>>> Programação Orientada a Objetos (POO)

. . . Modelagem Básica de Sistemas com UML

Prof: André de Freitas Smaira

[~]$ _

>>> Diagrama de Classes (UML)

* Unified Modeling Language
* Usado para descrever as classes do sistema referentes ao domínio de negócio
* Linguagem padrão de projeto de software
* Análise: Não são consideradas restrições impostas pela tecnologia a ser utilizada

[~]$ _

>>> UML

* Não é uma linguagem de programação!
* É uma linguagem de modelagem
    * Requisitos
    * Comportamento
    * Estrutura lógica
    * Dinâmica de processos
    * Comunicação/Interface com os usuários

[~]$ _

>>> UML

Por que modelar um sistema?

* Sistemas são complexos; é necessário decompô-los em pedaços compreensíveis abstraindo-se aspectos essenciais
* Diagramas auxiliam no entendimento
* Entender quais objetos fazem parte do sistema e como se comunicam
* O modelo induz ao projeto: previsão das necessidades, problemas e limitações

[~]$ _

>>> UML - Vantagens e Desvantagens

Por que modelar um sistema?

- Custos
  - **Maior trabalho** na modelagem
  - **Mais tempo** gasto
- Ganhos
  - **Menos trabalho** na construção
  - **Problemas encontrados em tempo hábil** para sua solução
  - **Dúvidas sanadas** mais cedo

>>> Diagramas da UML

* Diagrama de Classes
* Diagrama de Objetos
* Diagrama de Componentes
* Diagrama de Casos e usos
* Diagrama de Sequência
* Diagrama de Colaboração
* Diagrama de Estado
* Diagrama de Atividades

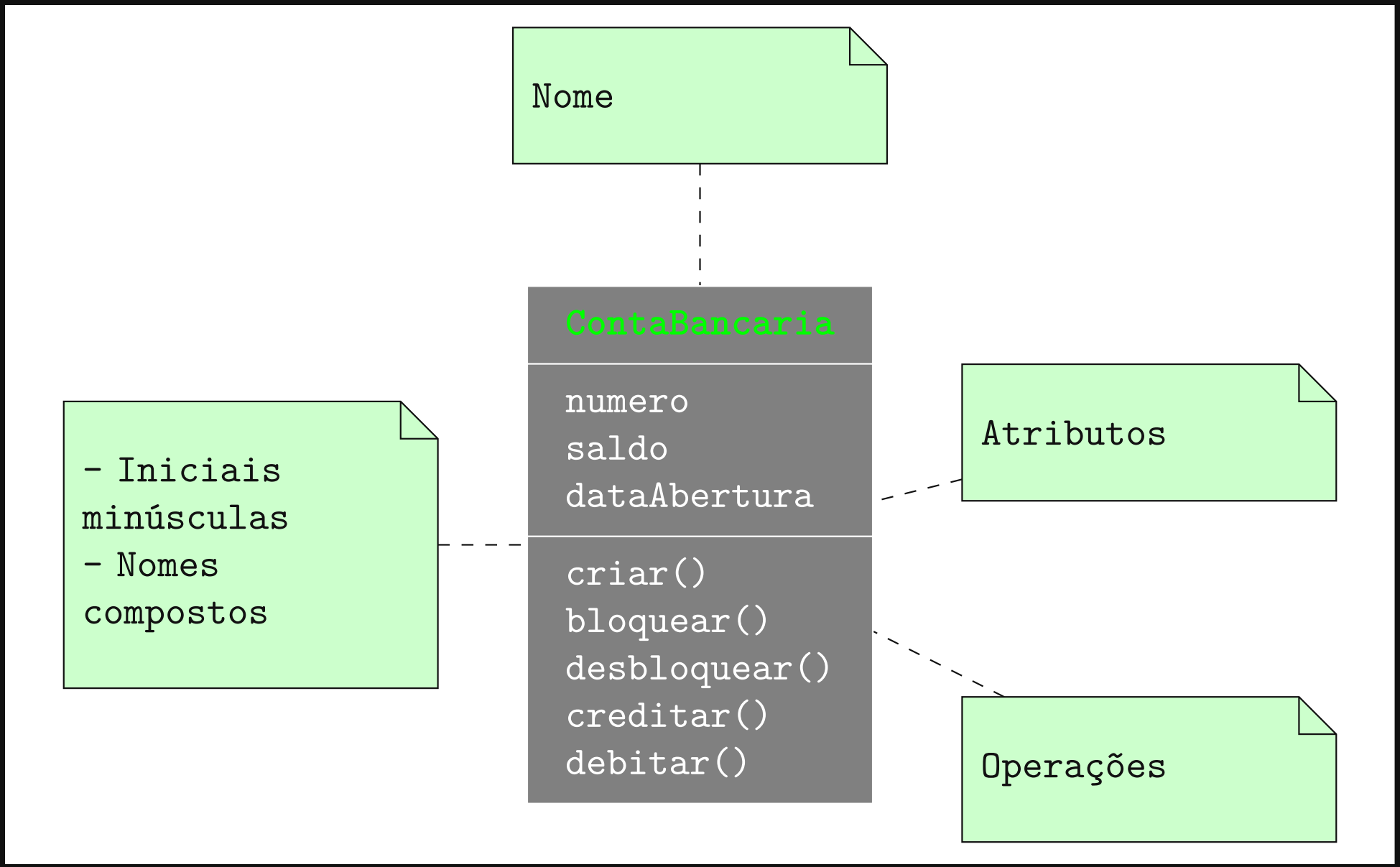
[~]$ _

>>> Diagrama de Classes

* O mais utilizado
* Objetivo: visualização das classes com atributos e métodos, bem como relacionamentos entre classes
* Visão estática do sistema
* Estrutura lógica

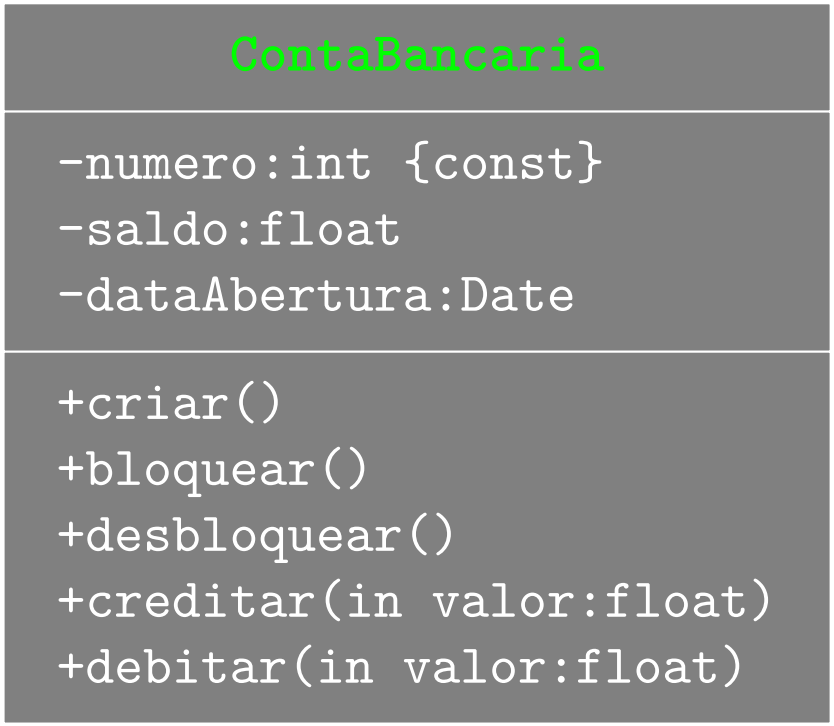
[~]$ _

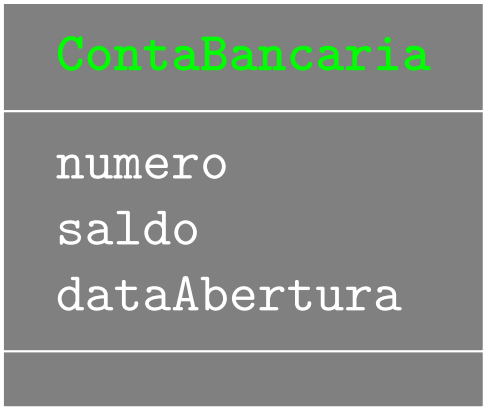
>>> Classes

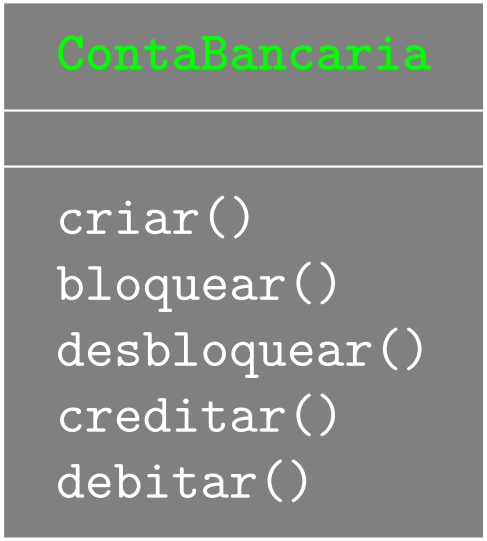

Nome
ContaBancaria
numero
saldo
dataAbertura
criar()
bloquear()
desbloquear()
creditar()
debitar()
- Iniciais minúsculas
- Nomes compostos
Atributos
Operações

[~]$ _

>>> Classes - Diferentes graus de abstração







[~]$ _

>>> Sintaxe

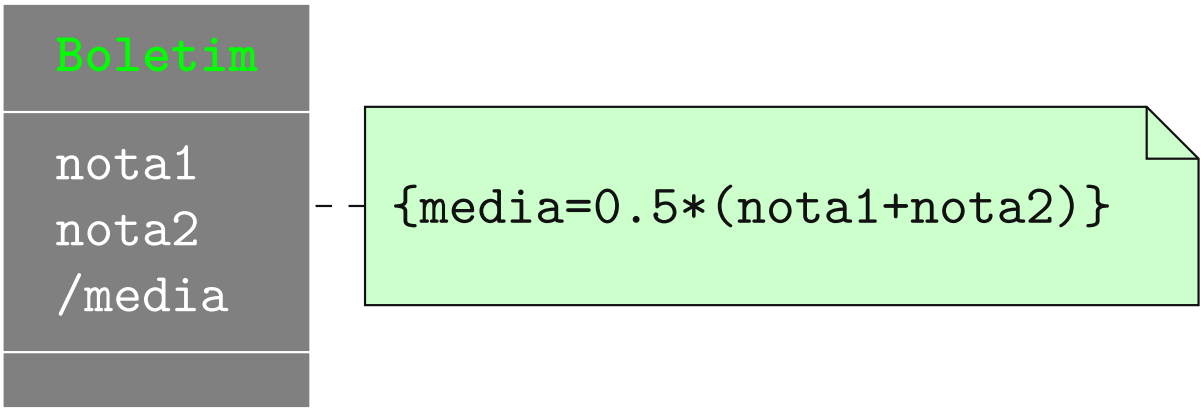
* **Atributos**
  visibilidade nome:tipo[multiplicidade] {propriedades} = valor_inicial
* **Operações**
  visibilidade nome(parametros):tipo_de_retorno {propriedades}

>>> Visibilidade

Visibilidade	Significado
+	**Pública**: visto e usado por qualquer objeto que tenha referência para a classe
#	**Protegida**: visto e usado apenas dentro da classe onde foi declarado e suas descendentes
_	**Privada**: visto e usado apenas dentro da classe onde foi declarada

