

# Sobre a aula de hoje

- 0 Objetivo da aula: apresentar conteúdos segundo o plano de ensino sobre conceitos e mecanismos da POO.
- 0 Metodologia: Aula expositiva, com recursos de imagem, exemplos e trabalhos práticos em aula
- 0 Textos base:
  - 0 DEITEL, Harvey M.; DEITEL, Paul J. Java como programar. Porto Alegre: Bookman, 2001. 1202p.

# Onde estamos?

Introdução ao desenvolvimento de sistemas reusáveis de software



Conceitos e mecanismos da programação orientada a objetos

Objetos e classes

Herança e  
Polimorfismo

Classes abstratas  
e interfaces

Diagrama de  
classes

Bibliotecas de  
classes

Tipos genéricos



Técnicas de uso comum em sistemas orientados a objetos

Interface gráfica com o  
usuário

Tratamento de exceções

Coleções

Persistencia de dados e  
objetos (serialização)



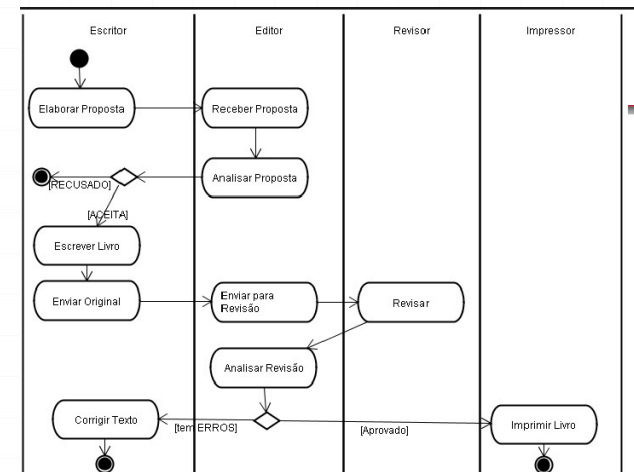
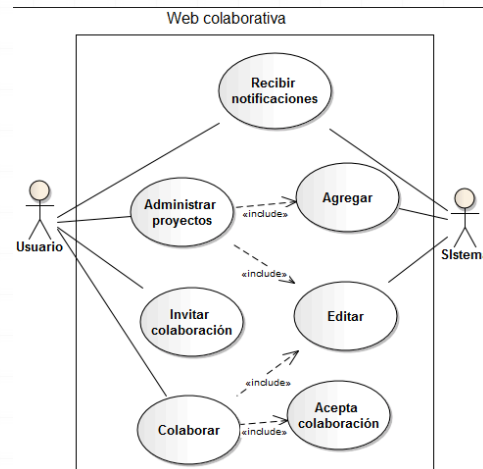
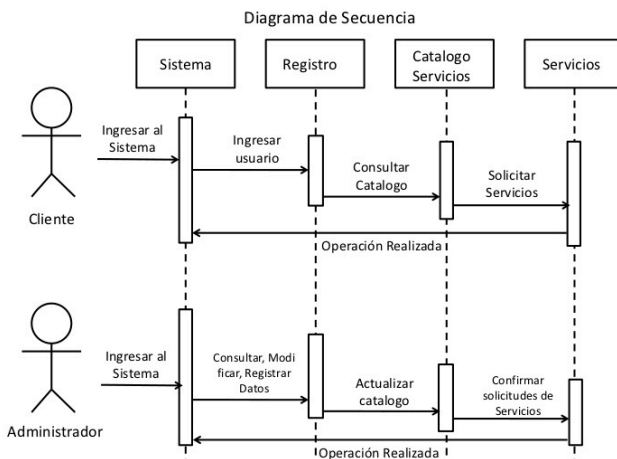
Práticas de Desenvolvimento de Software

Introdução a práticas/técnicas de desenvolvimento  
orientado a objetos

Construção de sistemas de software que demonstrem as  
características básicas da orientação a objetos.

