

# O Framework GREN

*Rosana T. Vaccare Braga*

# Conteúdo

- O Framework GREN
- Arquitetura do Framework GREN
- Algumas classes do GREN
- Exemplos da GUI do GREN
- Algumas dicas de utilização do VisualWorks para instanciar o GREN

# O Framework GREN

- Objetivo:
  - Construir classes reutilizáveis que permitam instanciar aplicações no domínio de gestão de recursos de negócios com o menor esforço possível
- Gestão de **RE**recursos de **N**egócios (GREN):
  - **Recursos de Negócios**: bens ou serviços gerenciados por aplicações na área de sistemas de informação
  - **Gestão**: quaisquer transações envolvendo locação, compra, venda, ou manutenção de bens ou serviços

# O Framework GREN

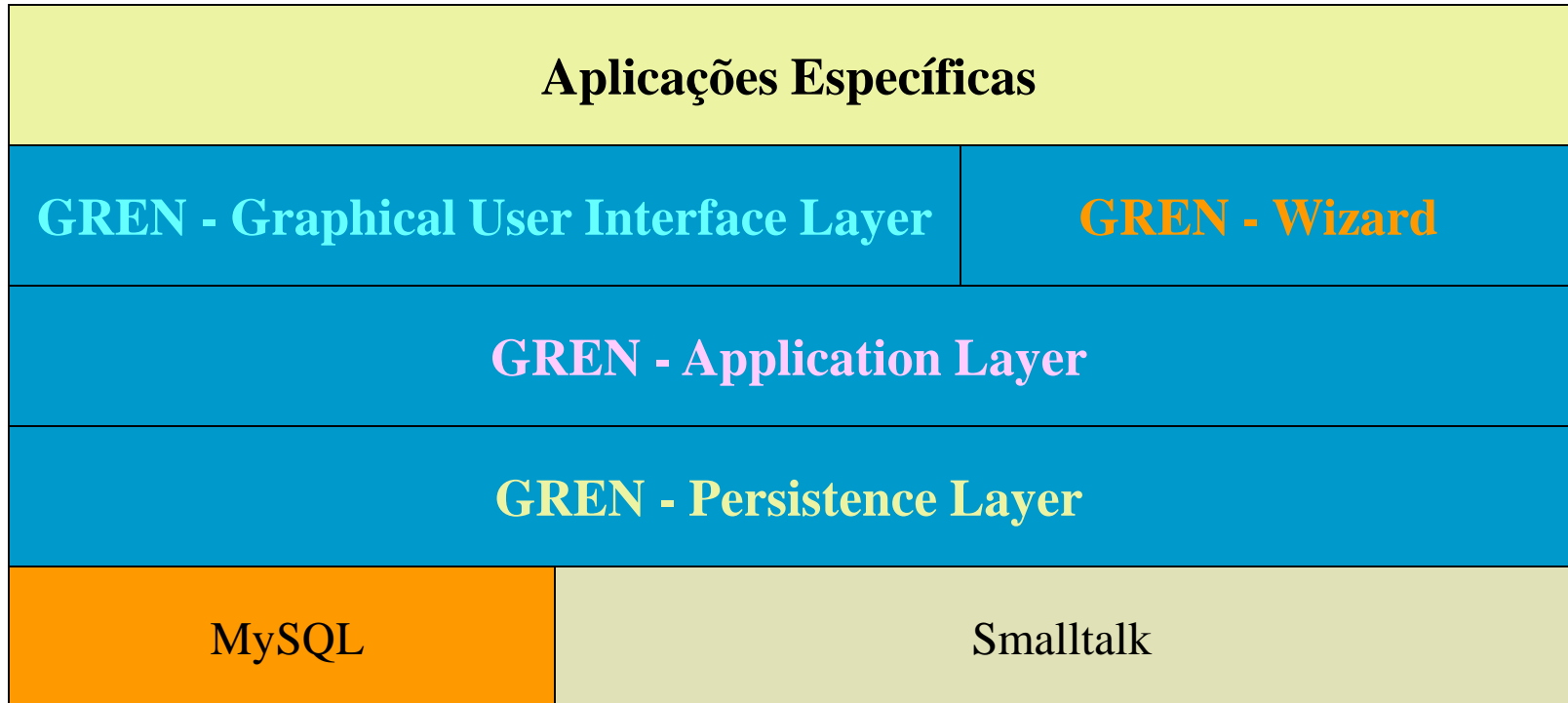
## ■ Tipos de Gestão:

- **Locação**: transferência temporária de um bem de uma entidade para outra
- **Comercialização**: transferência definitiva da posse de um bem
- **Manutenção**: conserto de um bem, podendo envolver serviços e produtos comercializados.

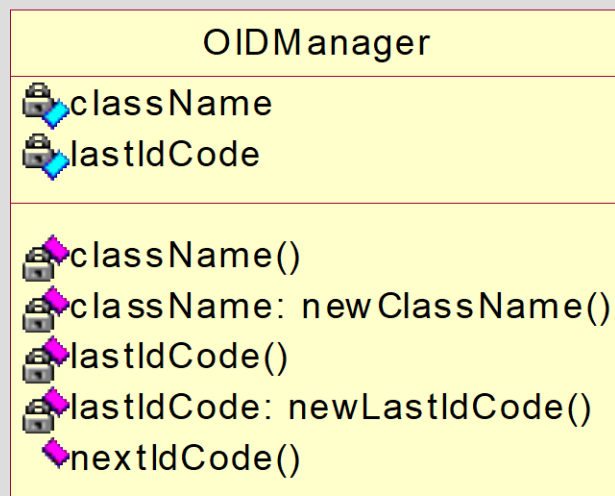
# O Framework GREN

- Ambiente utilizado:
  - VisualWorks 5i.4 NC
  - Base de dados MySQL
- Documentação existente:
  - Diagrama de classes
  - Código fonte
  - Manuais de instanciação do framework para uma aplicação específica