[~]$ _

>>> Atributo derivado



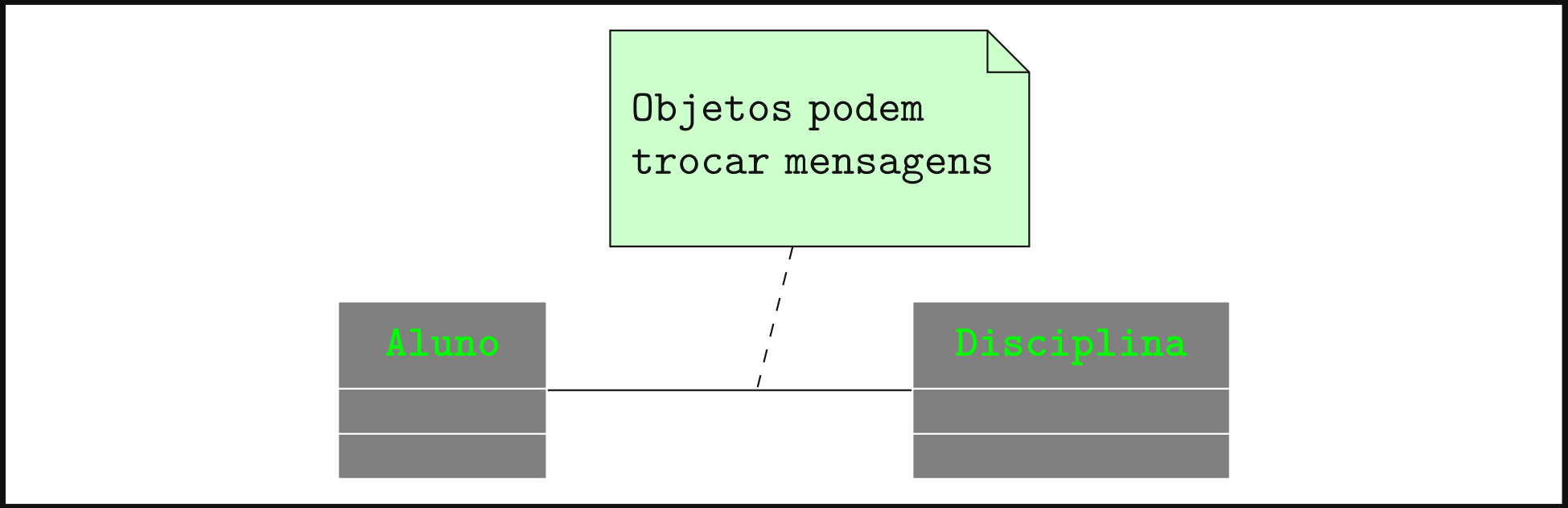
[~]$ _

>>> Relacionamentos

* Possibilita troca de mensagens entre objetos
* Possibilita que os objetos colaborem entre si
* Associação
    * Conecta duas classes, demonstrando colaboração entre instâncias delas
    * Agregação
    * Composição
* Herança

[~]$ _

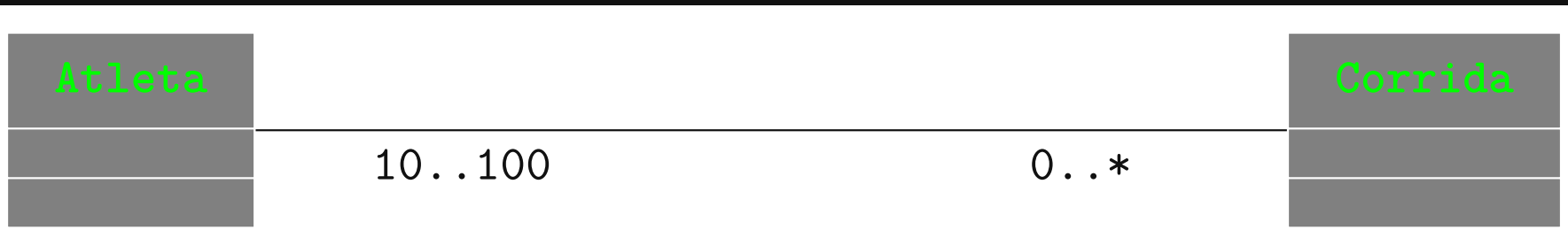
>>> Associação



[~]$ _

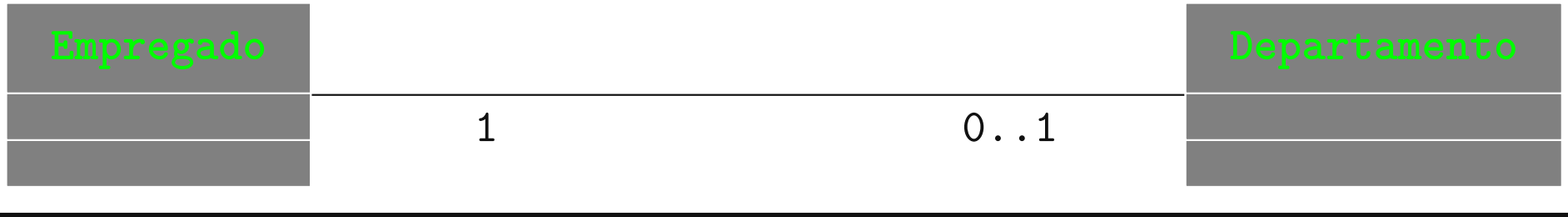
>>> Multiplicidade

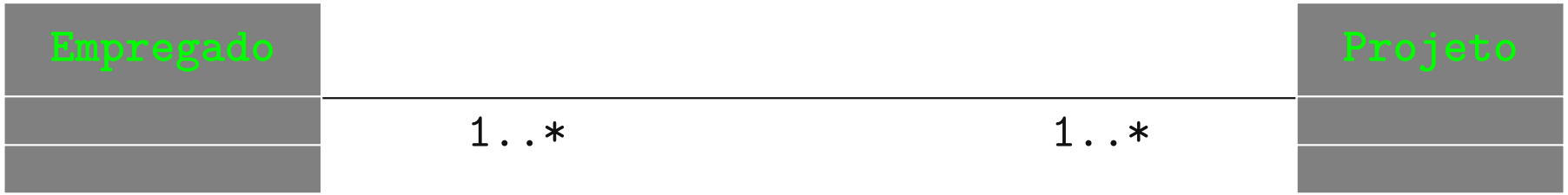
Símbolo	Significado
1 ou 1..1	Exatamente 1
* ou 0..*	0 ou muitos
1..*	1 ou muitos
0..1	0 ou 1
a..b	Intervalo específico