# UML

- 0 Linguagem de modelagem unificada (Unified Modeling Language)
- 0 Linguagem para estruturar projetos de software
- 0 Permite representar um sistema de forma padronizada

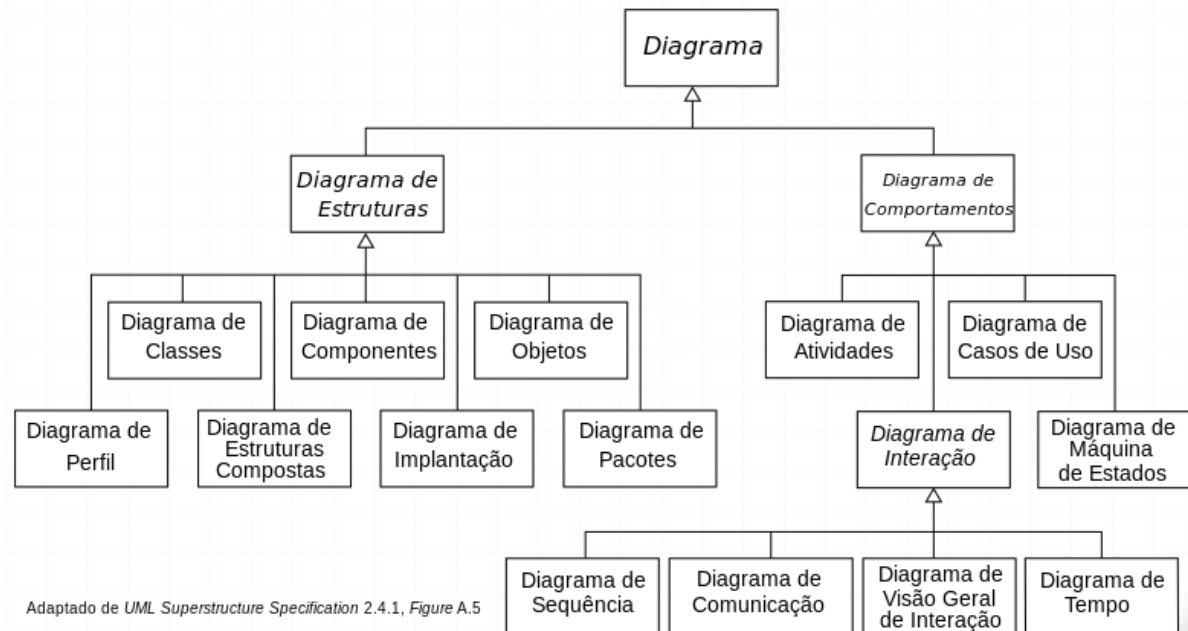


# UML

- 0 Linguagem de modelagem unificada (Unified Modeling Language)
- 0 Linguagem para estruturar projetos de software
- 0 Permite representar um sistema de forma padronizada

## 0 Objetivos:

- 0 Especificação
- 0 Documentação
- 0 Estruturação para maior visualização lógica do desenvolvimento de um sistema.



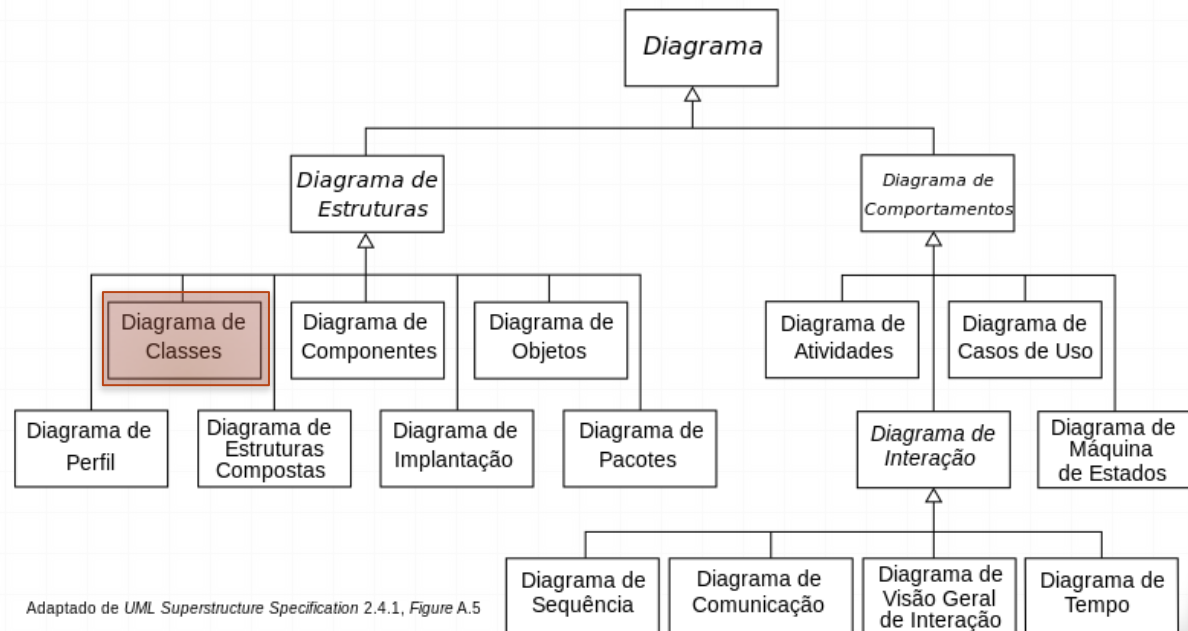
Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