# Arquitetura do framework GREN

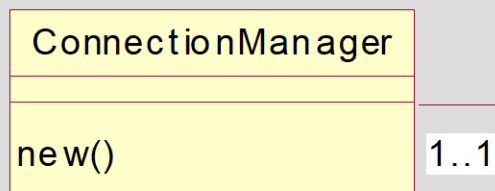


# Algumas classes do GREN - Camada de Persistência



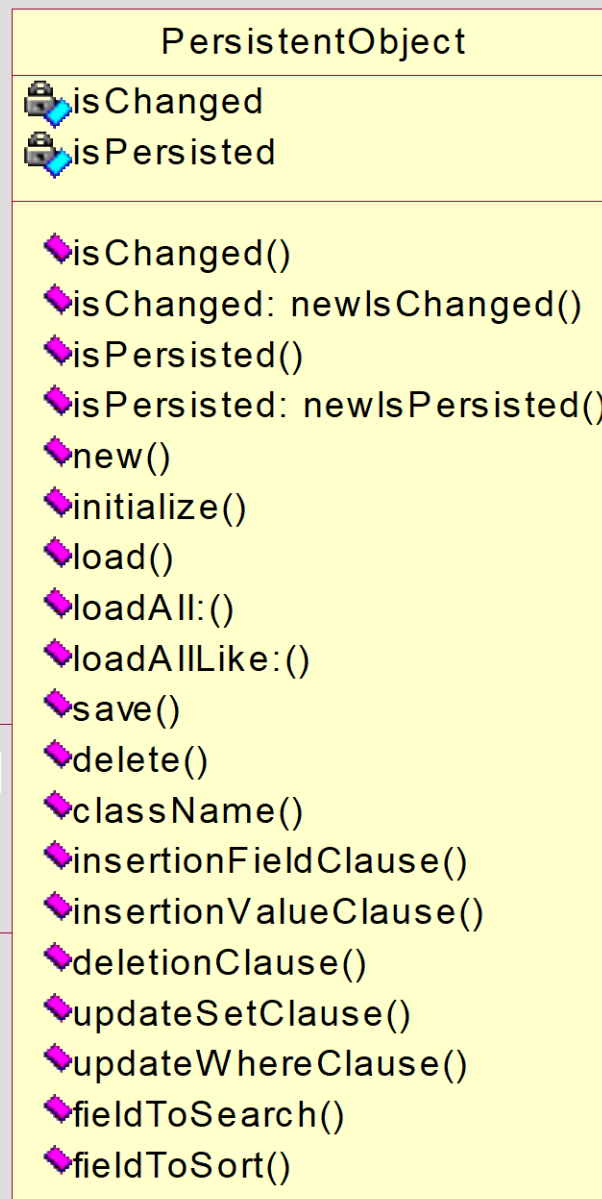
1..1

0..\*

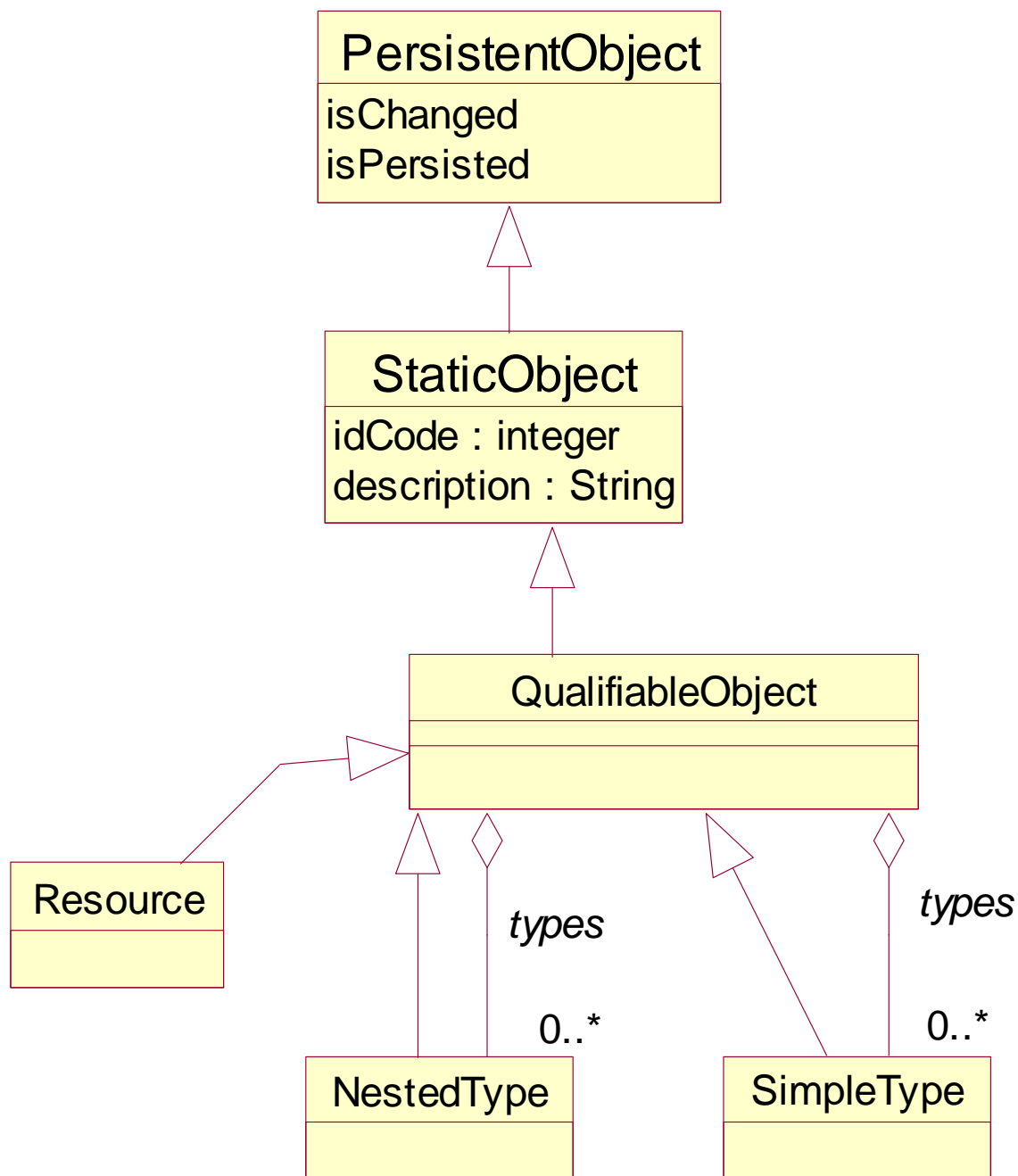


1..1

0..\*

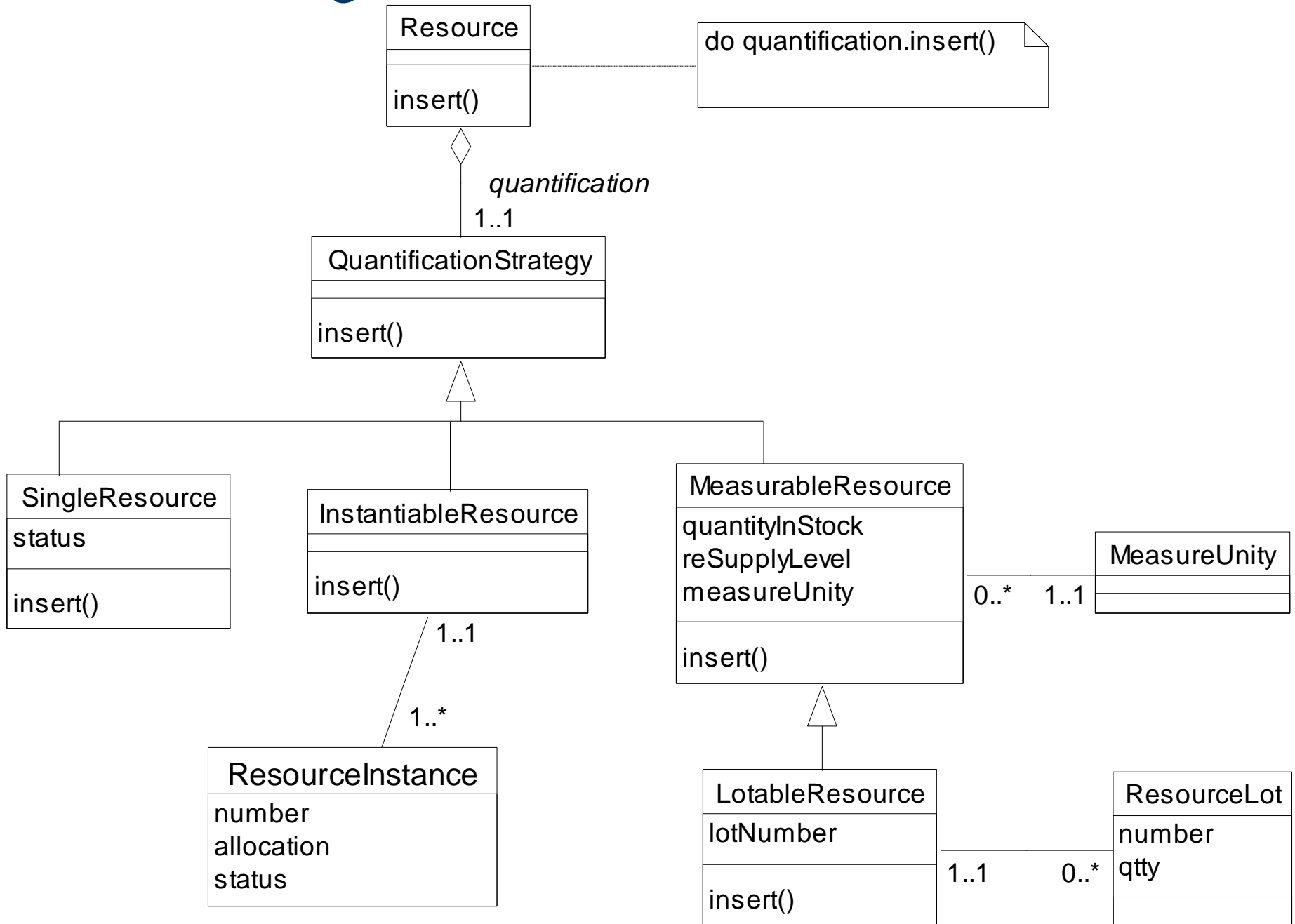


# Algumas classes do GREN – Camada de aplicação

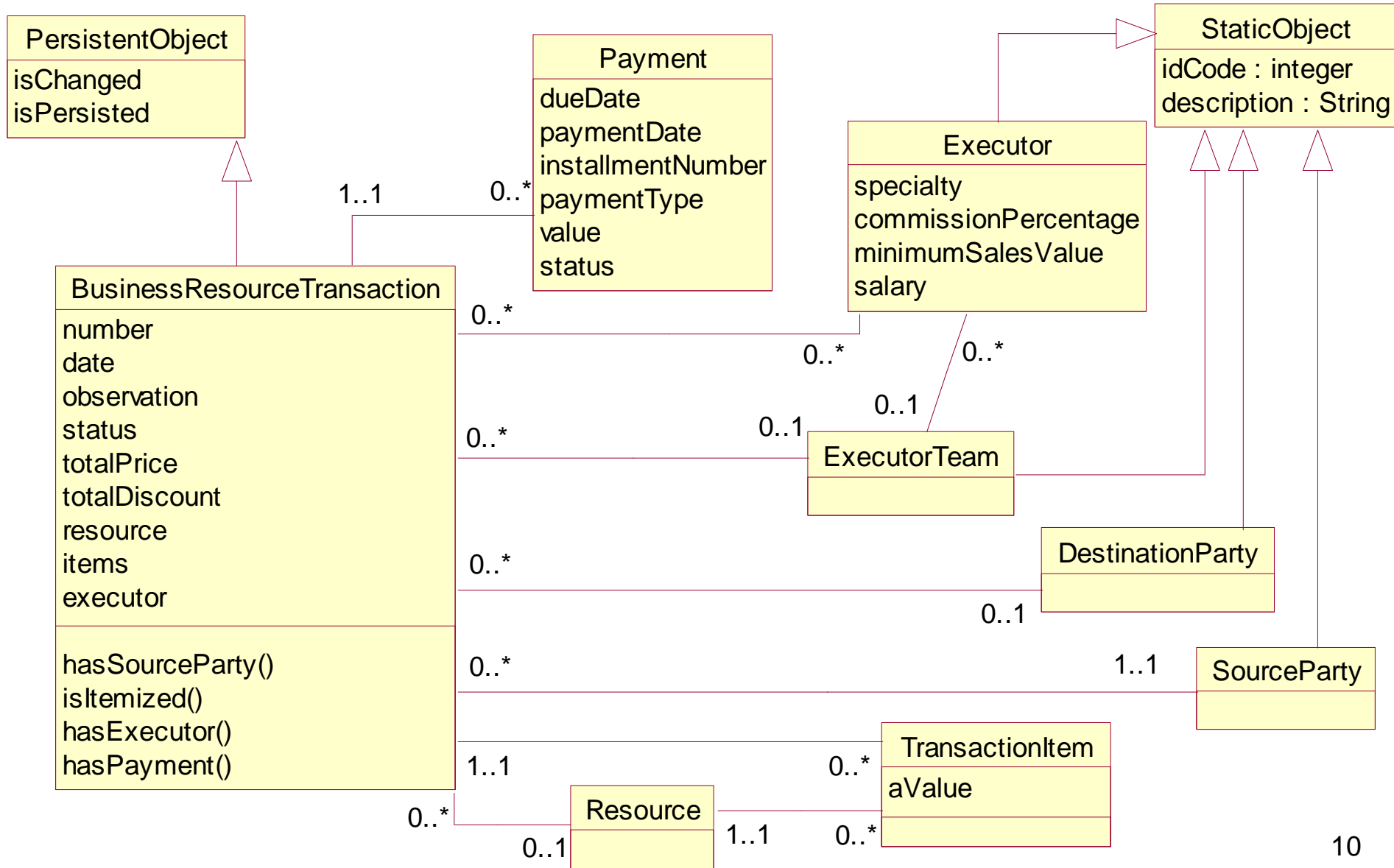




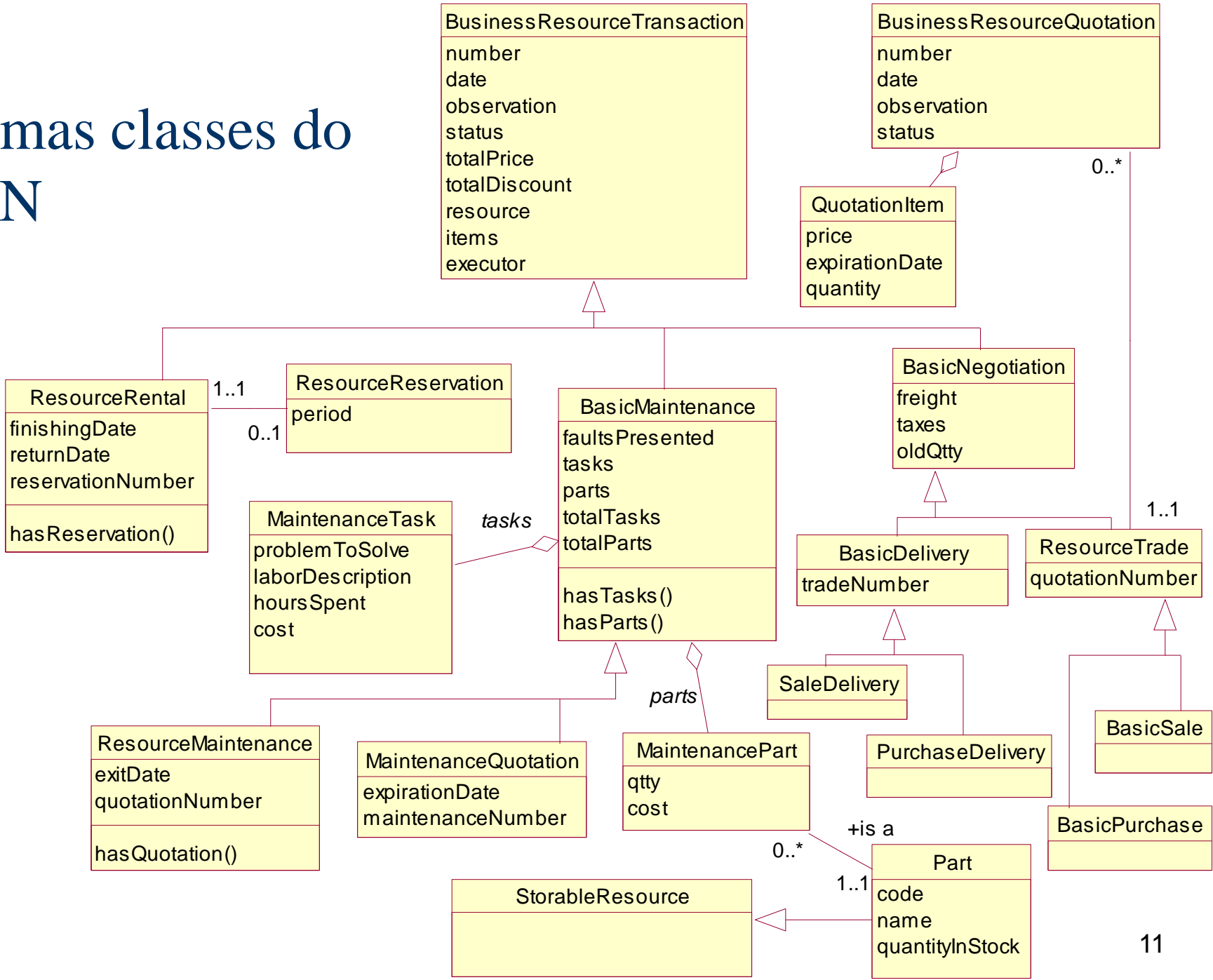
# Algumas classes do GREN



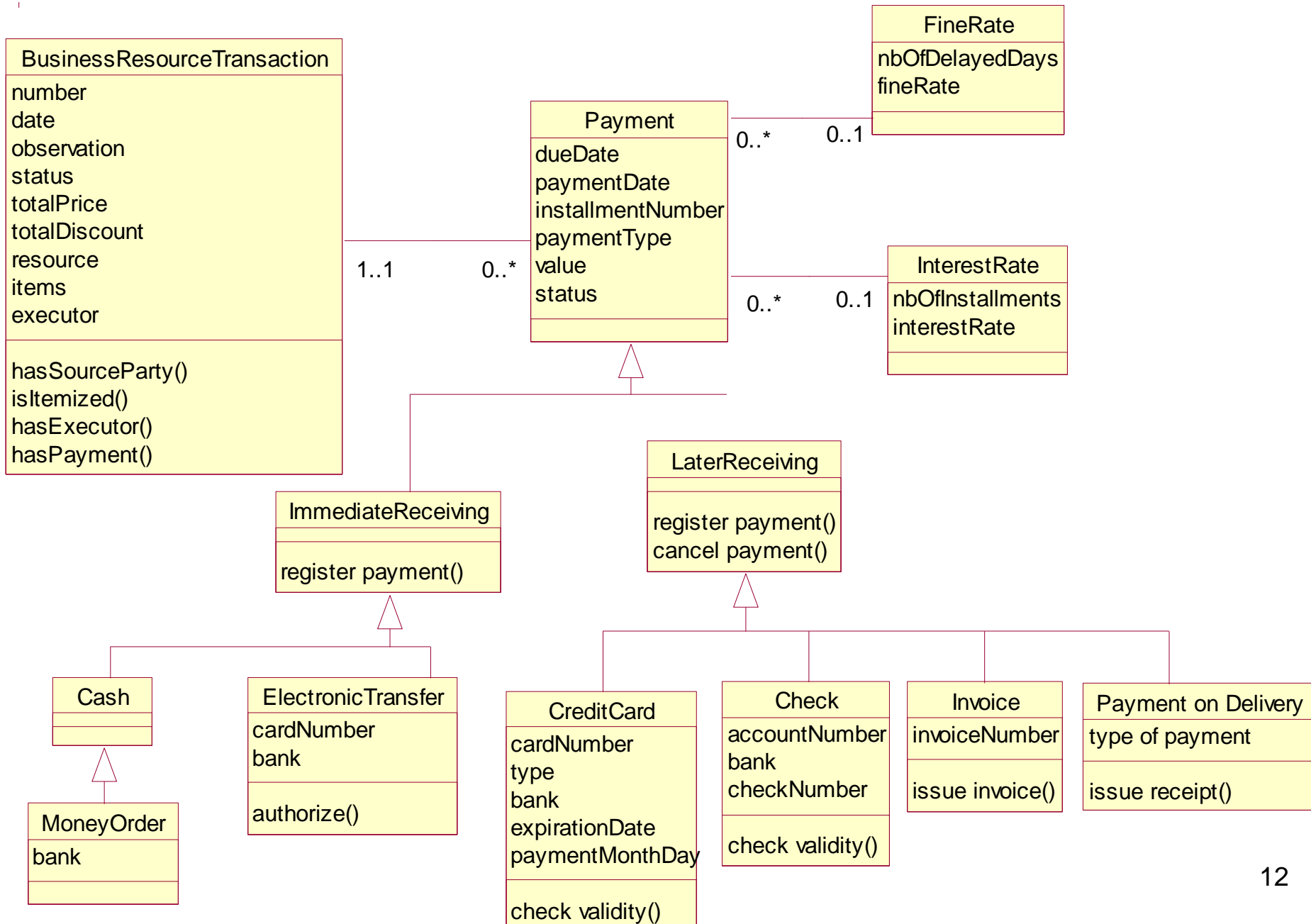