[~]$ _

>>> Multiplicidade



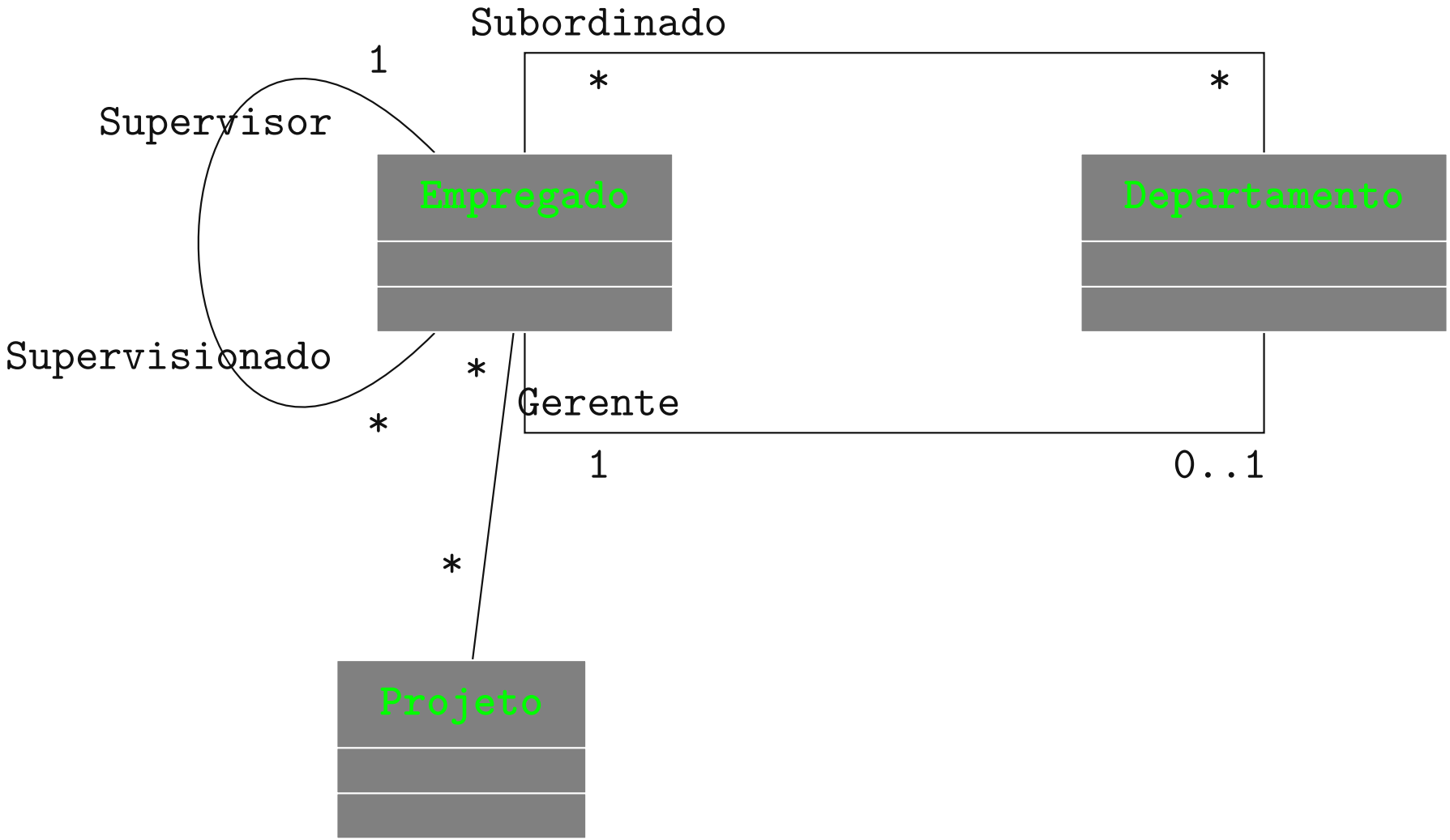


[~]$ _

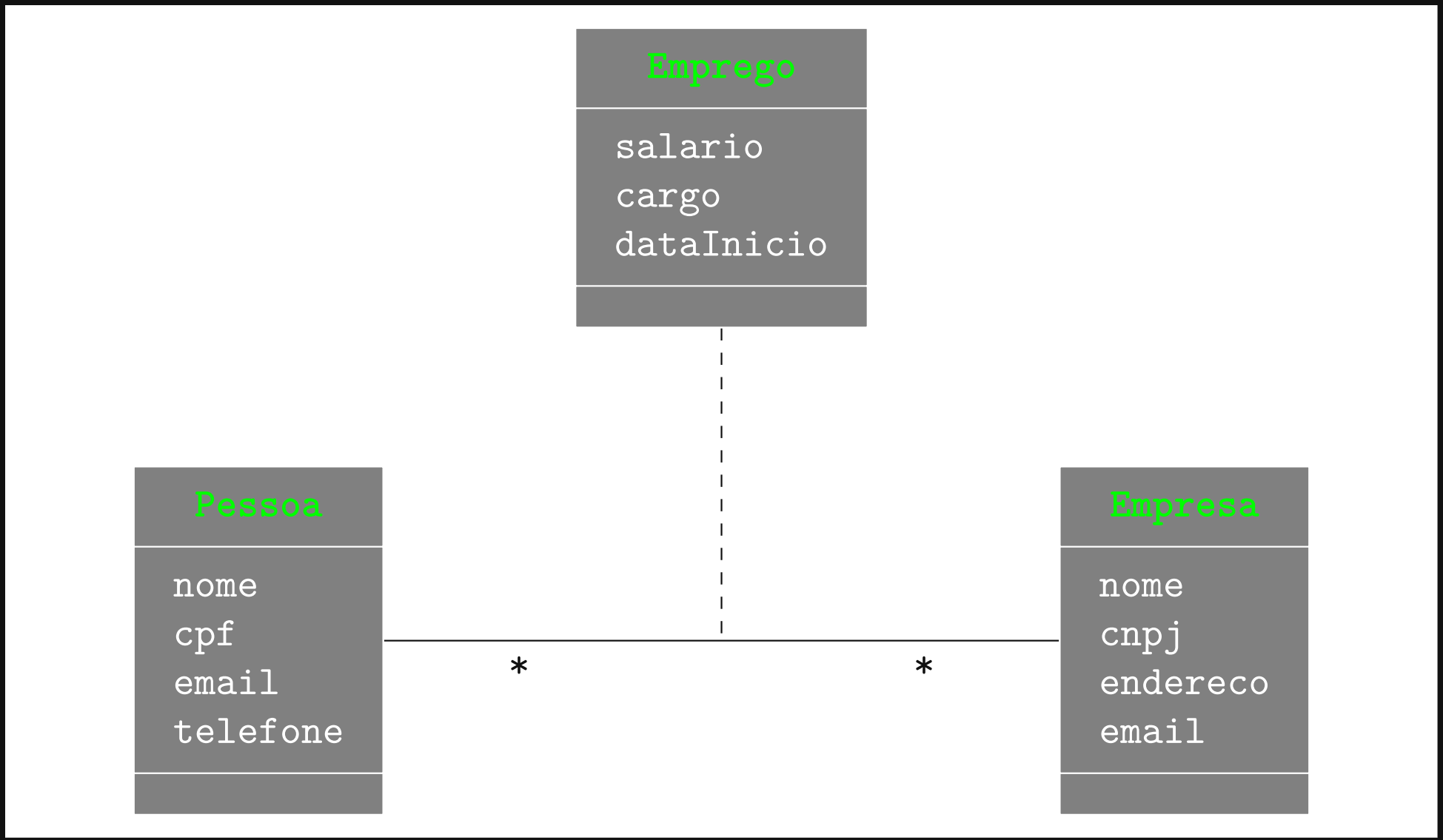
>>> Papéis (roles)

