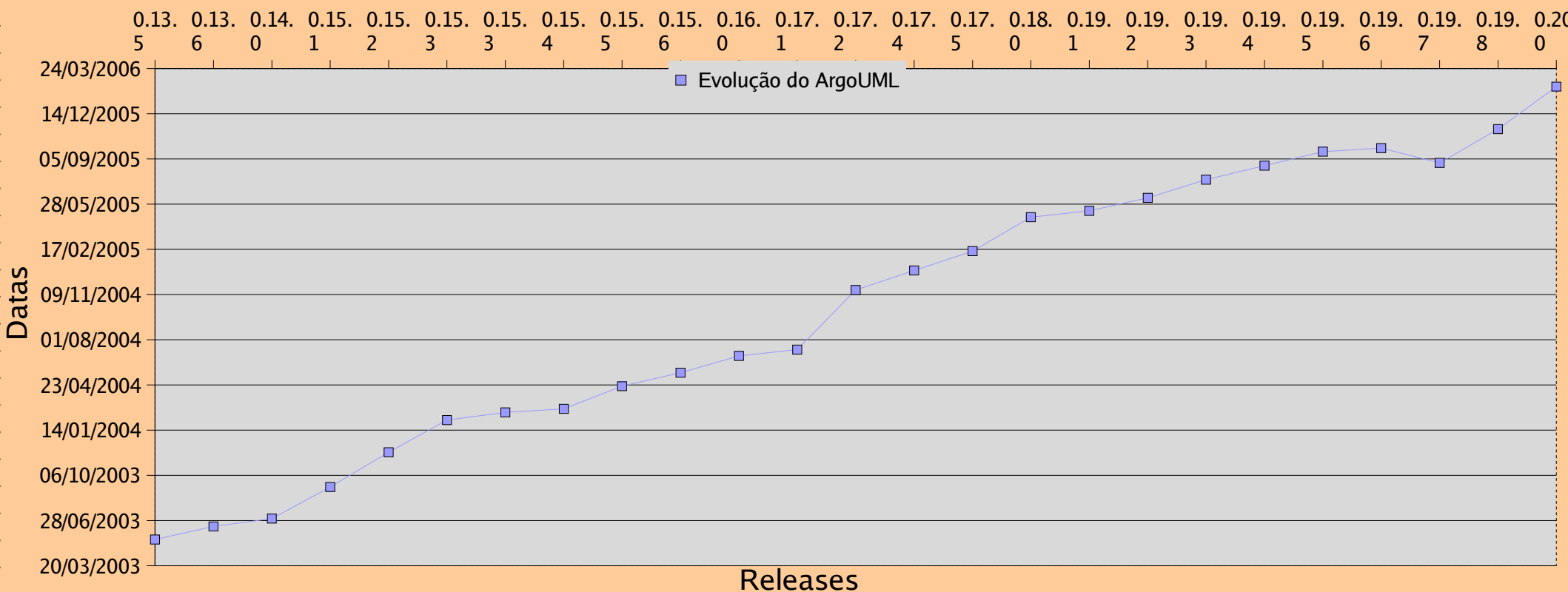
Releases Dia de 01/2003 a 08/2004 - Versão 0.91 a 0.94