# UML

- 0 Linguagem de modelagem unificada (Unified Modeling Language)
- 0 Linguagem para estruturar projetos de software
- 0 Permite representar um sistema de forma padronizada

## 0 Objetivos:

- 0 Especificação
- 0 Documentação
- 0 Estruturação para maior visualização lógica do desenvolvimento de um sistema.



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

# Diagrama de Classes

- 0 Representação das classes de um sistema.
  - 0 Relacionamentos
  - 0 Atributos
  - 0 Métodos
- 0 Conceitos: Multiplicidade, Visibilidade, Navegabilidade
- 0 Exemplo: Universidade
  - 0 Quais poderiam ser as classes?

Centro

Universidade

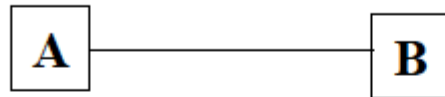
Departamento

Professor

Disciplina

# Relacionamento entre Classes

## Associação



Relacionamento do tipo *estrutural* indicando que existe uma “ligação semântica” entre objetos da classe A e os objetos da classe B.

## Agregação



Relacionamento do tipo *grupo/membro* onde objetos da classe A “utilizam” serviços ou dados de objetos da classe B.

## Composição



Relacionamento do tipo *todo/parte* denotando que os objetos da classe A “compõe-se” de objetos da classe B.

## Realização



Relacionamento do tipo *abstração/realização* onde todo objeto da classe B “realiza” um objeto da classe abstrata ou interface A.

## Herança

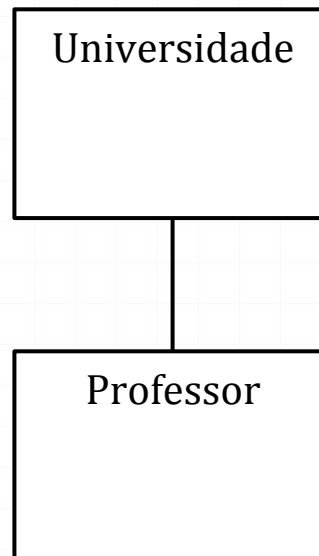


Relacionamento do tipo *generalização/especialização* onde todo objeto da classe B “é” um objeto da classe A.



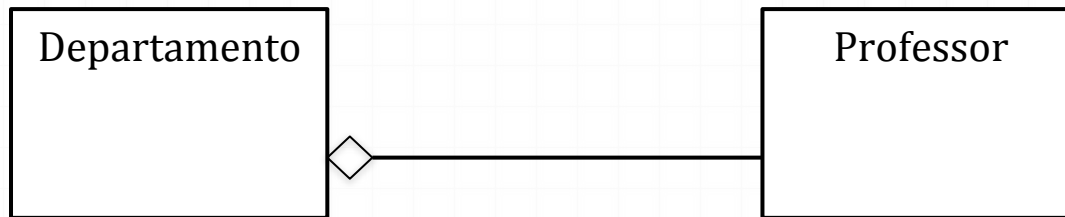
# Associação

- 0 Relacionamento mais comum
- 0 Relacionamento entre duas classes permitindo a comunicação entre elas
- 0 Instâncias de uma classe estão ligadas às instâncias da outra classe