[~]$ _

>>> Múltiplas associações

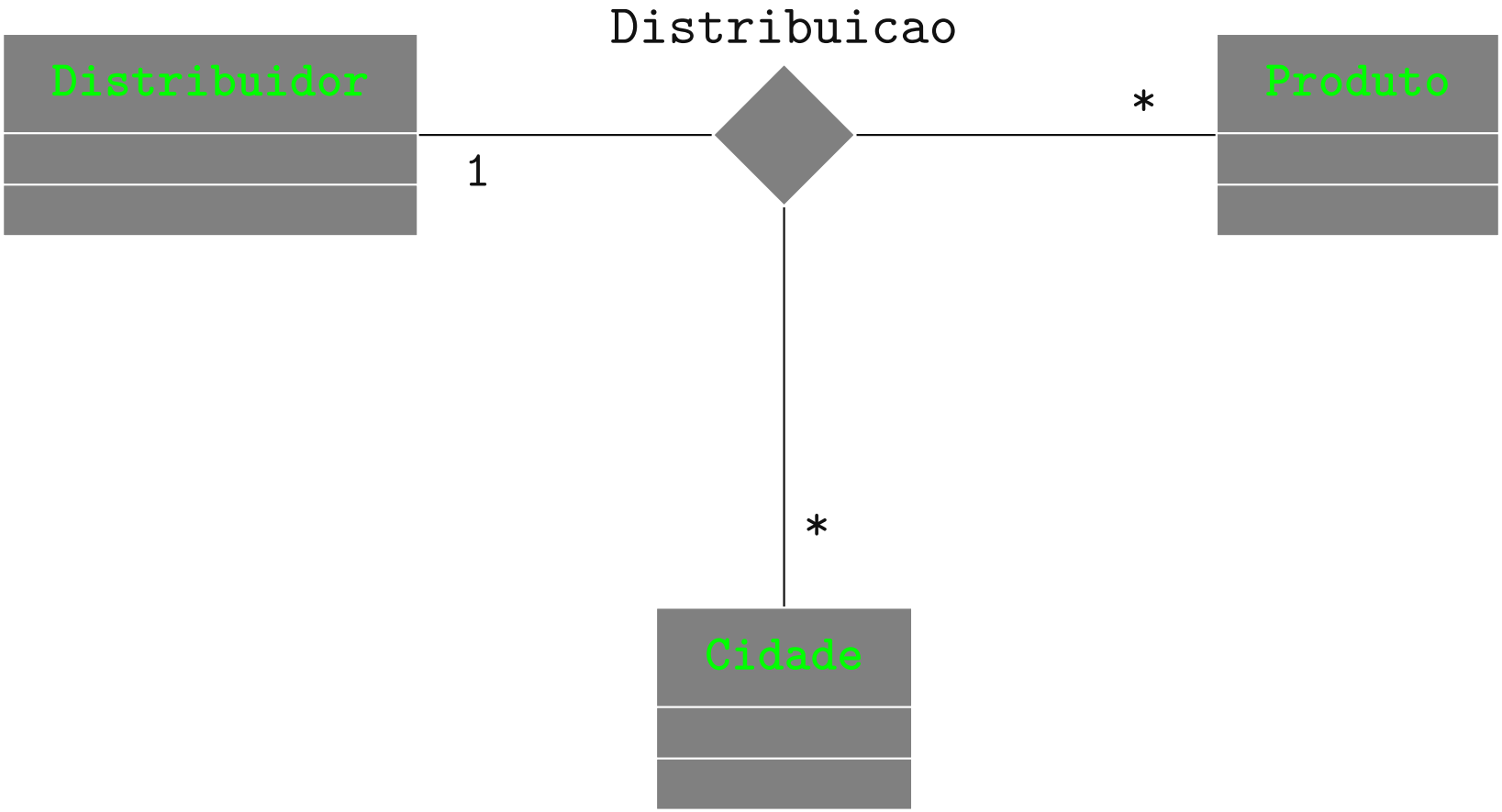


>>> Classes associativas



[~]$ _

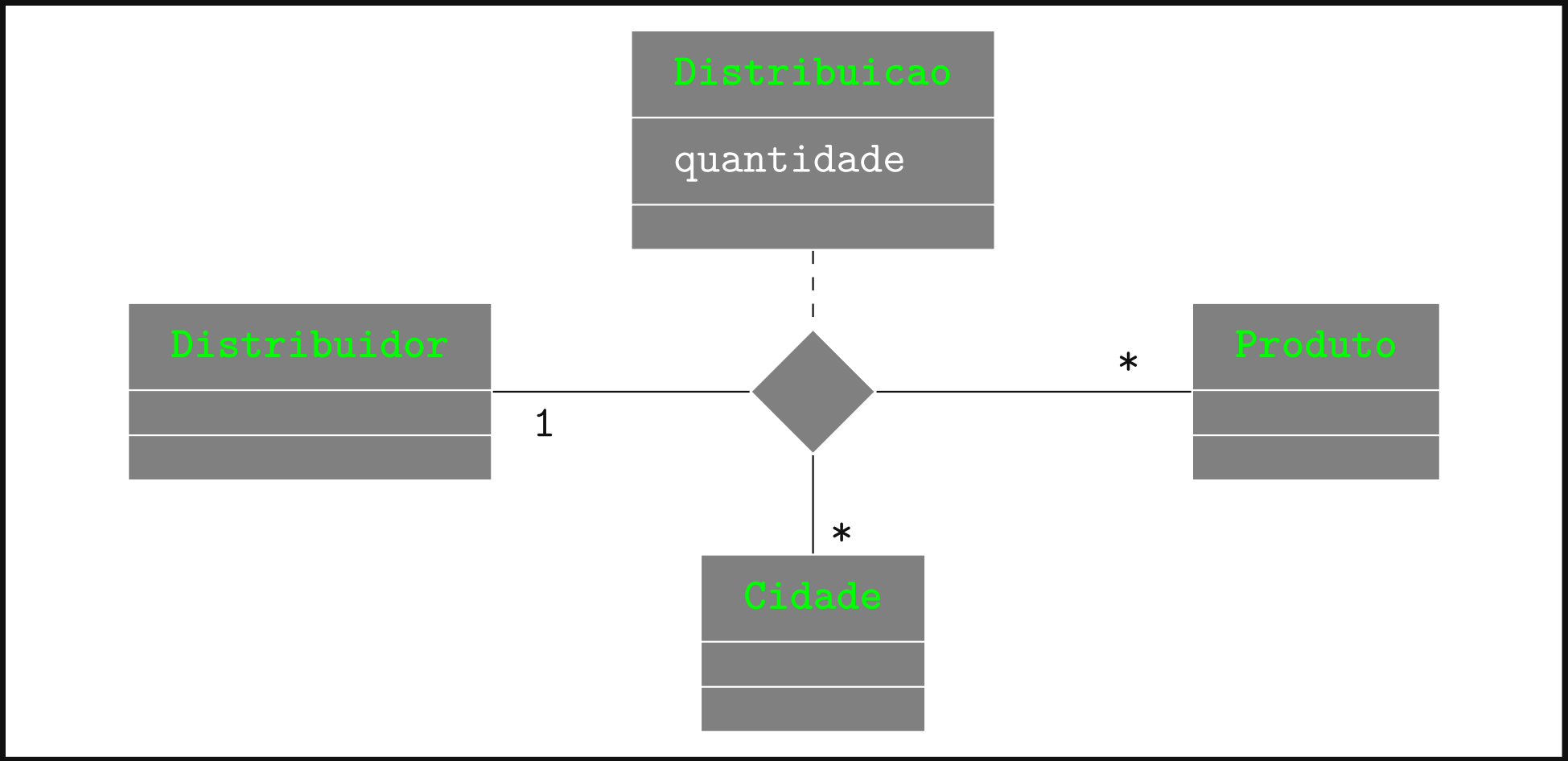
>>> Associação ternária



[~]$ _

>>> Associação ternária

Pode ser uma classe associativa



[~]$ _

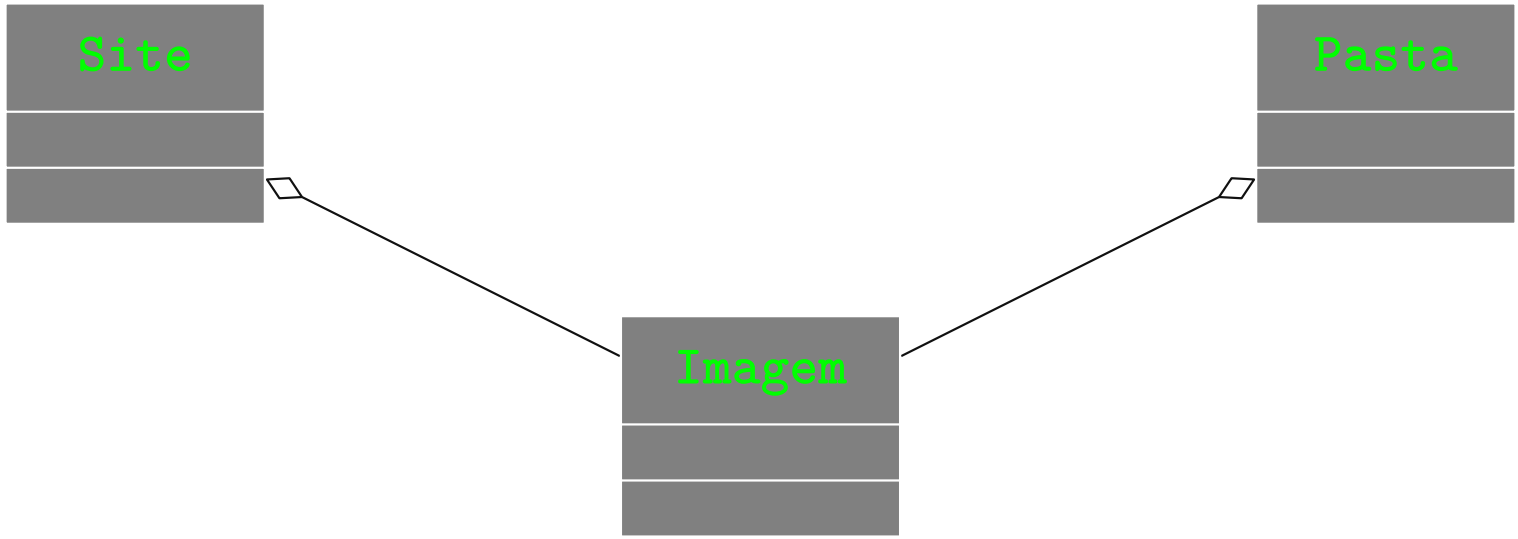
>>> Agregações e Composições

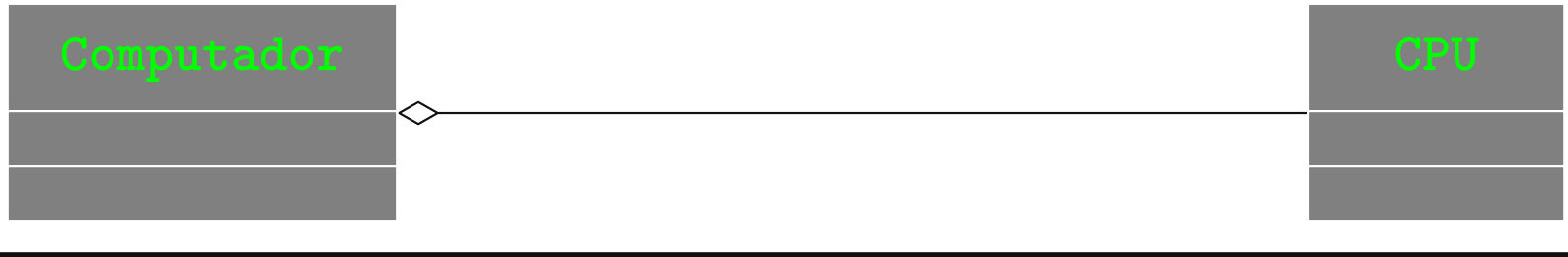
* TODO-PARTE
* **Assimétricas**: se A é parte de B, então B não é parte de A
* **Propagam comportamento**: se um comportamento se aplica ao TODO, também se aplica às PARTES

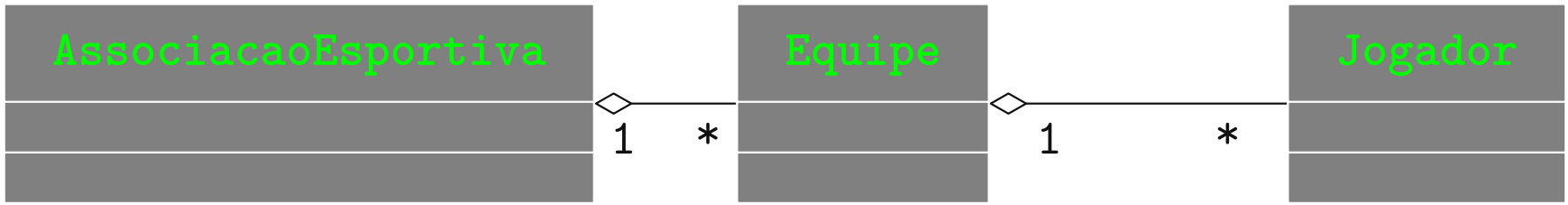
[~]$ _

>>> Agregações - Exemplo

Relacionamento fraco - a parte não depende do todo

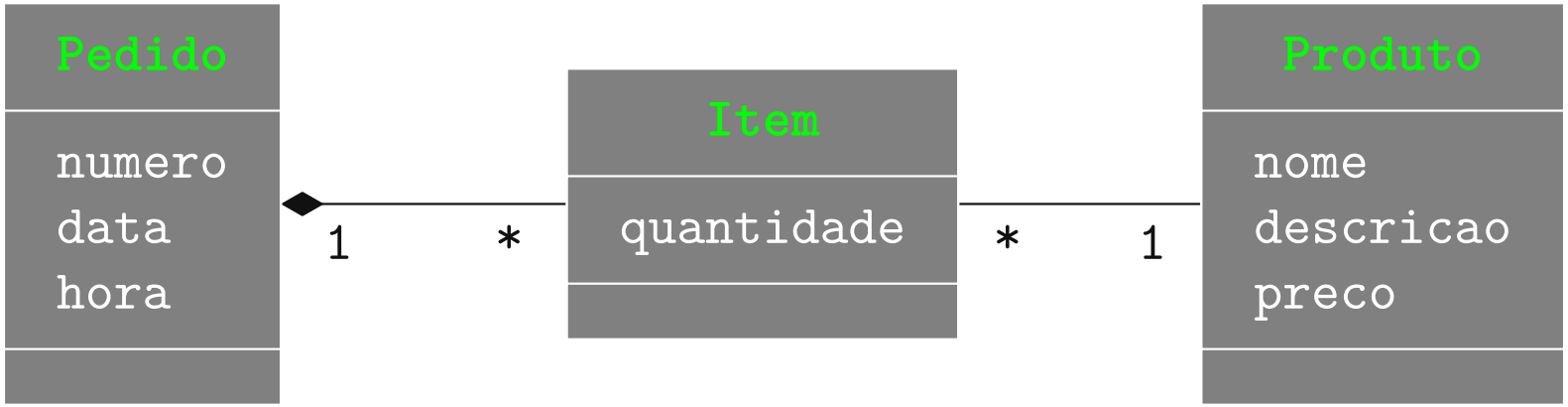






>>> Composições - Exemplo

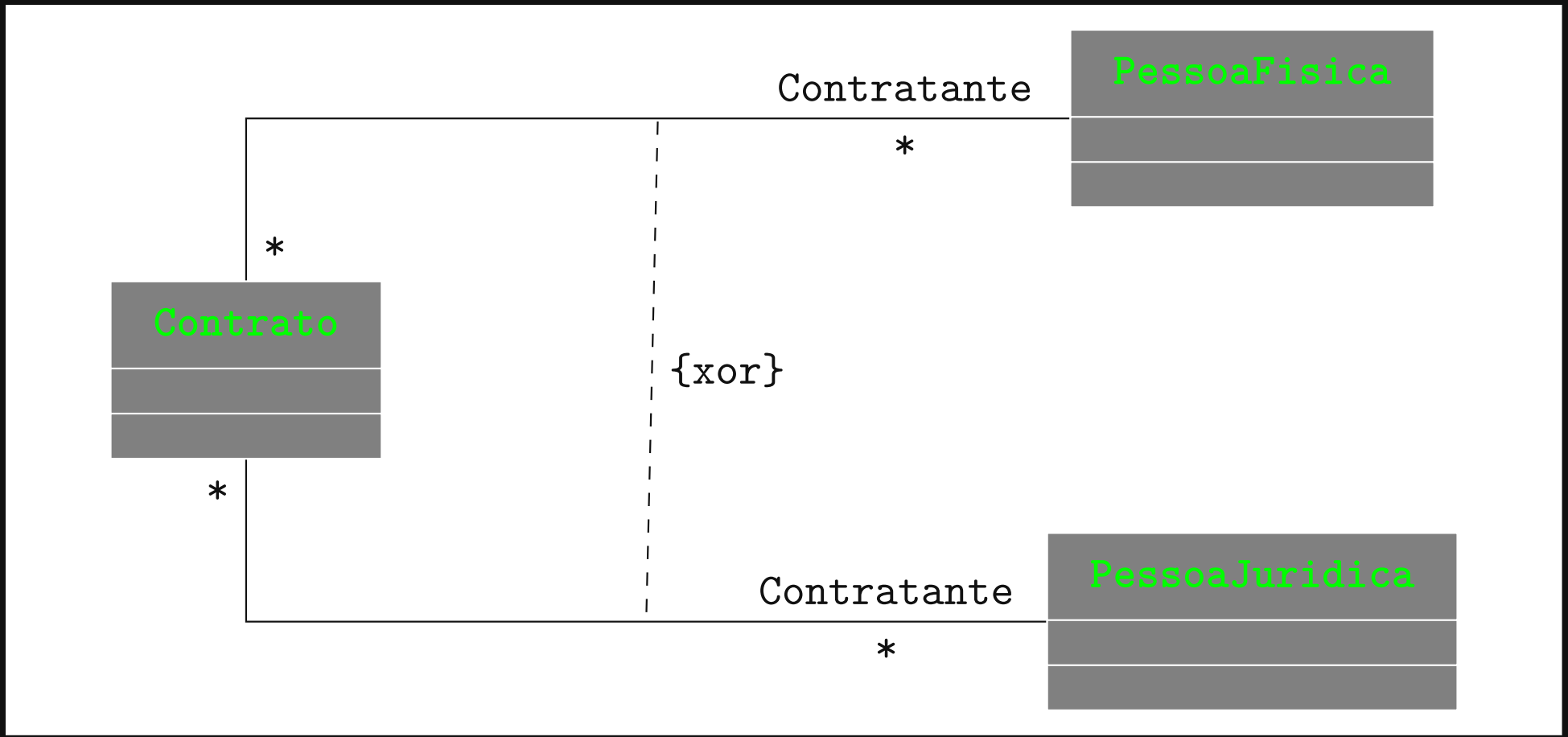
Relacionamento forte - a parte depende do todo



[~]$ _

>>> Associações - Restrições

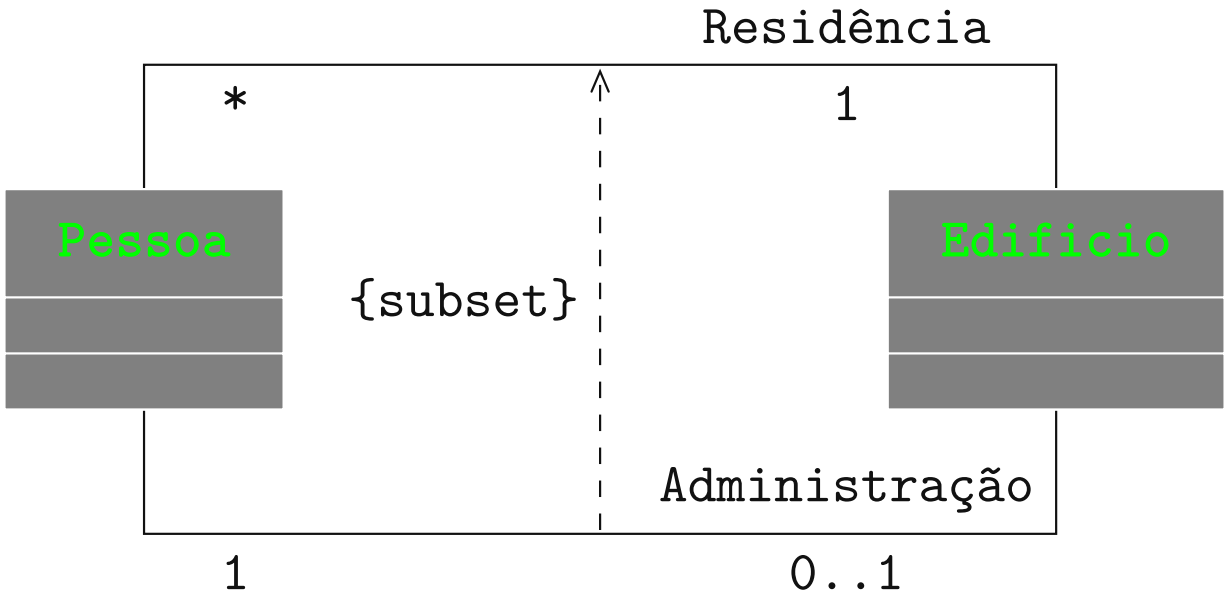
* **Associação XOR (exclusiva)**: Somente uma dentre todas pode ocorrer