# Algumas classes do GREN



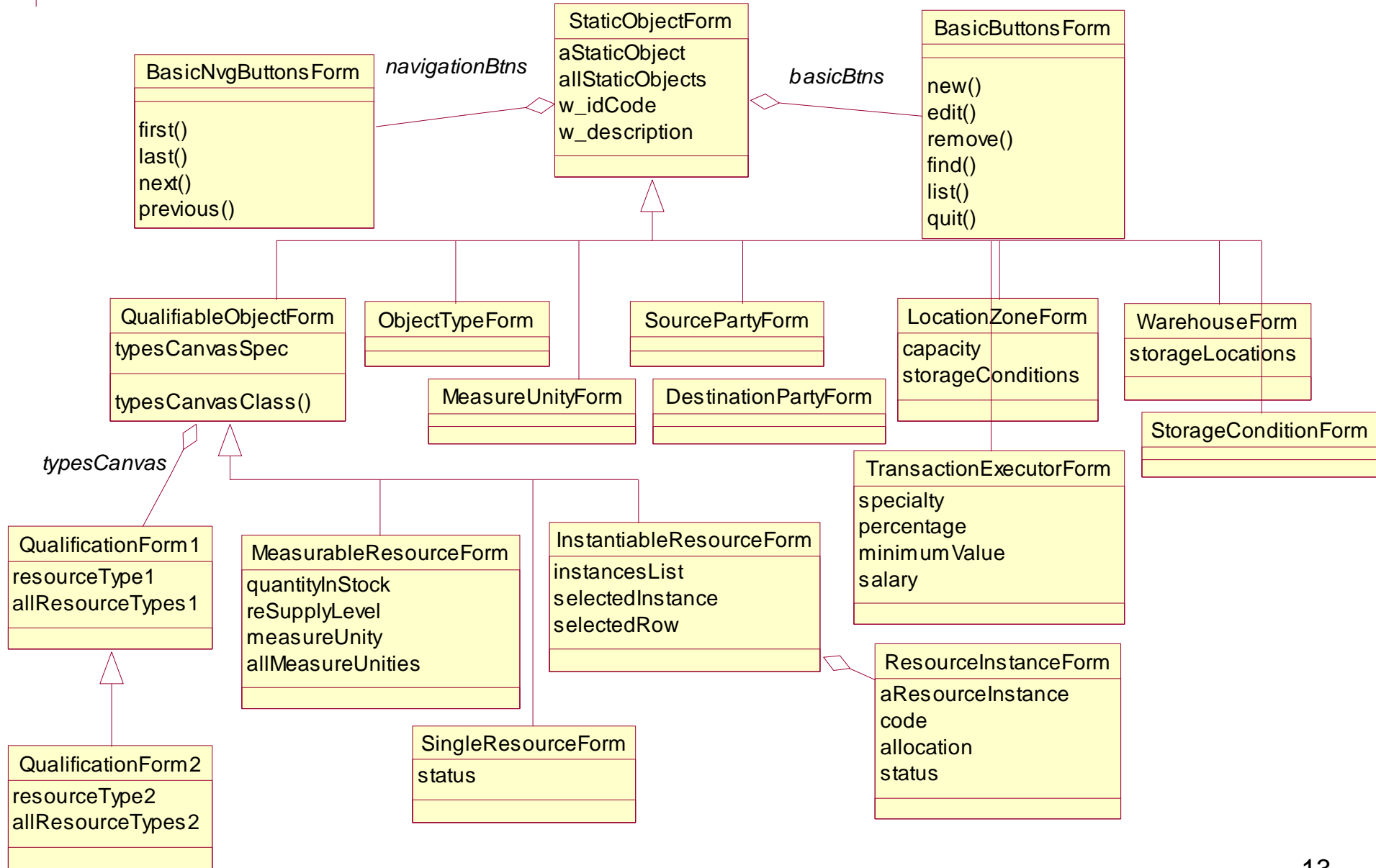
# Algumas classes do GREN



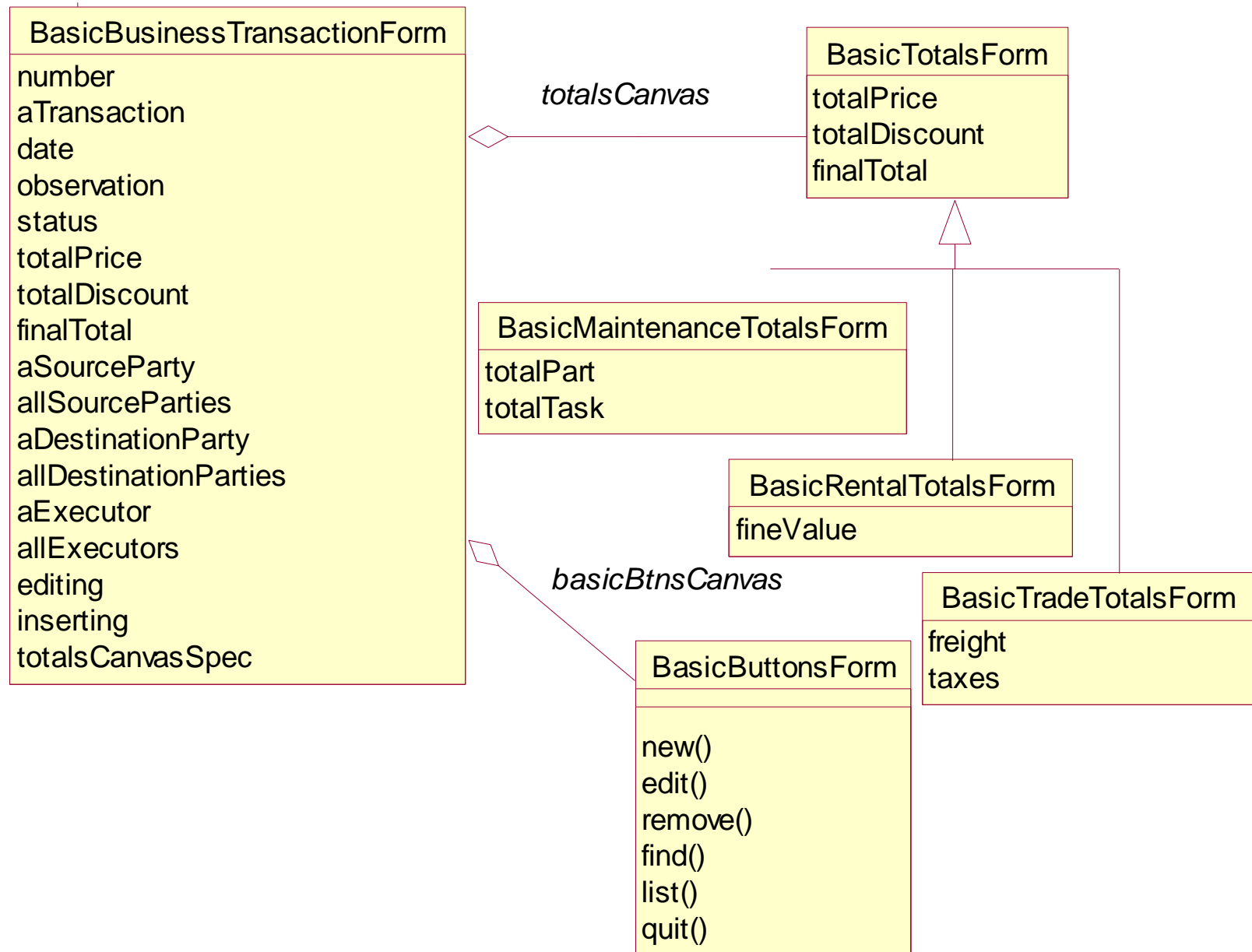
# Algumas classes do GREN



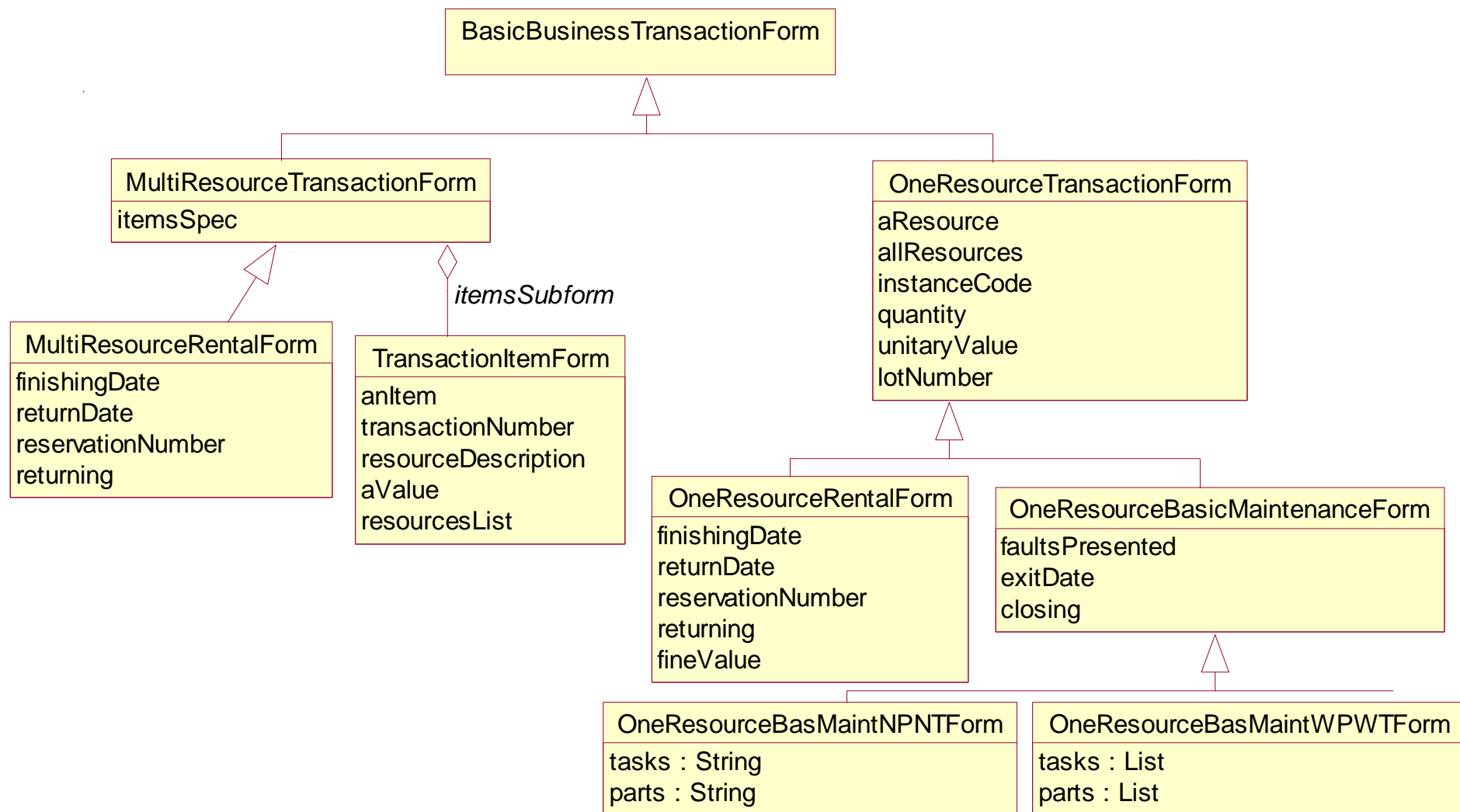
# Algumas classes do GREN - Camada GUI



# Algumas classes do GREN - Camada GUI



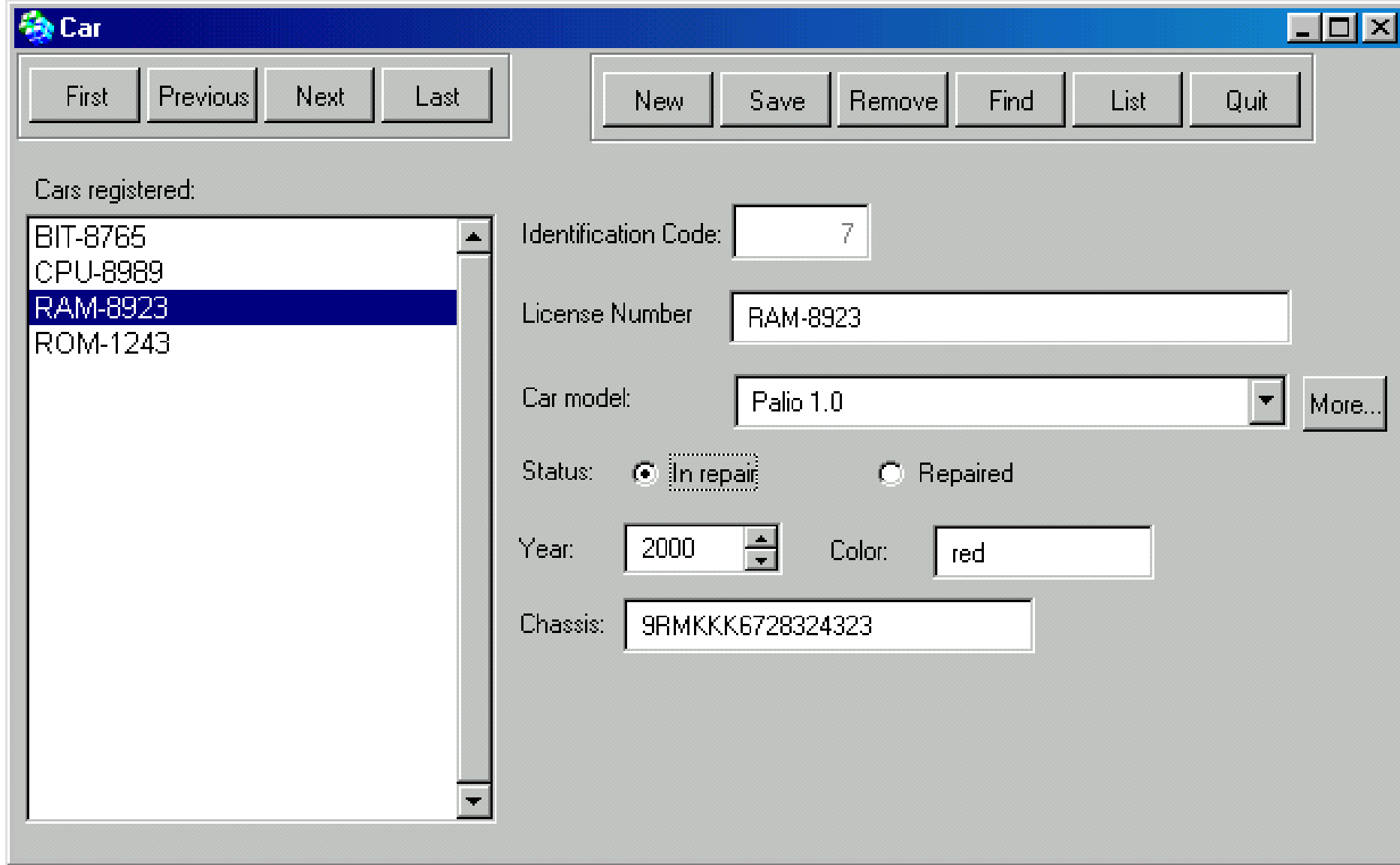
# Algumas classes do GREN - Camada GUI



## Exemplos da GUI do GREN

- O GREN oferece classes para ajudar a projetar a interface gráfica com o usuário (GUI)
- Existe uma variedade de GUIs que pode ser utilizada de acordo com cada aplicação específica, por exemplo, tipo de recurso, algumas configurações dos pontos variáveis do framework, etc.





## Exemplo de GUI para Recurso 17

**Car Rental** [Window Title Bar]

Buttons: New, Save, Remove, Find, List, Quit, Return Car

Status:  Pending,  Returned

Number:

Branch:

Customer:  More...

Date:  Finishing Date:  Devolution Date:

Observation:  Atendant:

Car:  More...