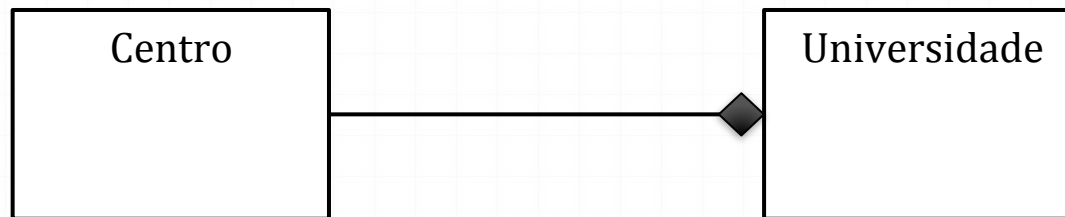
# Agregação

- 0 Grupo/membro (Todo/parte)
- 0 Relacionamento de pertencimento
  - 0 O objeto da classe com o losango contém o objeto da outra classe
  - 0 O objeto da classe sem o losango é membro da outra classe



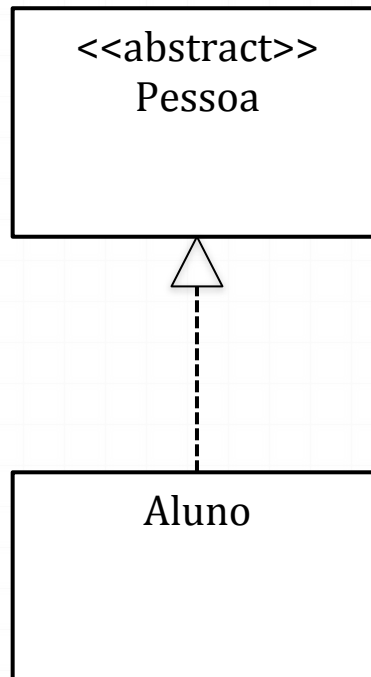
# Composição

- 0 Todo/parte
- 0 Dependência de conteúdo
  - 0 Se o todo não existe os elementos dele também não
  - 0 As partes são criadas e destruídas junto com o todo