[~]$ _

>>> Associações - Restrições

* **Associação Subset**: Objetos conectados por uma associação são subconjunto dos objetos conectados por outra



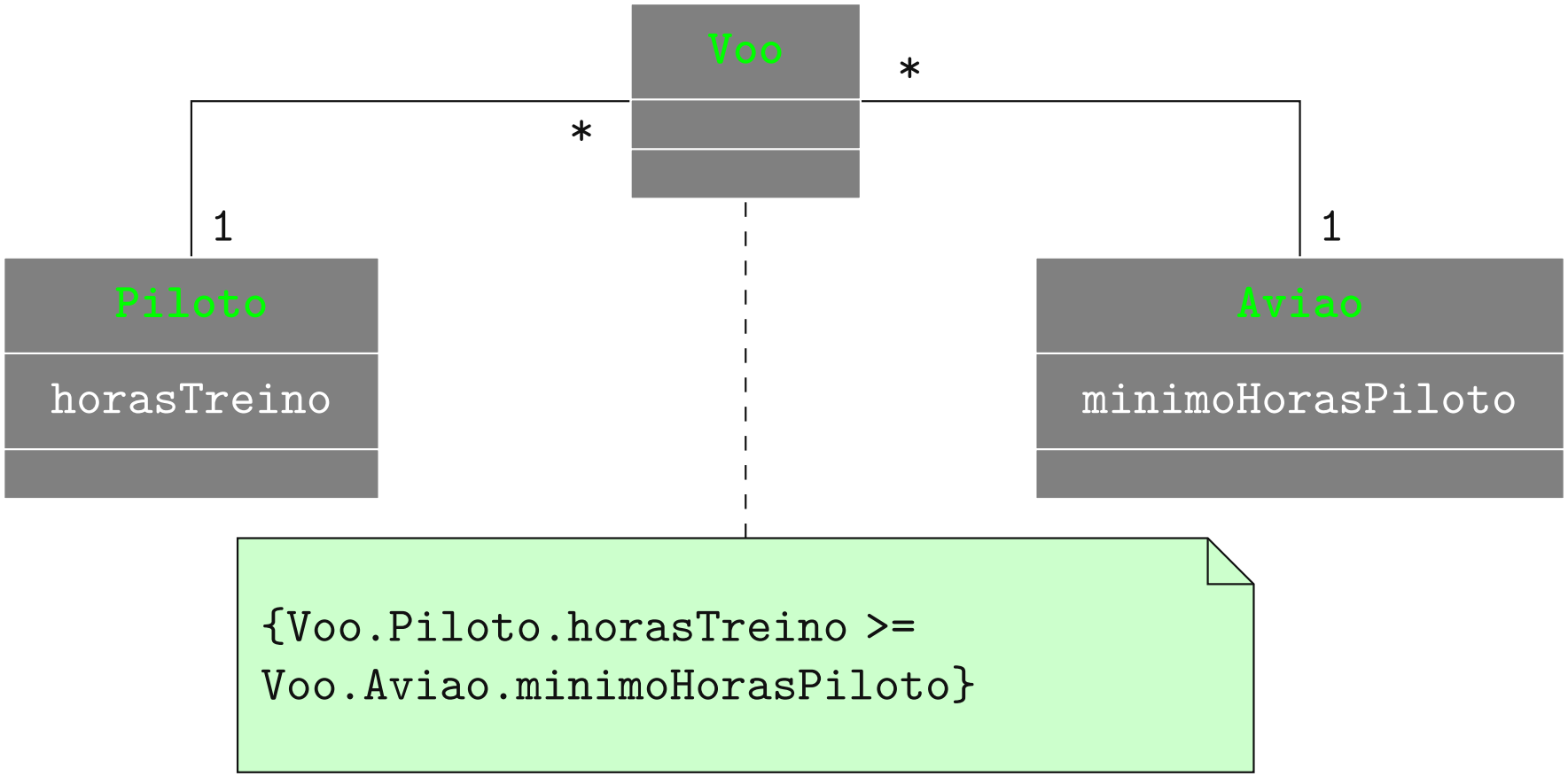
[~]$ _

>>> Outras restrições

* OCL - Object Constraint Language
    * Expressões podem fazer uso de propriedades de uma classe (atributos, operações e associações), operadores aritméticos (+, -, *, /) e lógicos (=, >, <, …)