Total Price:   
Total Discount:   
Final Total:

Exemplo de GUI para locação de um recurso

Status

Pending  
 Returned

Rental number:  Reservation Number:

Branch:

Customer:

Initial Date:  Finishing Date:

Observation:  Atendente:

Cars rented:

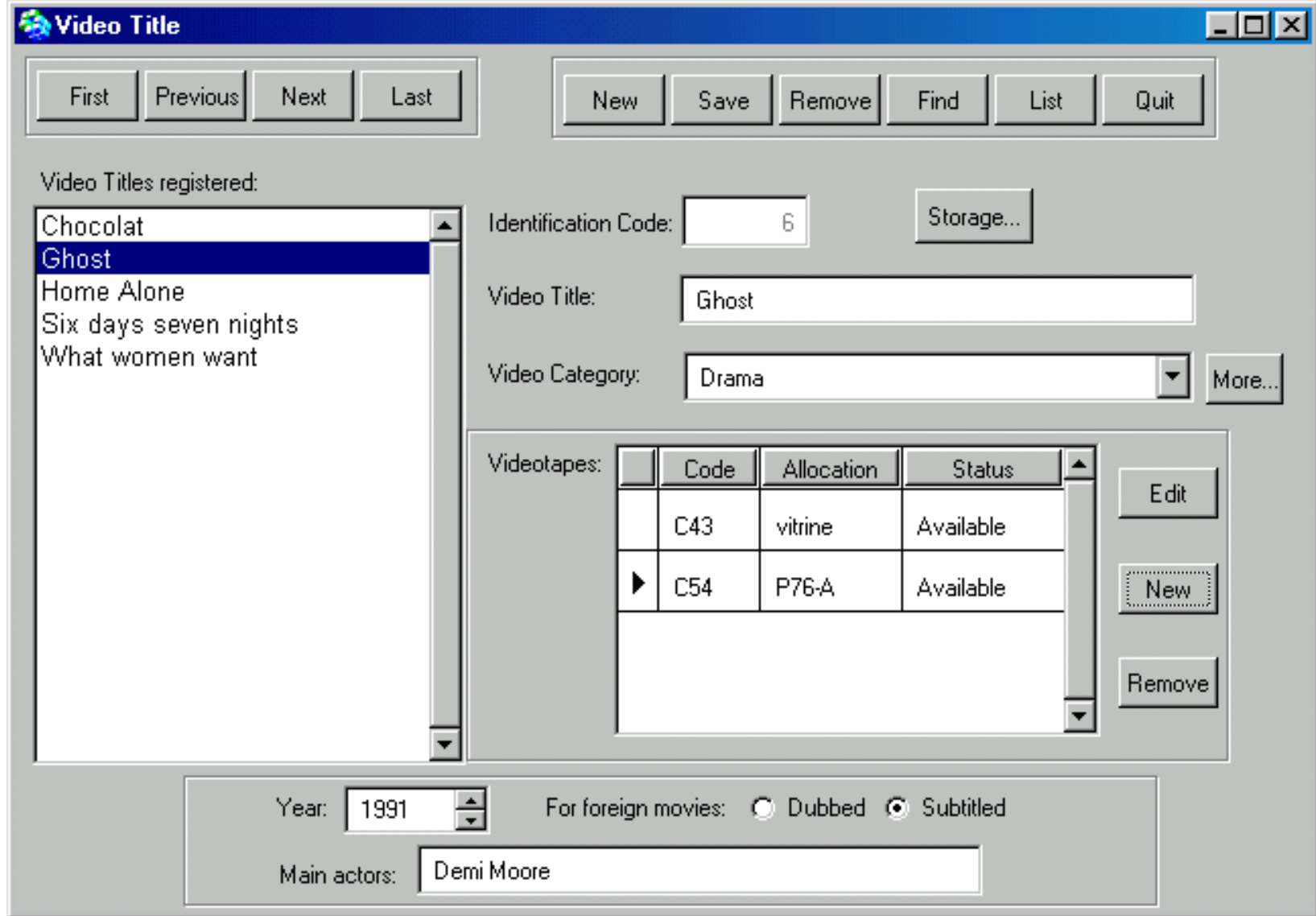
BIT-8765 - - 3.9	<input type="button" value="Edit"/>
ROM-1243 - - 3.9	<input type="button" value="New"/>
	<input type="button" value="Remove"/>

Total Price:

Total Discount:

Final Total:

Exemplo de GUI para locação de diversos recursos



Exemplo de GUI para um Recurso que possui vários exemplares

**Videotape Rental** [Minimize] [Maximize] [Close]

New Save Remove Find List Quit Return videotape

Status  Pending  
 Returned

Number:

Customer:  More...

Date:  Finishing Date:  Devolution Date:

Observation:

Video Title:  More...

Videotape Code:

Total Price:   
Total Discount:   
Final Total:

Exemplo de GUI para locação de recurso com vários exemplares

New Save Remove Find List Quit Return videotapes

Status  Pending  
 Returned

Number:

Customer:

Initial Date:  Finishing Date:

Observation:

Videotapes rented:

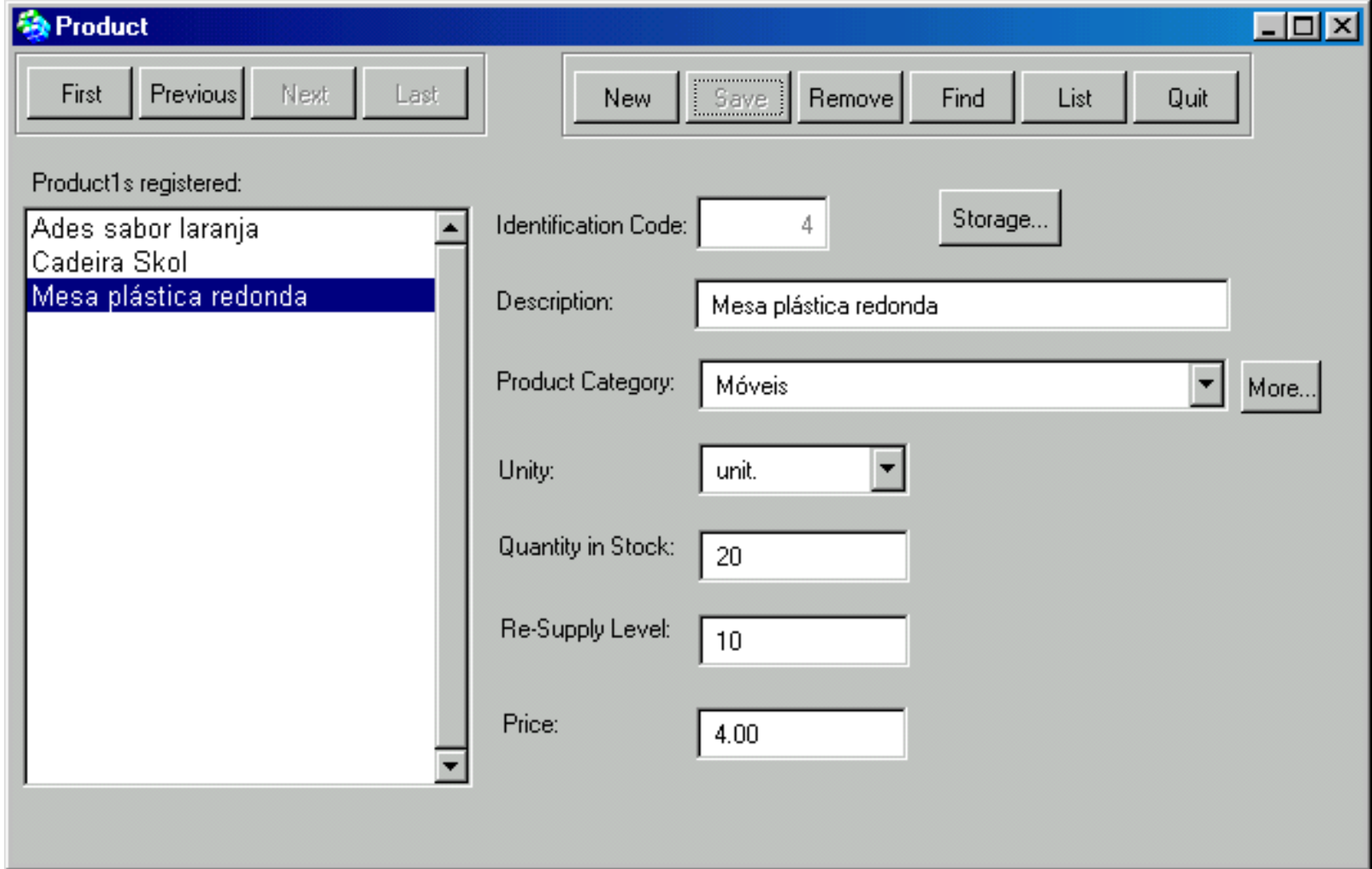
Home Alone - c3 - 8.3	<input type="button" value="Edit"/>
What women want - c4 - 17.56	<input type="button" value="New"/>

Total Price:

Total Discount:

Final Total:

# Exemplo de GUI para locação de diversos recursos



Exemplo de GUI para um  
Recurso alugado em quantidade

**Party Products Rental**

New Save Remove Find List Quit Return Product

Status  Pending  
 Returned

Number: 4 Reservation Number: 0 Search Reservation...

Branch: Matriz

Customer: Antonio Carlos do Amaral Fonseca More...

Date: 21-04-01 Finishing Date: 22-04-01

Observation: entregar no período da tarde Executor: José Antonio

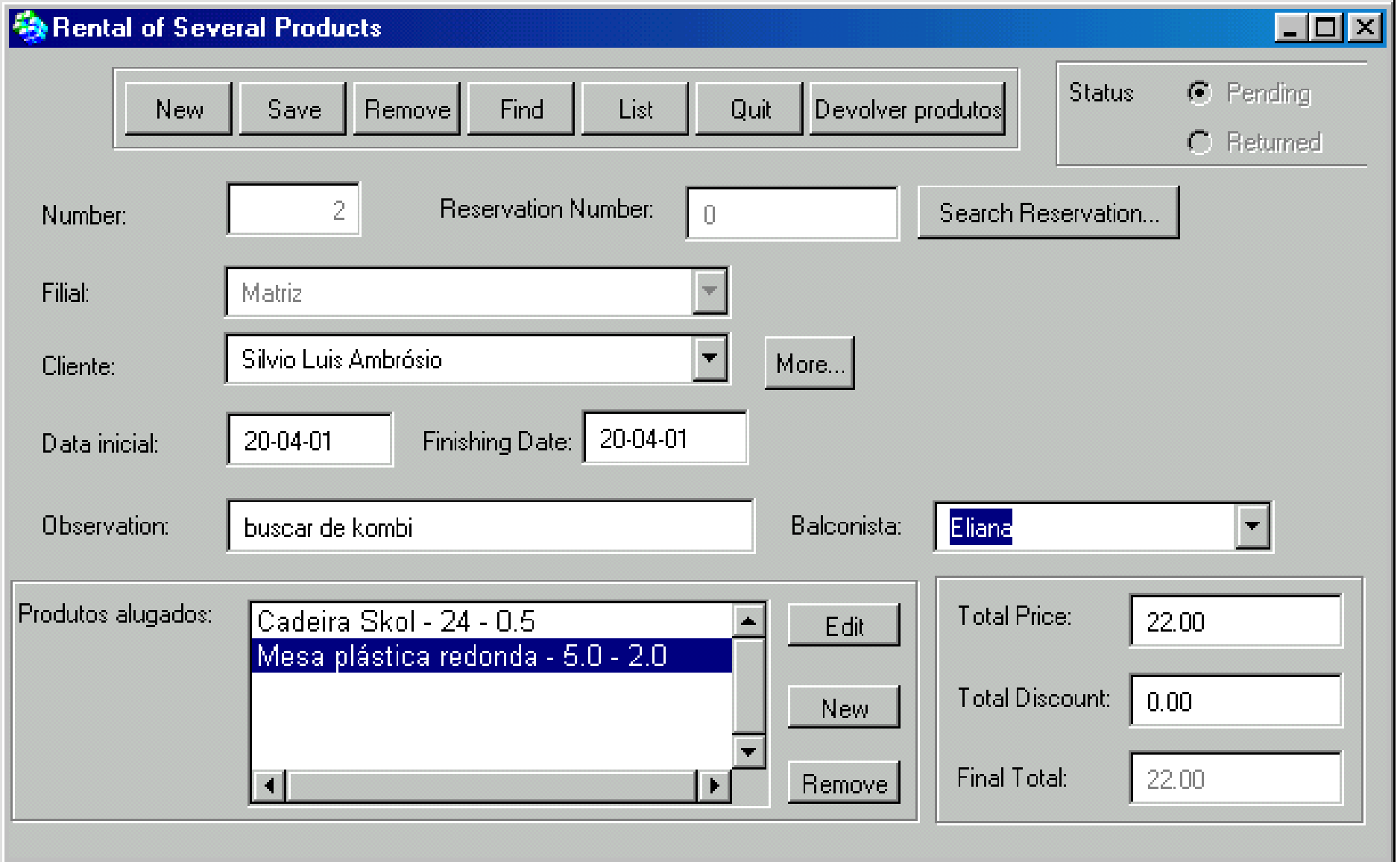
Product: Mesa plástica redonda More...