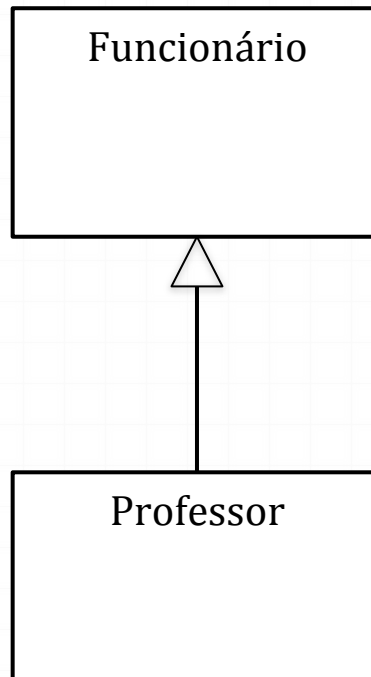
# Realização

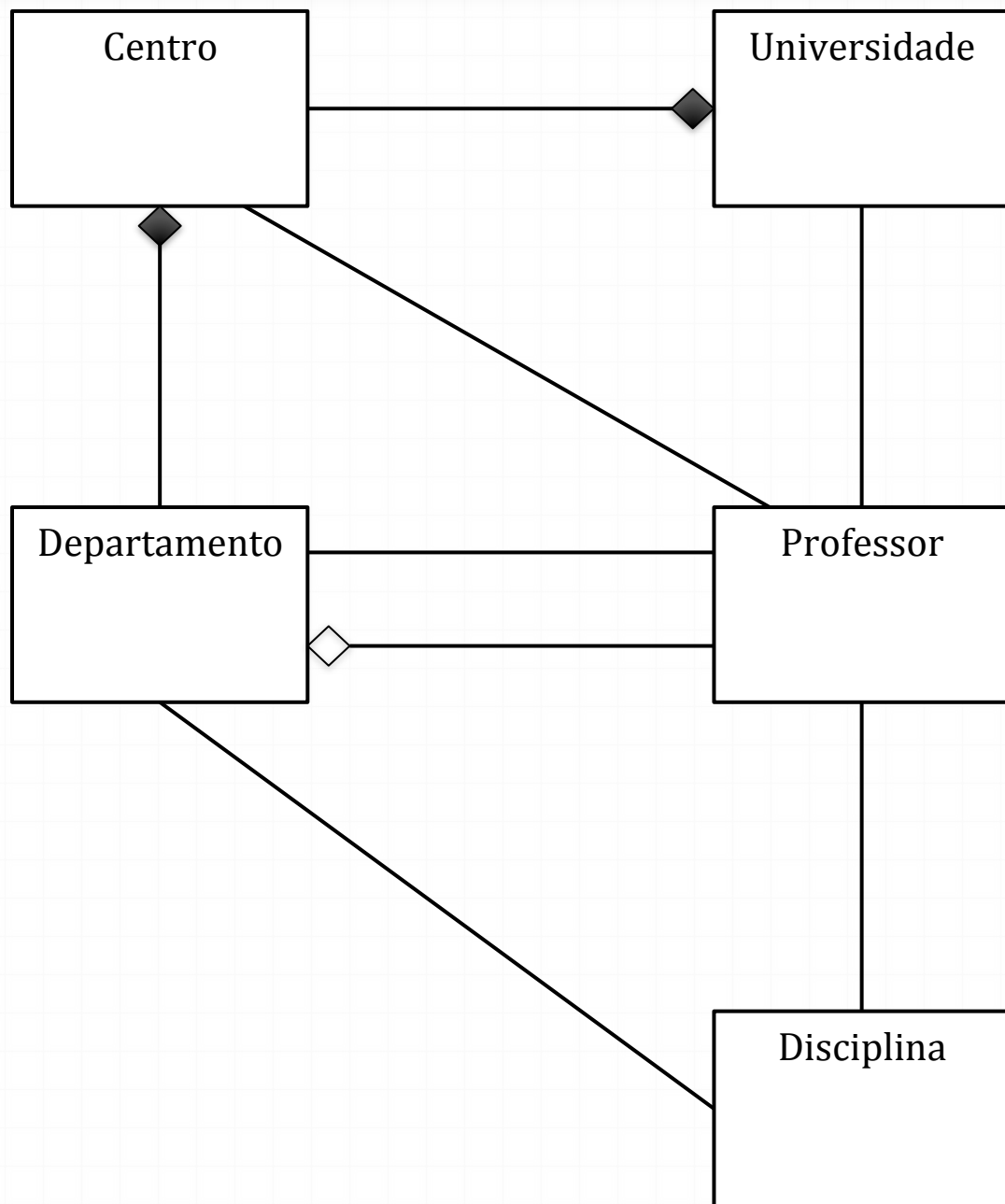
- 0 O relacionamento é do tipo **abstração/realização**
- 0 Representados por classes abstratas ou interfaces



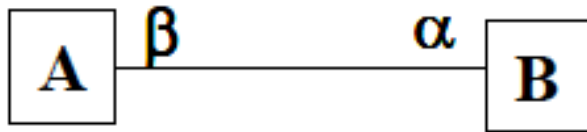
# Herança

- 0 Relacionamento de **generalização/especialização**
- 0 É um





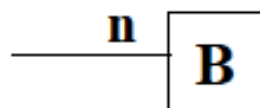
# Multiplicidade



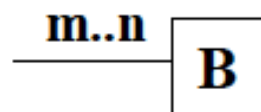
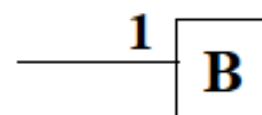
- Cada objeto de **A** relaciona-se com  $\alpha$  objeto de **B**
- Cada objeto de **B** relaciona-se com  $\beta$  objeto de **A**



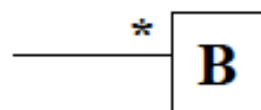
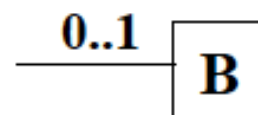
Indica o número de objetos da classe B envolvidos no relacionamento com a classe A.



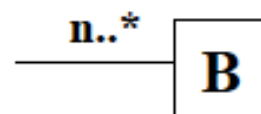
Exatamente **n** objetos. ( $n \geq 0$ )



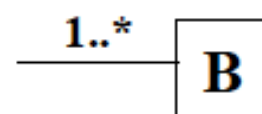
De **m** a **n** objetos. ( $n > m \geq 0$ )



Zero ou mais objetos.