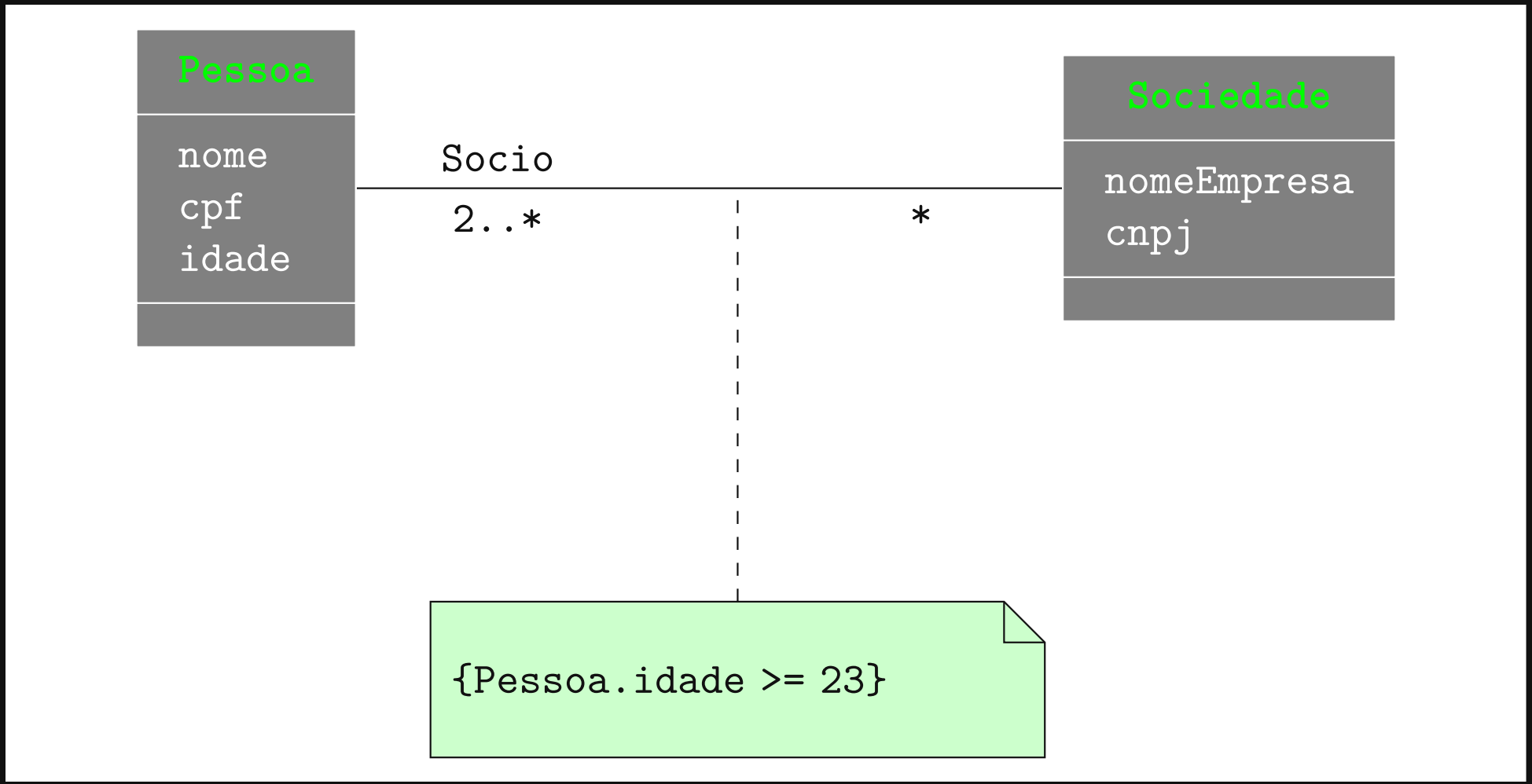
[~]$ _

>>> Outras restrições - Exemplo



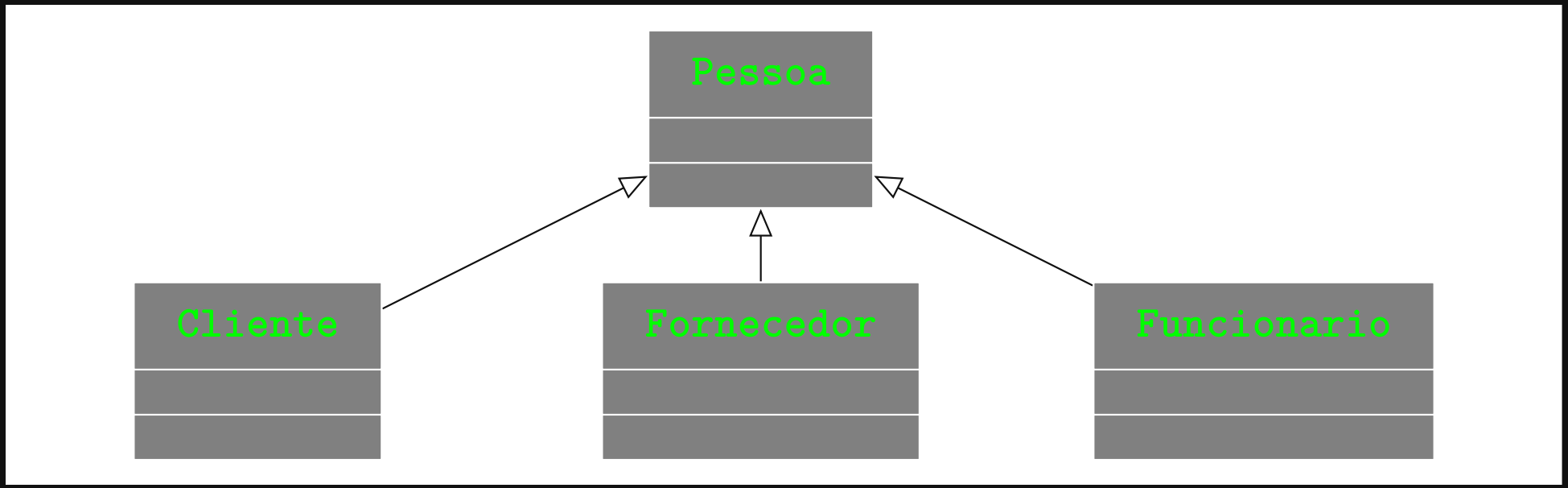
[~]$ _

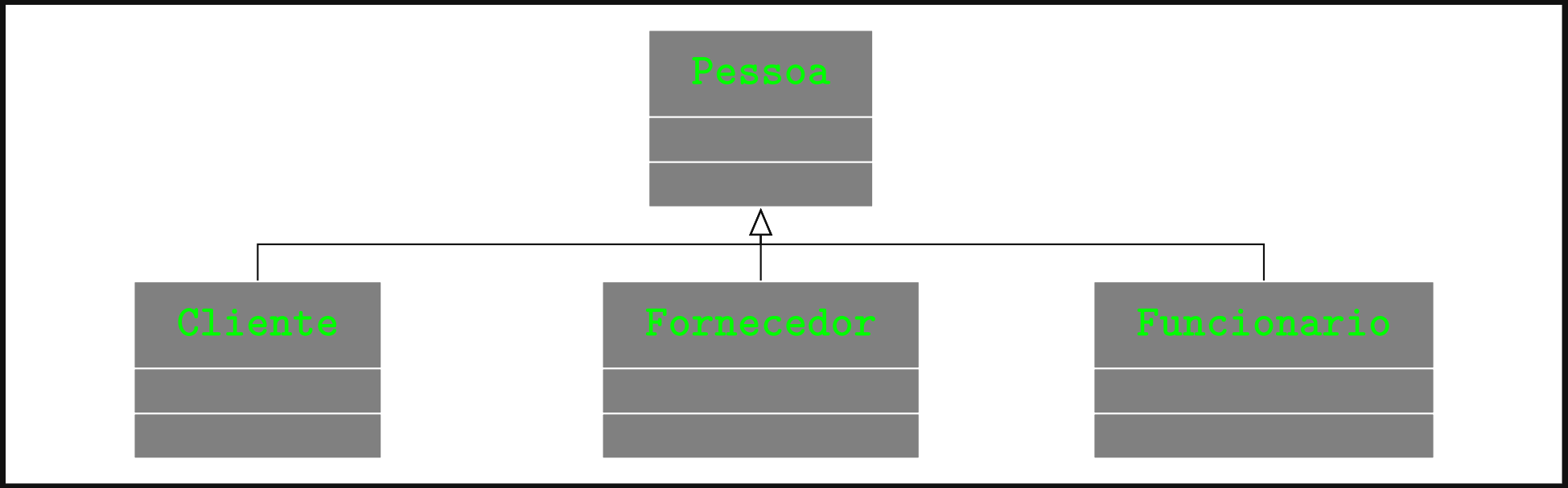
>>> Outras restrições - Exemplo



[~]$ _

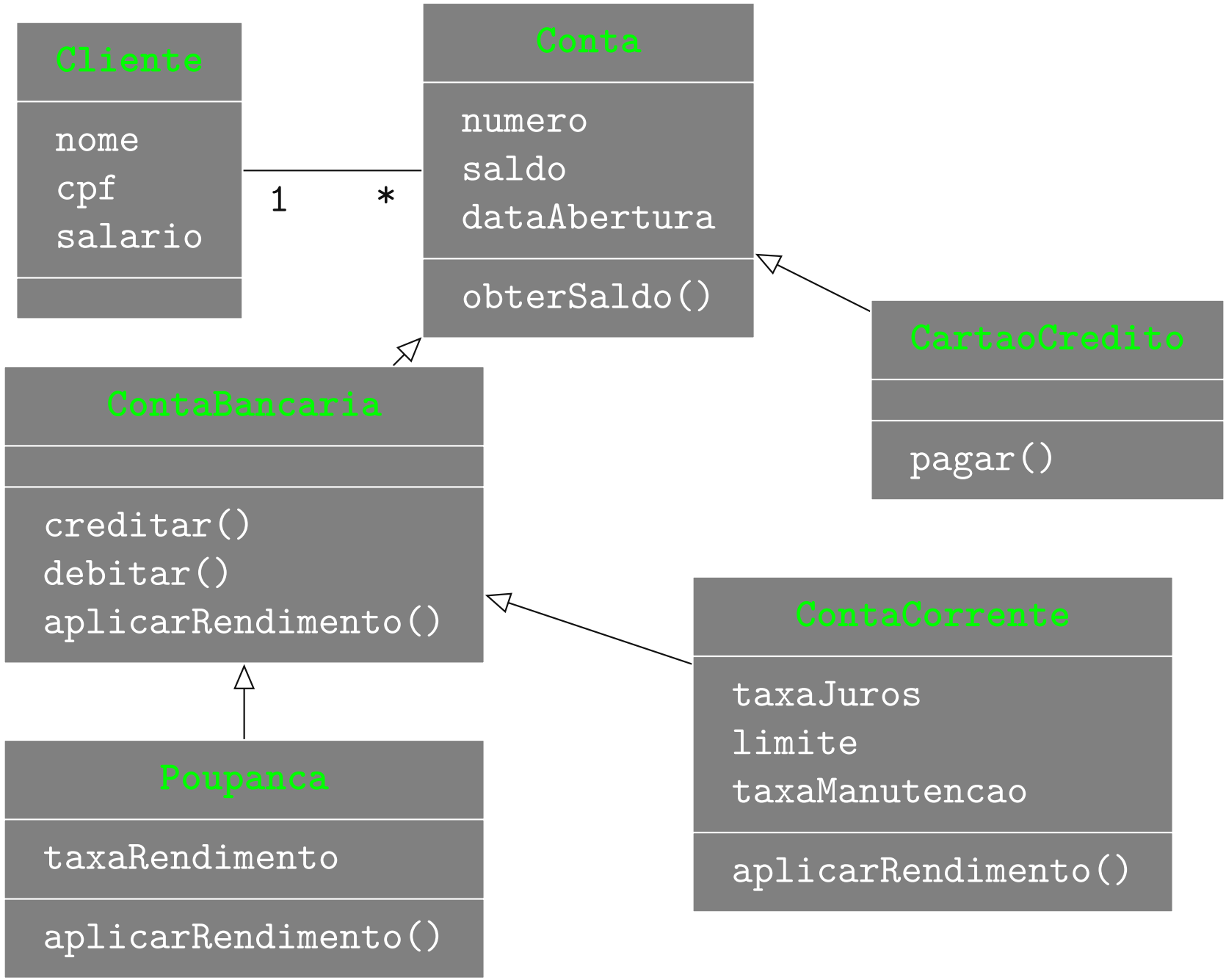
>>> Generalização/Especialização - Herança





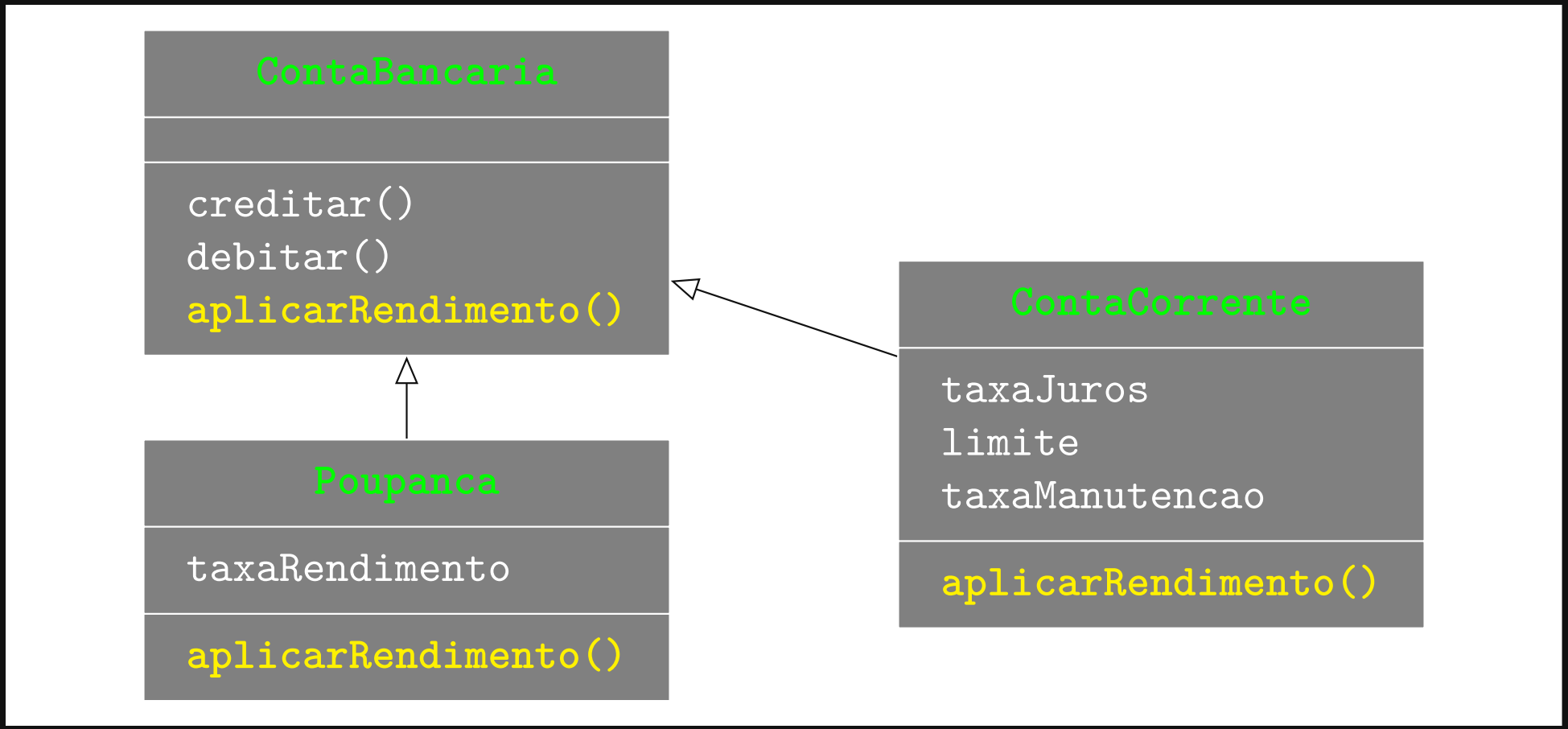
[~]$ _

>>> Outras restrições - Exemplo

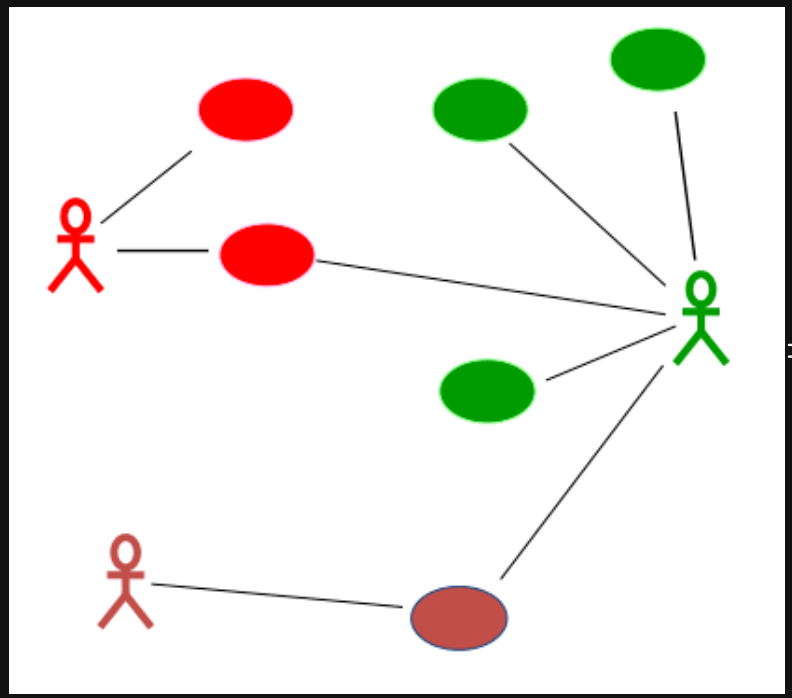


>>> Métodos abstratas

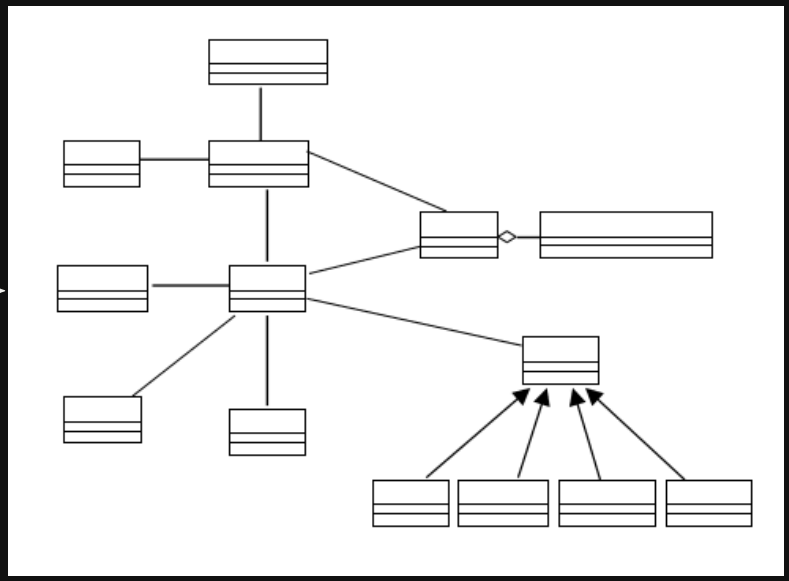
* Organiza características comuns
* Define apenas forma de operações