Quantity: 10

Total Price: 40.00  
Total Discount: 0.00  
Final Total: 40.00

Exemplo de GUI para locação de recurso alugado por quantidade





Exemplo de GUI para locação de recurso alugado por quantidade

**Car repair with tasks/parts**

Status:  Pending  Closed

Number: 
 Quotation Number:

Source Party:

Destination Party:

Entry Date:

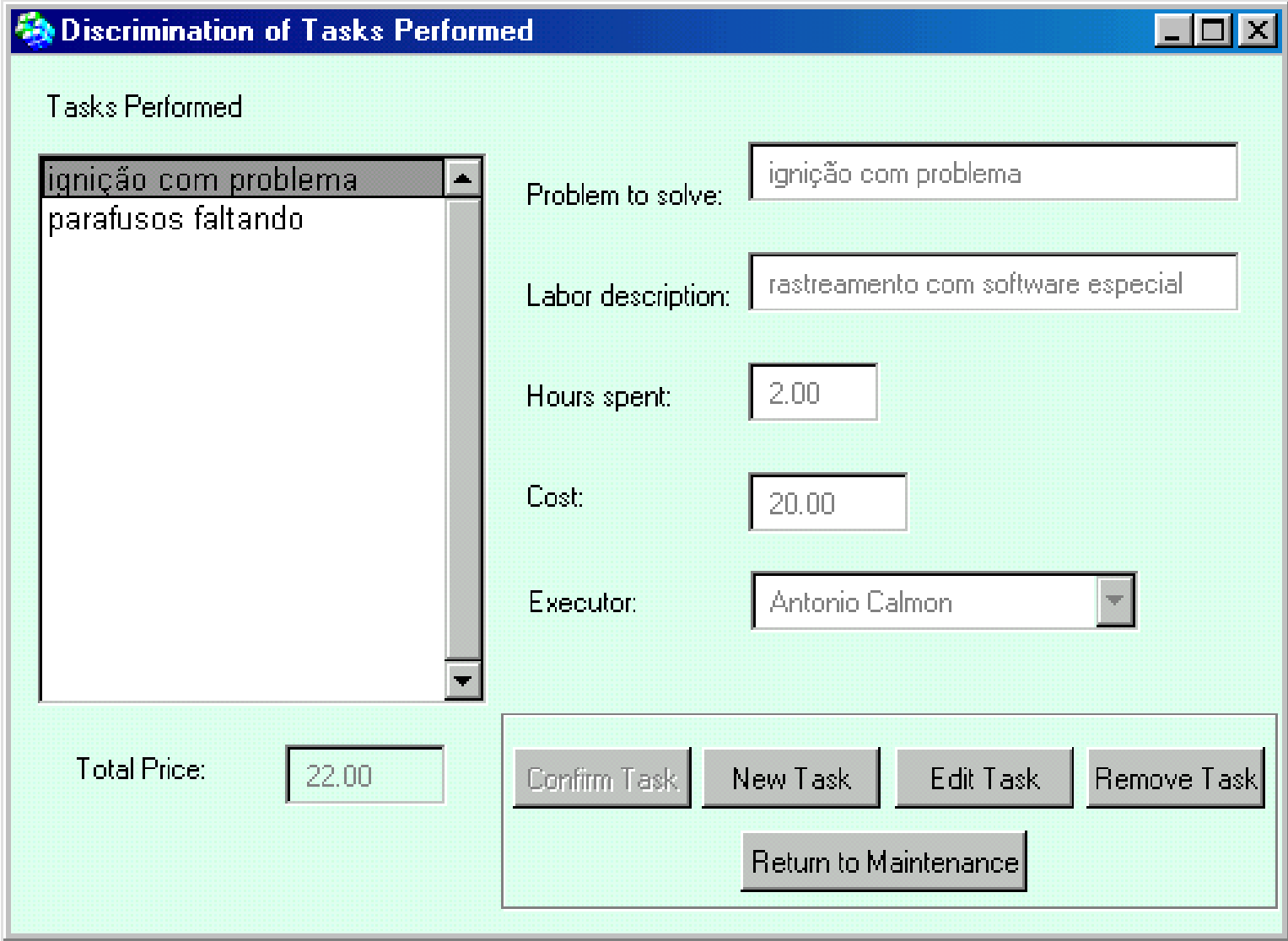
Observation: 
 Executor:

Resource:

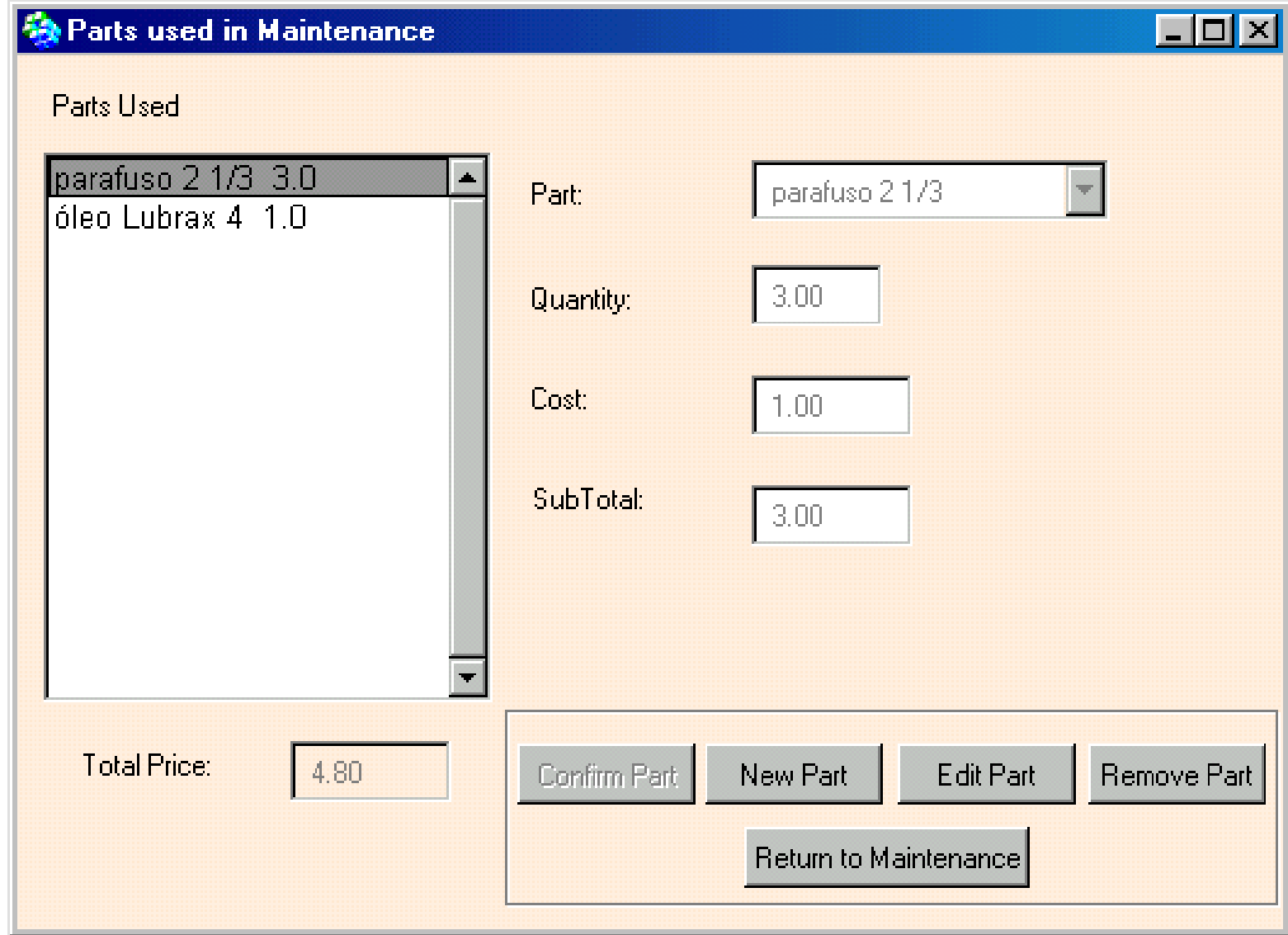
Faults presented:

Total Tasks	<input type="text" value="22.00"/>	Total Parts	<input type="text" value="4.80"/>	Total price	<input type="text" value="26.80"/>
		Total Discount	<input type="text" value="0.80"/>	Final Total	<input type="text" value="26.00"/>

## Exemplo de GUI para Manutenção de recurso



Exemplo de GUI para tarefas executadas



Exemplo de GUI para peças trocadas

Payment Details

Total # of Installments: 0

First Prev Next Last Installment #: 1

Status  Pending  Paid

Due date: Payment date:

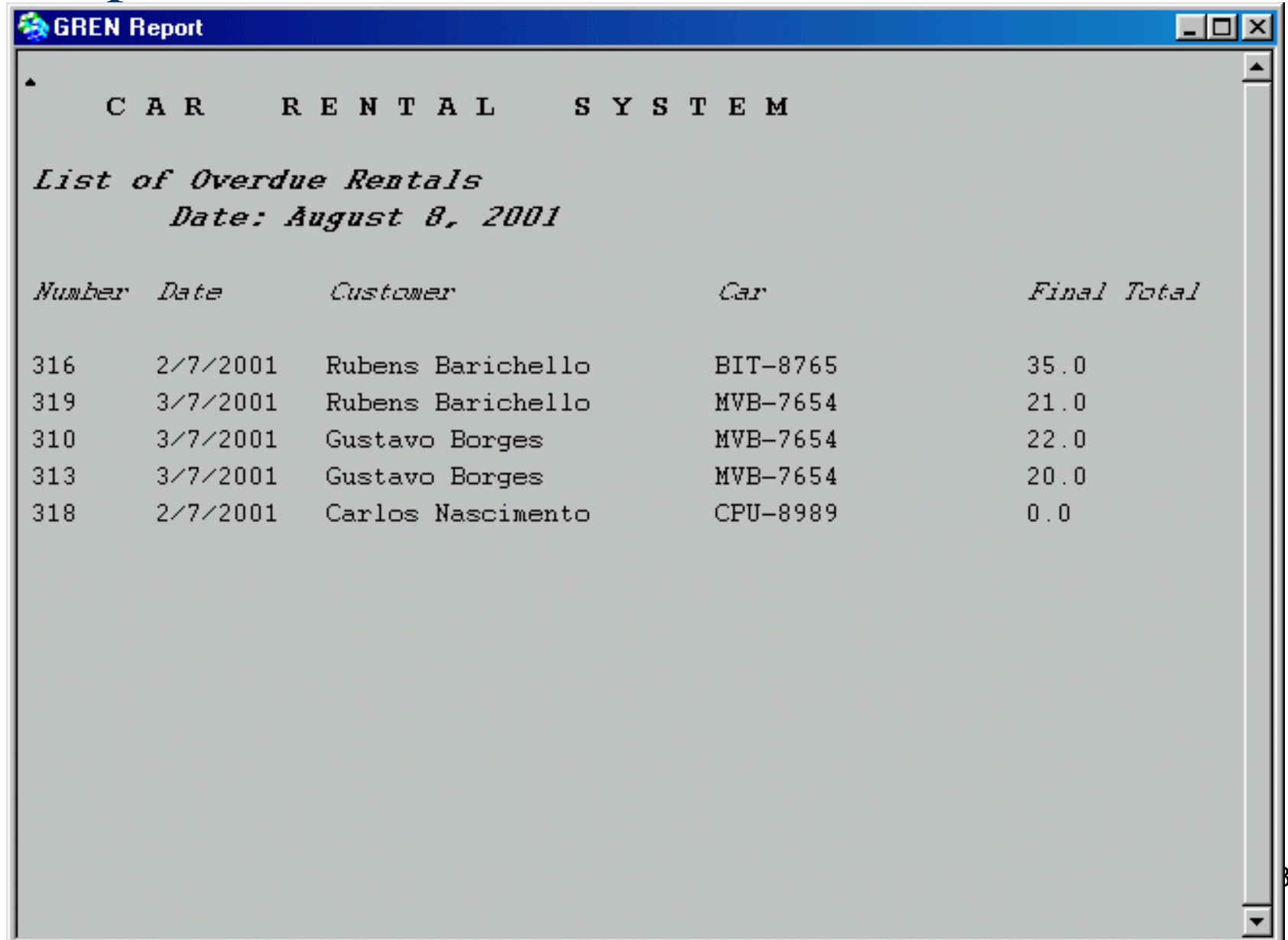
Payment Type:

Value: 0.00 Interest: 0.00 Fine: 0.00 Total: 0.00

Save Cancel

Exemplo de GUI para pagamento de parcelas

# Exemplo de relatório



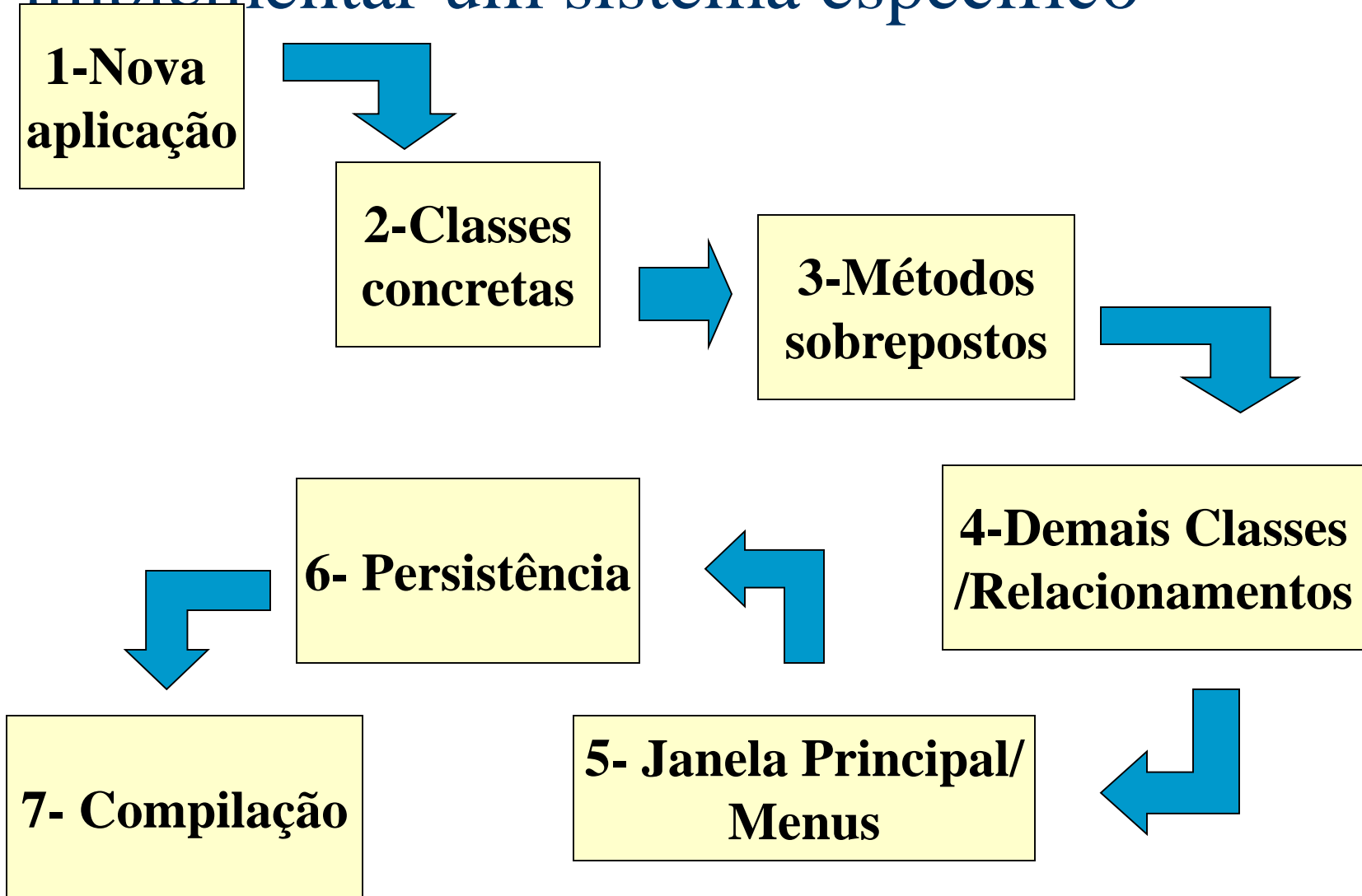
**GREN Report**

C A R   R E N T A L   S Y S T E M

*List of Overdue Rentals*  
*Date: August 8, 2001*

<i>Number</i>	<i>Date</i>	<i>Customer</i>	<i>Car</i>	<i>Final Total</i>
316	2/7/2001	Rubens Barichello	BIT-8765	35.0
319	3/7/2001	Rubens Barichello	MVB-7654	21.0
310	3/7/2001	Gustavo Borges	MVB-7654	22.0
313	3/7/2001	Gustavo Borges	MVB-7654	20.0
318	2/7/2001	Carlos Nascimento	CPU-8989	0.0

# Algumas dicas de Utilização do GREN para implementar um sistema específico



# Implementação do sistema usando o framework GREN

- **Passo 1:** Criar um novo projeto para conter a aplicação
  - No VisualWorks isso poder ser feito por meio de um novo Namespace e uma nova Categoria para conter as novas classes
- **Passo 2:** Criar as novas classes
  - Camada de aplicação e Camada de interface gráfica com o usuário
  - Consultar exemplos e utilizar a hierarquia de classes do GREN para saber de quem herdar
- **Passo 3:** Criar os métodos nas novas classes
  - Métodos obrigatoriamente a serem sobrepostos
  - Métodos opcionais, de acordo com novos atributos ou operações, ou para modificação de atributos das telas. (ver tabelas do anexo)



# Implementação do sistema usando o framework GREN

- **Passo 4:** Implementar novas classes/ relacionamentos não previstos pelo framework
- **Passo 5:** Criar a janela principal da aplicação, juntamente com os menus.
- **Passo 6:** Criar as tabelas no MySQL
- **Passo 7:** Compilar/instalar a nova aplicação

# Implementação do sistema usando o framework GREN

- **Passo 1:** Criar um novo projeto para conter a aplicação
  - Criar o Namespace chamado “Videolocadora”

```
Smalltalk defineNameSpace: #Videolocadora
  private: false
  imports: '
            private GREN.*
            private Smalltalk.*
          '
  category: 'Videolocadora'
```

# Implementação do sistema usando o framework GREN

## ■ Passo 2: Criar as novas classes

- Para cada uma das classes da nova aplicação, já organizadas durante o processo, existe uma **“Tabela de Mapeamento”** com quatro colunas:
  - nome da classe da aplicação concreta;
  - nome da superclasse correspondente no framework;
  - atributos acrescentados a essas novas classes mas que não fazem parte do framework
  - operações acrescentadas a essas novas classes, mas que não fazem parte do framework

# Implementação do sistema usando o framework GREN

## Exemplo da Tabela de Mapeamento - Videolocadora Classes da Aplicação

<b>Classe da aplicação a ser criada</b>	<b>Superclasse do GREN</b>	<b>Atributos adicionais</b>	<b>Métodos adicionais</b>
Filme	Resource	atoresPrincipais, anoDoFilme, precoDeLocacao, diretor, dubladoOuLegendado	(set e get para todos os novos atributos)
Genero	SimpleType	-	-
FitaDeVideo	ResourceInstance	-	-
Locacao	ResourceRental	-	-
ItemLocacao	TransactionItem	-	-
TaxaDeMulta	FineRate	-	-
Cliente	DestinationParty	endereco, cidade, estado, telefone, email, documentoIdentificacao, dataNascimento	(set e get para todos os novos atributos)

# Implementação do sistema usando o framework GREN

- Crie então cada classe concreta da aplicação, com base na “tabela de mapeamento”

## **Exemplo de criação de nova classe no VisualWorks**

```
Smalltalk defineClass: #Filme
  superclass: #{GREN.Resource}
  indexedType: #none
  private: false
  instanceVariableNames: 'atoresPrincipais
    anoDoFilme precoDeLocacao diretor
    dubladoOuLegendado'
  classInstanceVariableNames: ''
  imports: ''
  category: 'Videolocadora'
```

# Implementação do sistema usando o framework GREN

## ■ Passo 2: Criar as novas classes (cont.)

- Para cada classe na qual foram acrescentados novos atributos (ver a terceira coluna da tabela de mapeamento) deve ser criada uma classe de interface com o usuário, que possui o formulário de entrada/ alteração/ exclusão de dados. Nesse formulário devem ser acrescentados os campos dos tipos apropriados, de acordo com os atributos novos.
- Na quarta etapa foram identificadas essas classes, e já foram determinadas as classes do GREN a serem especializadas para gerá-la.