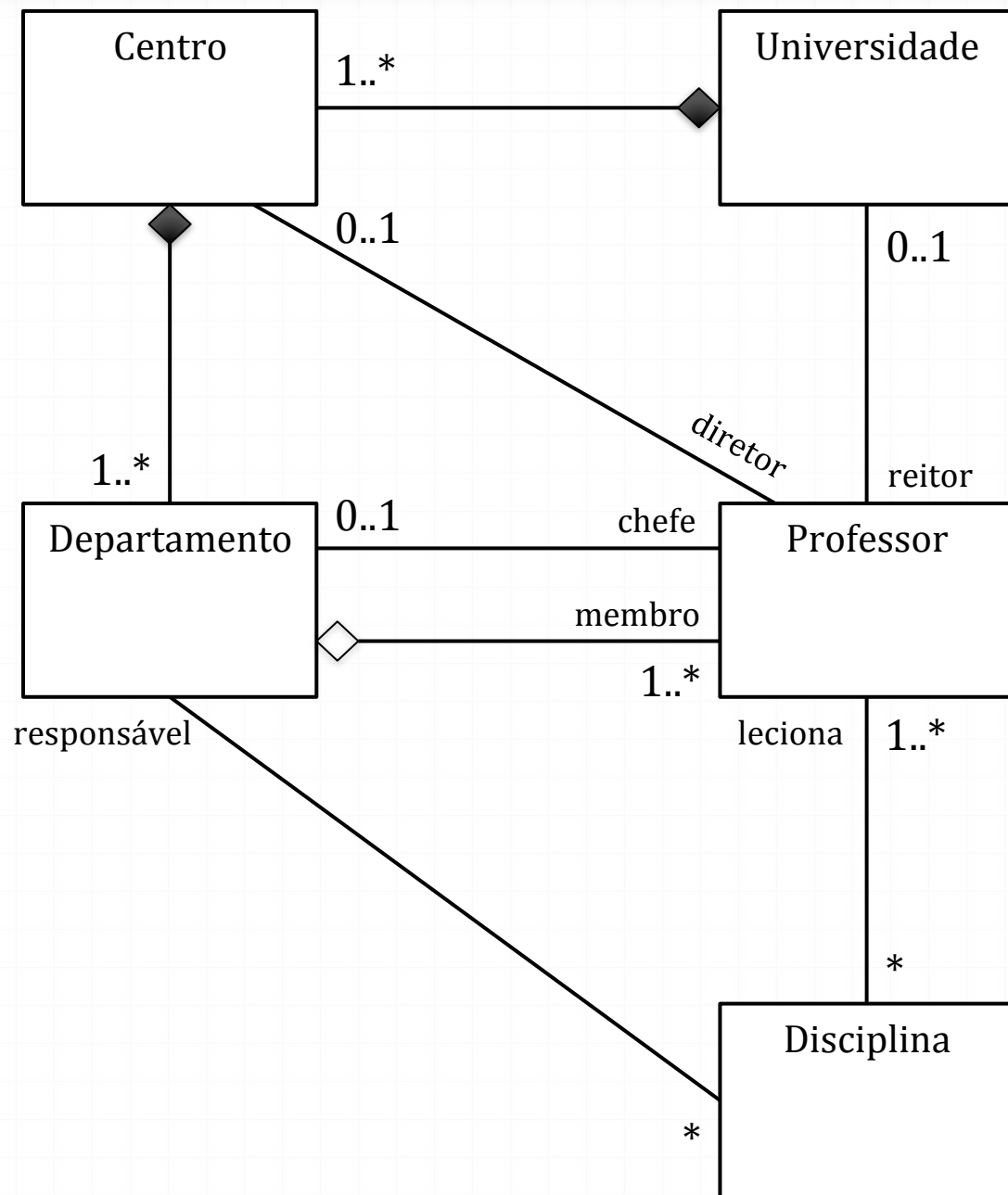
Pelo menos **n** objetos. ( $n \geq 1$ )





# Multiplicidade

Multiplicidade	Significado
0..1	No mínimo zero e no máximo um.
1..1	Um e somente um.
0..*	No mínimo zero e no máximo muitos.
*	Muitos.
1..*	No mínimo um e no máximo muitos.
3..5	No mínimo três e no máximo cinco