>>> Classes e Casos de Uso



⇒



[~]$ _

```
>>> Exemplo completo
Concurso
```

>>> Caso de Uso - Cadastrar Concurso

Descrição: Cadastro de um novo concurso e preparo do sistema para as inscrições

Ator: Empresa de Concursos (Usuário)

Condição Anterior: Usuário logado e já na opção de inclusão de novo concurso

Fluxo principal:

1. **Usuário** informa o contratante do concurso
2. **Usuário** cadastra **número do edital** e **período de inscrição**
3. Para cada **cargo**, o **usuário** cadsatra o **nome**, a **taxa de inscrição** e o **número de vagas**.
4. Para cada **cargo**, o **usuário** deve cadastrar uma lista de provas de caráter **classificatório** e/ou **eliminatório**, informando **tipo** (objetiva, discursiva, prática), **área de conhecimento**, **peso** e **data**.
5. **Usuário** deve cadastrar a ordem de desempate das provas
6. **Usuário** deve cadastrar o **percentual mínimo de acertos** para classificação

[~]$ _

>>> Caso de Uso - Cadastrar Concurso

7. Sistema cadastra os dados
8. Sistema atualiza estado do concurso para "aguardando inscrições"

Condição posterior: Concurso cadastrado

[~]$ _

>>> Caso de Uso - Inscrever em Concurso

Descrição: Cadastro de inscrição de candidatos
Ator: Candidato (Usuário)
Condição Anterior: Chamda de listar concursos em execução
Fluxo principal:

1. **Sistema** lista os concursos com inscrição aberta
2. **Usuário** seleciona o concurso desejado
3. **Sistema** lista os cargos oferecidos no concurso
4. **Usuário** informa seu **nome**, **endereço** (logradouro, número, complemento, bairro, CEP, cidade e UF), **telefones**, **e-mail**, **data de nascimento**, **sexo**, **documento**, **cargo**, se tem e qual **deficiência física**, todos obrigatórios.
5. **Sistema** cadastra e gera o número de inscrição

Condição posterior: Lista de candidatos atualizada, candidato inscrito e número de incrição gerado e fornecido.

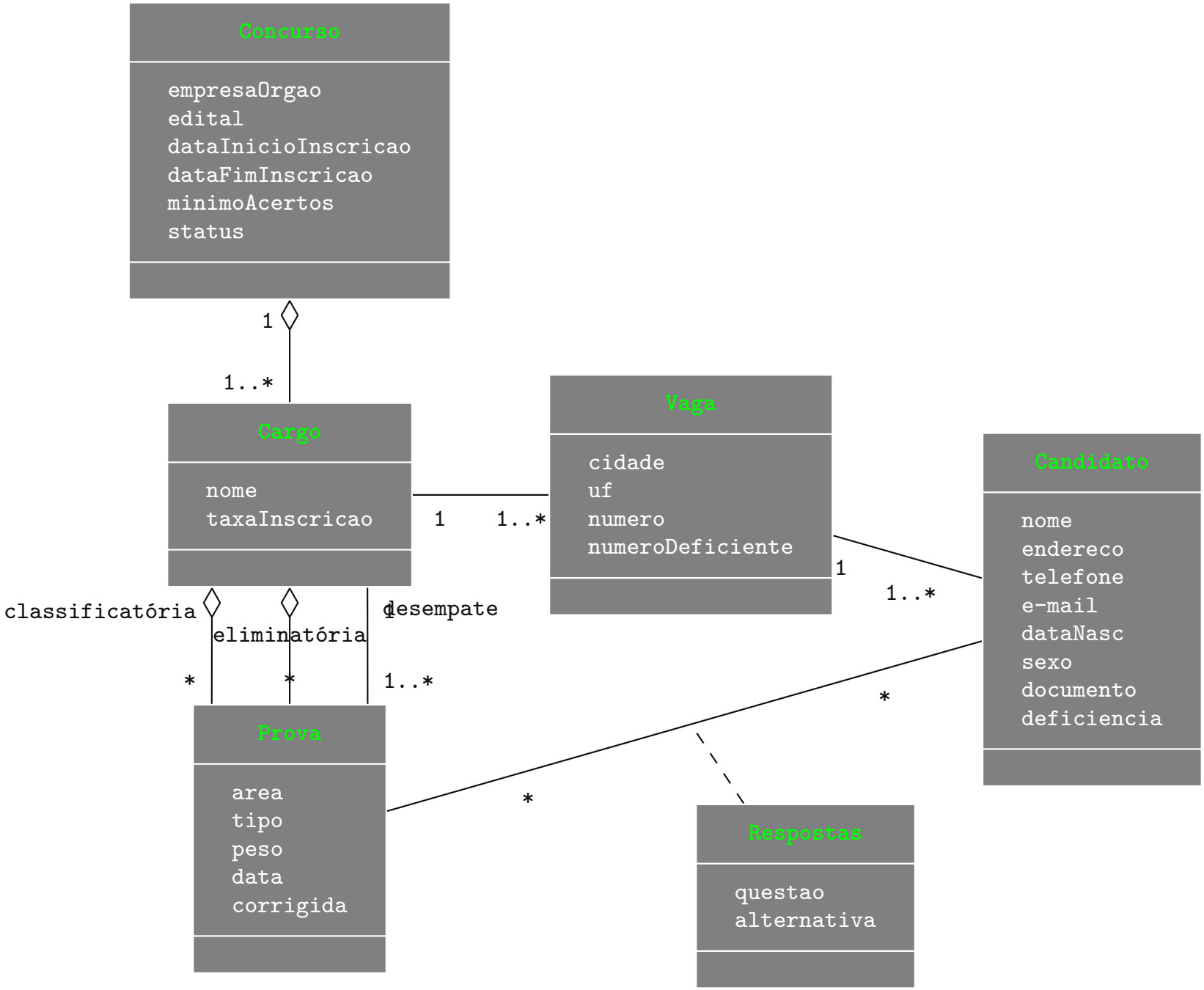
[~]$ _

>>> Classes

Possíveis Classes	Atributos
Concurso	EmpresaOrgao, edital, dataInicioInscricao, dataFimInscricao, minimoAcertos, cargos, status
Cargo	nome, taxaInscricao, cidade, UF, listaClassificatorias, listaEliminatorias, …
Prova	tipo, data, corrigida, …
…	…
Candidato	nome, enderecoLogradouro, enderecoNumero, …, deficiencia

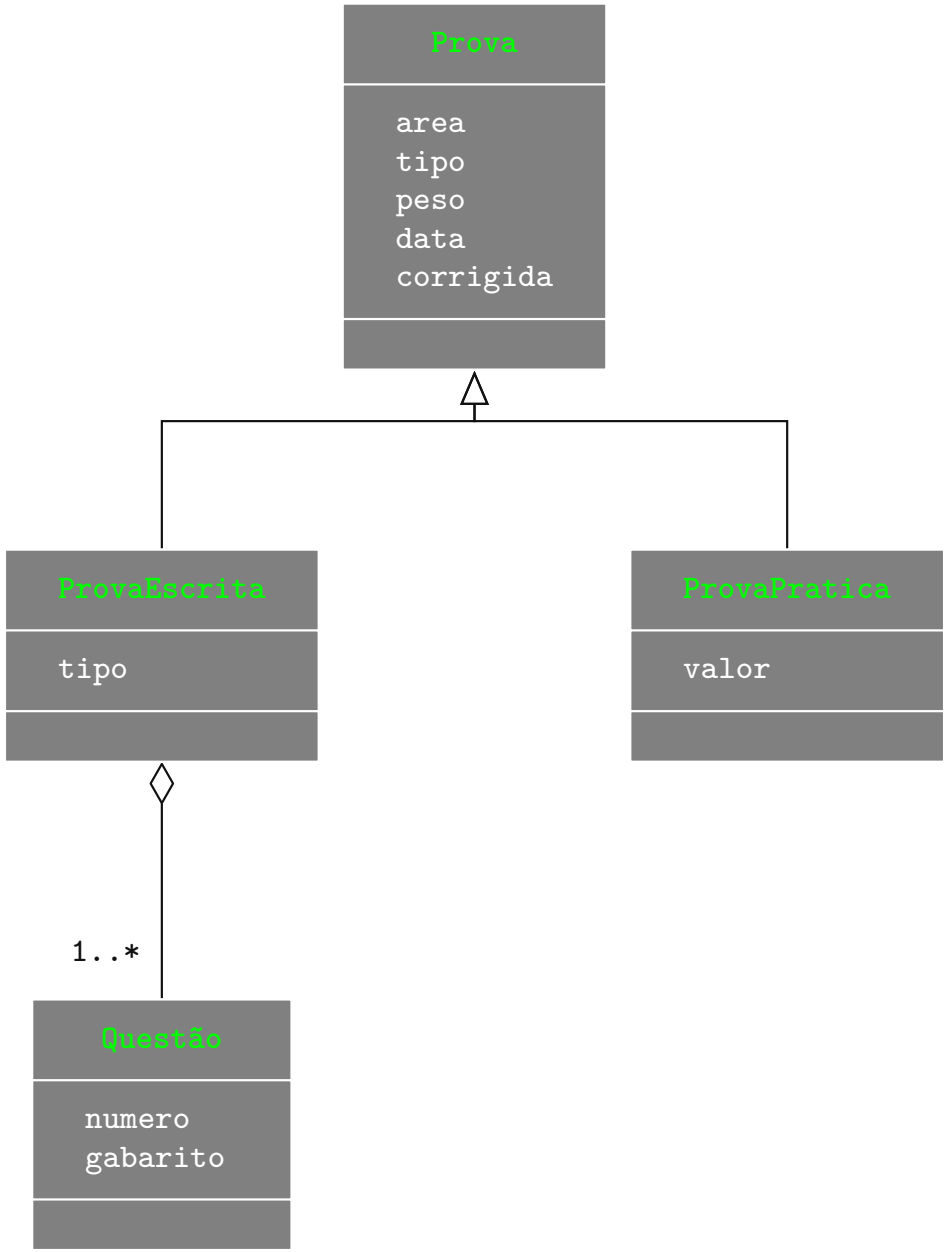
[~]$ _

>>> Classes


Concurso
empresaOrgao
edital
dataInicioInscricao
dataFimInscricao
minimoAcertos
status
1
1..*
Cargo
nome
taxaInscricao
1
1..*
Vaga
cidade
uf
numero
numeroDeficiente
1
1..*
Candidato
nome
endereco
telefone
e-mail
dataNasc
sexo
documento
deficiencia
classificatória
eliminatória
desempate
1
*
*
1..*
Prova
area
tipo
peso
data
corrigida
*
*
Respostas
questao
alternativa

[~]$ _

>>> Classes


Prova
area
tipo
peso
data
corrigida
ProvaEscrita
tipo
ProvaPratica
valor
1..*
Questão
numero
gabarito