# Implementação do sistema usando o framework GREN

Pode ser criada uma tabela para auxiliar nesse mapeamento, como abaixo:

<b>Classe de interface a ser criada</b>	<b>SuperClasse do GREN</b>	<b>Nome do Spec no GREN</b>	<b>Spec a ser criado</b>
FilmeForm	InstantiableResourceForm	instResSpec	filmeSpec
FitaDeVideoForm	ResourceInstanceForm	windowSpec	*
GeneroForm	ObjectTypeForm	resTypeSpec	*
LocacaoForm	MultiResourceRentalForm	instResRentalSpec	*
ClienteForm	DestinationPartyForm	windowSpec	clienteSpec
ItemLocacaoForm	TransactionItemForm	instTransItemSpec	*

\* não há necessidade de criar o spec, pois não foram acrescentados atributos à essa classe durante a primeira etapa

# Implementação do sistema usando o framework GREN

## ■ **Passo 3:** Criar os métodos nas novas classes

### ■ **3.1. Métodos a serem sobrepostos:**

- Tabela 3 (camada de aplicação) e Tabela 4 (camada GUI)
- De acordo com a hierarquia de classes do GREN (um dos anexos), verifique, para cada nova classe criada na instanciação (ver tabelas de mapeamento), os métodos a serem sobrepostos (3ª coluna das Tabelas 3 e 4).
- Para tal, verifique se a nova classe herda de cada uma das classes que constam na 1ª coluna das tabelas 3 e 4.
  - Por exemplo, a classe Filme herda de Resource, que por sua vez herda de QualifiableObject. Portanto o método typeClasses deve ser sobreposto, no protocolo classReferences, e ele é um<sup>40</sup> método da classe.



# Exemplo da hierarquia de classes do GREN

**Object()**

**PersistentObject** ('isChanged' 'isPersisted')

1- **BusinessResourceTransaction** ('number' 'date' 'observation' 'status' 'totalPrice' 'totalDiscount'  
'finalTotal' 'destinationParty' 'sourceParty' 'resource' 'items' 'executor' 'transQuantification')

2- **BasicMaintenance** ('faultsPresented' 'tasks' 'parts' 'totalTasks' 'totalParts')

3- **MaintenanceQuotation** ('expirationDate' 'maintenanceNumber')

4- **ResourceMaintenance** ('exitDate' 'quotationNumber')

5- **BasicTrade** ('quotations' 'deliveries')

6- **BasicPurchase** ()

...

19- **StaticObject** ('idCode' 'description')

20- **DestinationParty** ()

21- **LocationZone** ('capacity' 'storageConditions')

22- **MeasureUnity** ()

23- **QualifiableObject** ('qualification')

24- **ObjectType** ()

25- **PaymentType** ()

26- **Resource** ('quantification')

27- **StorableResource** ('locationZone' 'storageConditions')

## Exemplo de parte da Tabela 3

Nome da superclasse	Protocolo	Nome do método a sobrepor	Classe/ Instância	Descrição
QualifiableObject	classReferences	typeClasses	C	Retorna uma lista com os nomes das classes referentes ao tipo do objeto ( 1 ou mais) ou nil se o objeto não possuir tipos. Exemplos: ^List with: CarType ^List with: Category with: MidiaType
StaticObjectForm/ QualifiableObject Form/ ResourceForm	classReferences	objectClass	C	Retorna o nome da classe concreta da camada de aplicação que contém o objeto sendo inserido/alterado no formulário, como no exemplo: ^VideoTitle

# Implementação do sistema usando o framework GREN

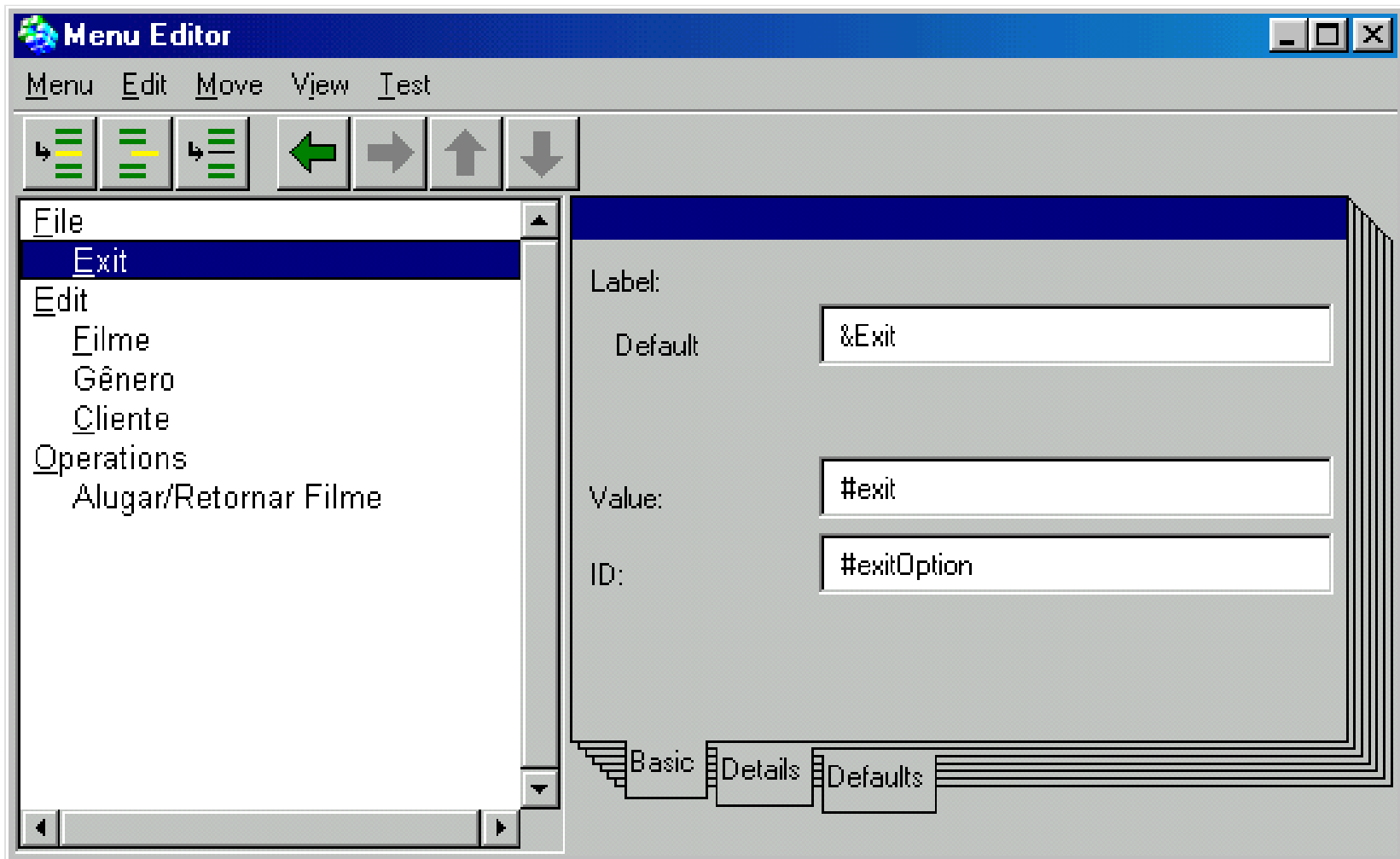
- **Passo 3:** Criar os métodos nas novas classes
- **3.2. Métodos opcionais a serem sobrepostos:**
  - **ESPECIFICAÇÃO DOS CANVAS:** no GREN, para cada classe na qual foram acrescentados novos atributos (ver a terceira coluna da tabela de mapeamento), é necessário modificar o método que contém a especificação do canvas (formulário de entrada de dados), para que ele contenha os novos widgets. Esse método é chamado de *spec* e pertence à classe que contém o formulário de interface com o usuário. Existe um editor de canvas que constrói esse método automaticamente (programação visual).

# Implementação do sistema usando o framework GREN

- **Passo 4:** Implementar novas classes/ relacionamentos não previstos pelo framework

# Implementação do sistema usando o framework GREN

- **Passo 5:** Criar a janela principal da aplicação, juntamente com os menus.



# Implementação do sistema usando o framework GREN

- **Passo 5:** Criar a janela principal da aplicação, juntamente com os menus.
- Algumas variáveis globais devem ser inicializadas na classe que contém a janela principal da aplicação, para configurar a base de dados, o nome da aplicação (para relatórios) e a linguagem.

## **initialize**

```
DatabaseName := 'videolocadora'.  
ReportHeader := 'L O C A D O R A   V Í D E O L A Z E R'.  
Language := #portuguese
```

# Implementação do sistema usando o framework GREN

## – Passo 6: Criar as tabelas no MySQL

```
cd\mysql\bin
```

```
mysqladmin create Videolocadora
```

## – Criar um script com a definição das tabelas e executar esse script:

```
cd\mysql\bin
```

```
mysql < Videolocadora.sql
```

# Implementação do sistema usando o framework GREN

## – Exemplo de script SQL (Videolocadora.sql)

```
use Videolocadora;
create table IdCode (
  className char(35),
  lastIdCode integer);
create table Filme (
  idCode integer not null,
  description char(35),
  genero integer,
  atoresPrincipais char(50),
  anoDoFilme char(4),
  precoDeLocacao float,
  diretor char(30),
  dubladoOuLegendado char(1));
```

. . .



# Implementação do sistema usando o framework GREN

## ■ Passo 7: Compile/Instale a nova aplicação

- No VisualWorks a compilação não é necessária. Utilize o “Resource Finder” para localizar a nova aplicação criada e executá-la.
- Basta ativar a janela principal da aplicação (VideolocadoraWindow) e escolher as opção Start do menu. Lembre-se que o MySQL deve estar ativado para que a aplicação seja executada satisfatoriamente.

# Validação do sistema

## ■ Processo de Validação

- Execute a nova aplicação criada na terceira etapa.
- Verifique se a janela principal oferece todas as operações de interface com o usuário, conforme os requisitos do sistema produzidos na primeira etapa.
- Tomando como base o documento de requisitos do sistema, elabore casos de teste para validar cada um dos requisitos funcionais, seguindo recomendações da engenharia de software. A etapa de testes não é tratada em detalhes aqui, mas pode ser utilizada, por exemplo, a técnica de Harrold et al. (1992) para teste incremental de estruturas de classes orientadas a objetos.
- Utilize o sistema executável para executar cada um dos casos de teste elaborados.