Início do Projeto : 1998

Evolução do ArgoUML

Releases ArgoUML de 05/2003 a 02/2006 - Versão 0.13 a 0.20

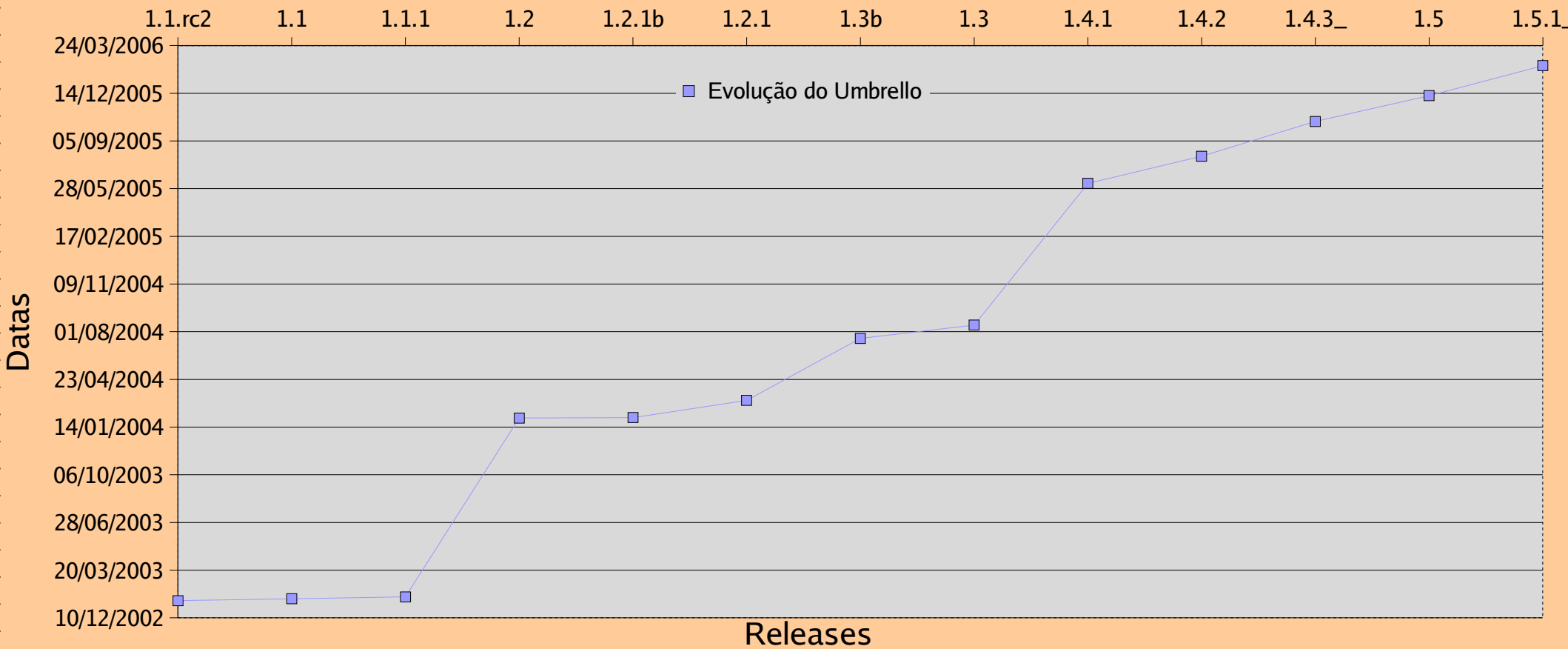


Início do Projeto : 1998

Média de 1 release a cada 3 meses

Evolução do Umbrello

Releases Umbrello de 01/2003 a 02/2006 - Versão 1.1 a 1.5



Início do Projeto : 2001

Referência

•Web Sites

- <http://www.umbrello.org>
- <http://codigolivre.org.br/projects/dia2sql/>
- <http://argouml.tigris.org>
- <http://www.gnome.org/projects/dia/>
- <http://www.omg.com.br>
- <http://www.andromda.com.br>
- <http://www.andromda.org>
- <http://www.agata.org.br/>

•Revistas e Livros

- SQL Magazine Edições 19,22,26 (Marcio Junior Vieira)
- Fowler, Martin
UML essencial: Um breve guia para a linguagem- padrão de modelagem de objetos/ Martin Fowler e Kendal Scott- 2.ed. - Porto Alegre:Bookman, 2000.
- Fowler, Martin
Refatoração: Aperfeiçoando o projeto de código existente.
Bookman, 2004.
- Lozano, Fernando
JAVA em GNU/ Linux – Rio de Janeiro: Alta Books, 2002

7º Fórum Internacional Software Livre



fisl 7.0



A tecnologia que liberta

19 a 22 de abril de 2006

Centro de Exposições FIERGS | Porto Alegre - RS - Brasil

<http://fisl.softwarelivre.org>

Patrocínio



Apoio



Promoção



Organização



Avaliação



Local



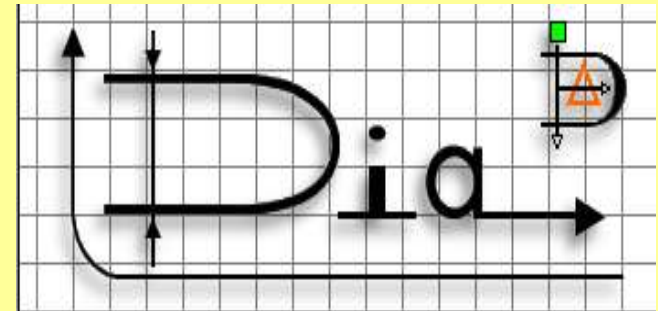
Divulgação FISL

- Expectativa de 5.000 participantes.
- Cerca de 200 palestrantes.
- Richard Stallman .
- Um dos maiores eventos de Software Livre do mundo.

Links

Dia

- www.gnome.org/projects/dia/ - Projeto Oficial



Umbrello

- www.umbrello.org - Projeto Oficial.



ArgoUML

- www.argouml.org - Projeto Oficial.
- www.tigris.org - Tigris open source.



Palestras

- www.ambientelivre.com.br/~marcio/ - Palestras.

UML

- www.uml.org - Unified Modeling Language.
- www.omg.org - Object management Group.



Contatos, convites para palestras, dúvidas

E-mail: marcio@ambientelivre.com.br