[~]$ _

>>> Categorias de Objetos

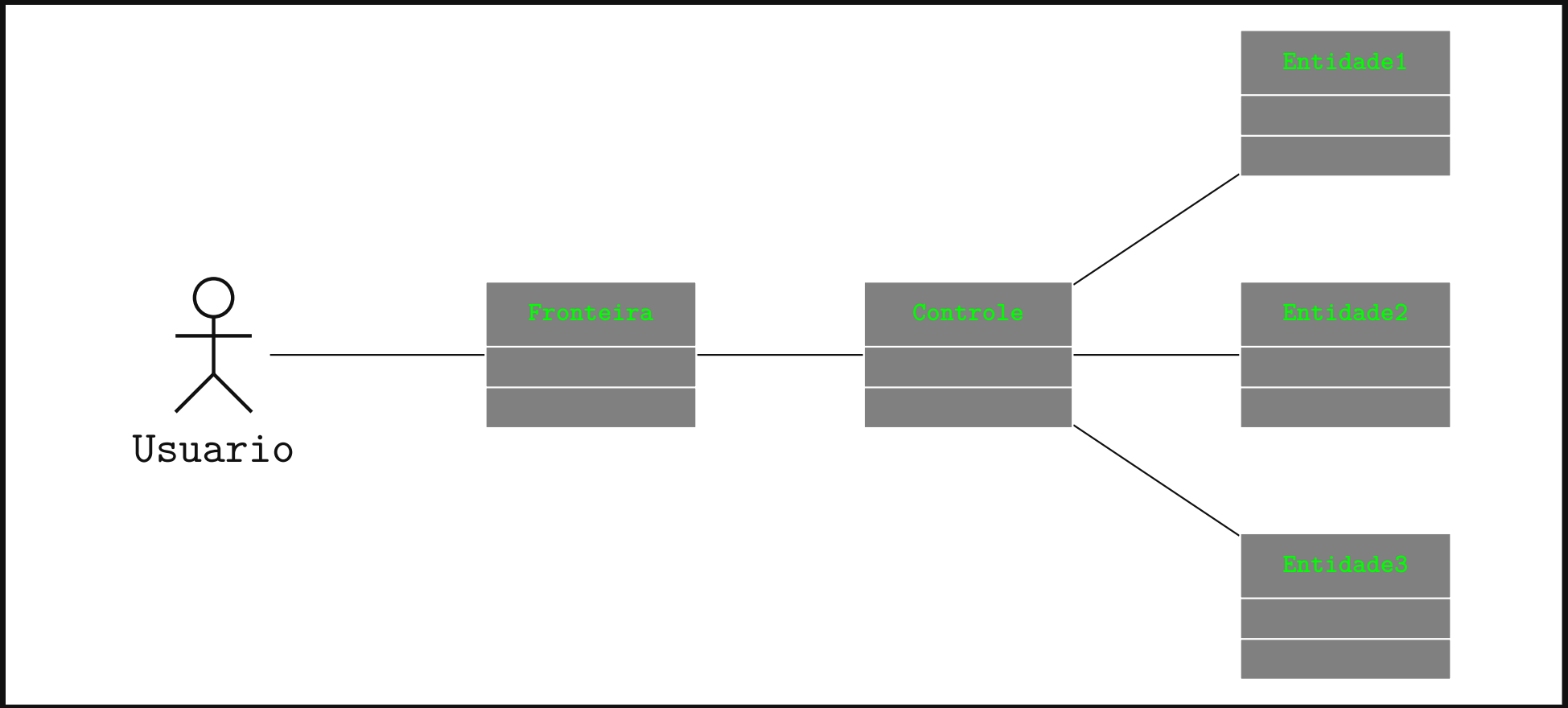
[~]$ _

>>> Categorias de Objetos - Proposta de Jacobson

* Cada objeto é **especialista** em um tipo de ação
* **Divisão de tarefas** - capacidade de adaptação a mudanças
* **Objetos de entidade**: manutenção de informações
* **Objetos de controle**: coordenação de caso de uso
* **Objetos de fronteira**: comunicação com atores

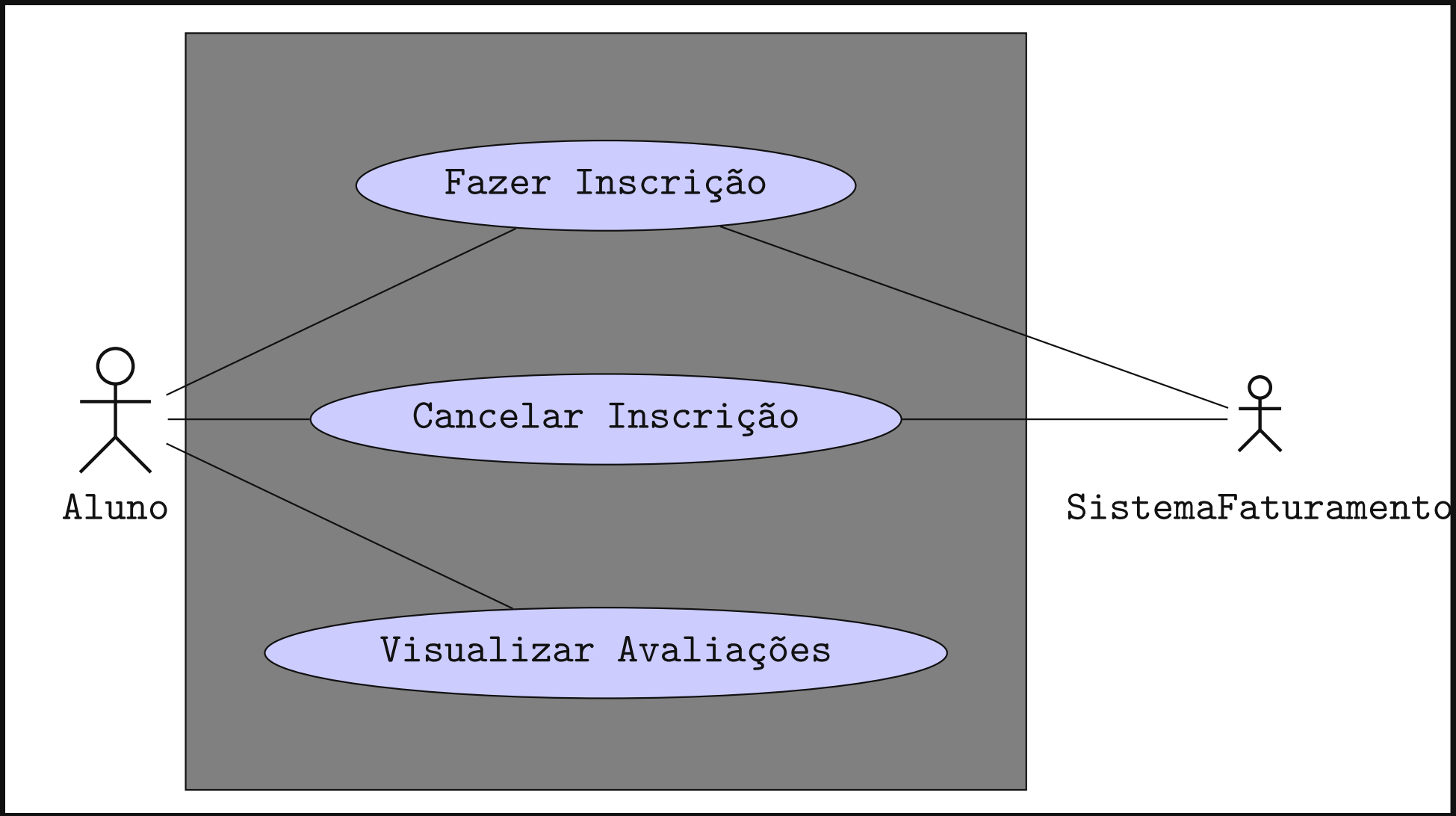
[~]$ _

>>> Categorias de Objetos - Proposta de Jacobson



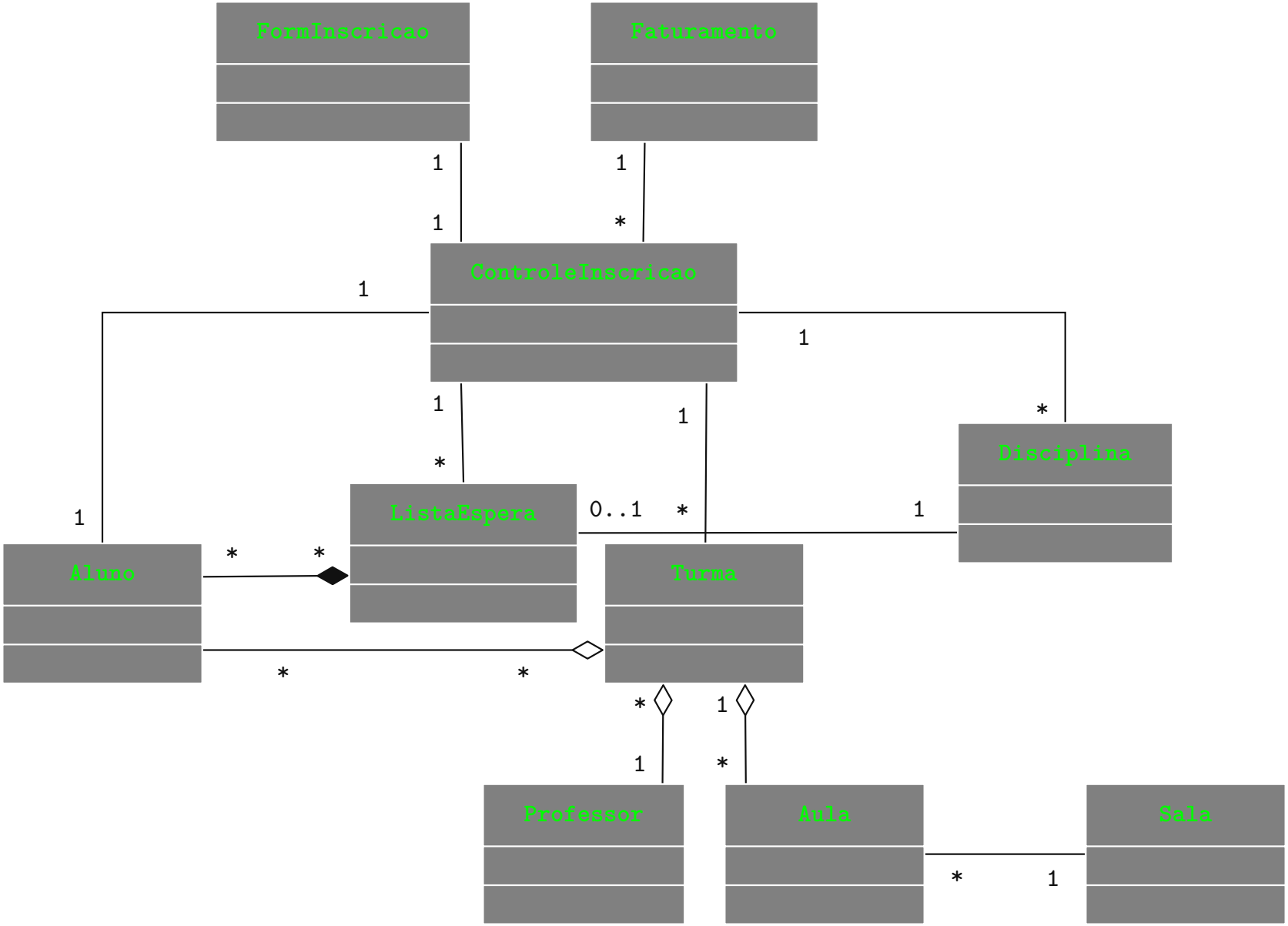
[~]$ _

>>> Exemplo - Realizar Inscrção



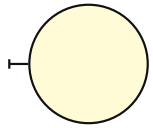
[~]$ _

>>> Categorias de Objetos - Proposta de Jacobson

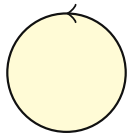


[~]$ _

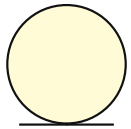
>>> Notação Simplificada



FormInscricao:classeDeFronteira



ControleInscricao:classeDeControle



Aluno:classeDeEntidade