# Visibilidade

0 Pública - Public

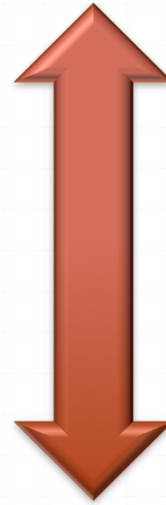
0 Protegida - Protected

0 Privada - Private

+

#

-



Mais acessível

Menos acessível

# Visibilidade

## Public

- Acessível por qualquer outra classe

## Protected

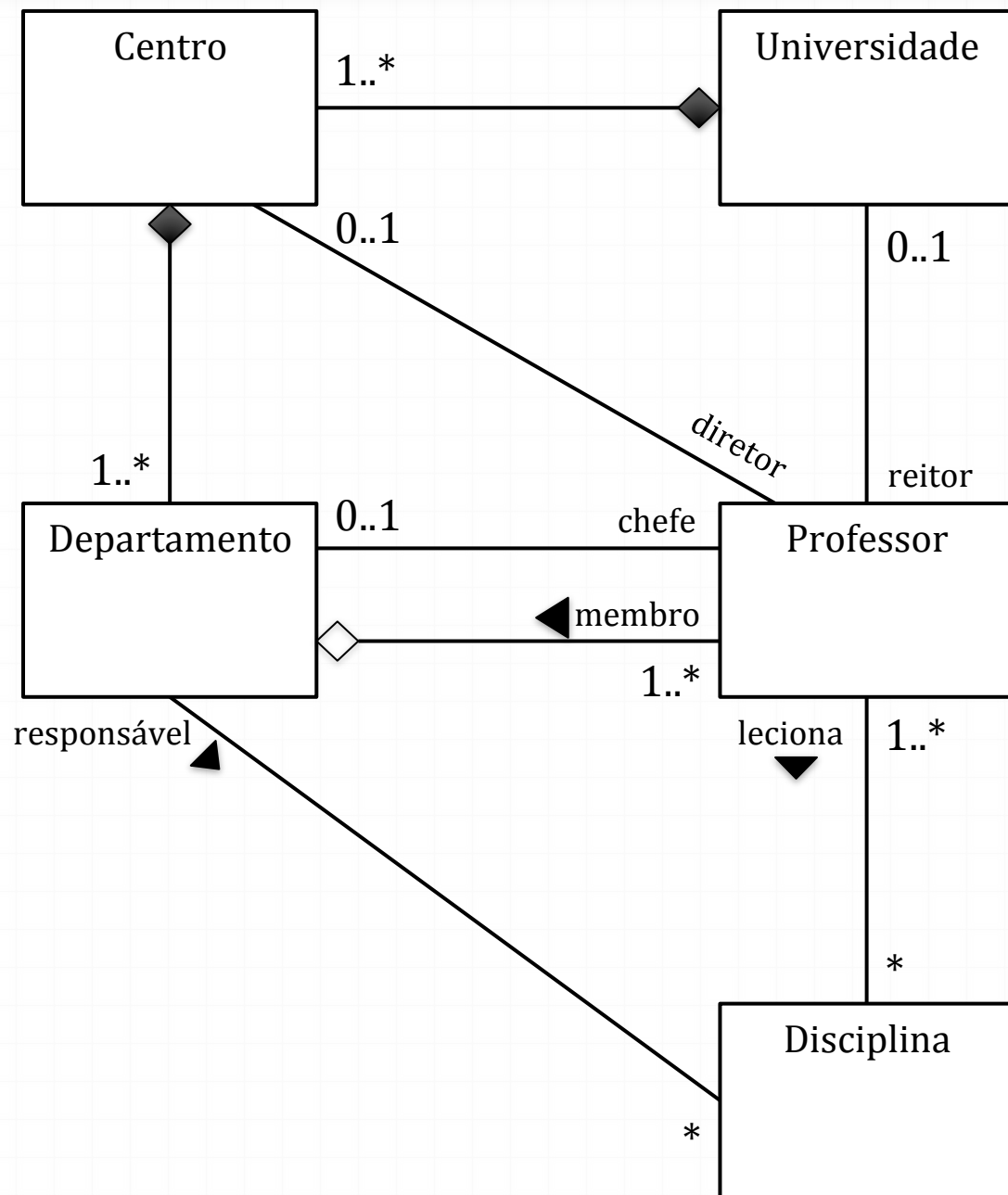
- Acessível pela classe e classes que a estendem

## Private

- Acessível dentro da classe

# Navegabilidade

- 0 Usado para a leitura do relacionamento
- 0 Mostra a direção da navegação entre os objetos e classes



# Perspectiva

## 0 Conceitual

- 0 Conceitos do domínio do problema

- 0 Mostra conceitos e relações entre conceitos

## 0 Especificação

- 0 Foco em principais métodos

## 0 Implementação

- 0 Navegabilidade, visibilidade.

# Perspectiva

Nome da classe

Nome da classe

- Atributo
- Atributo

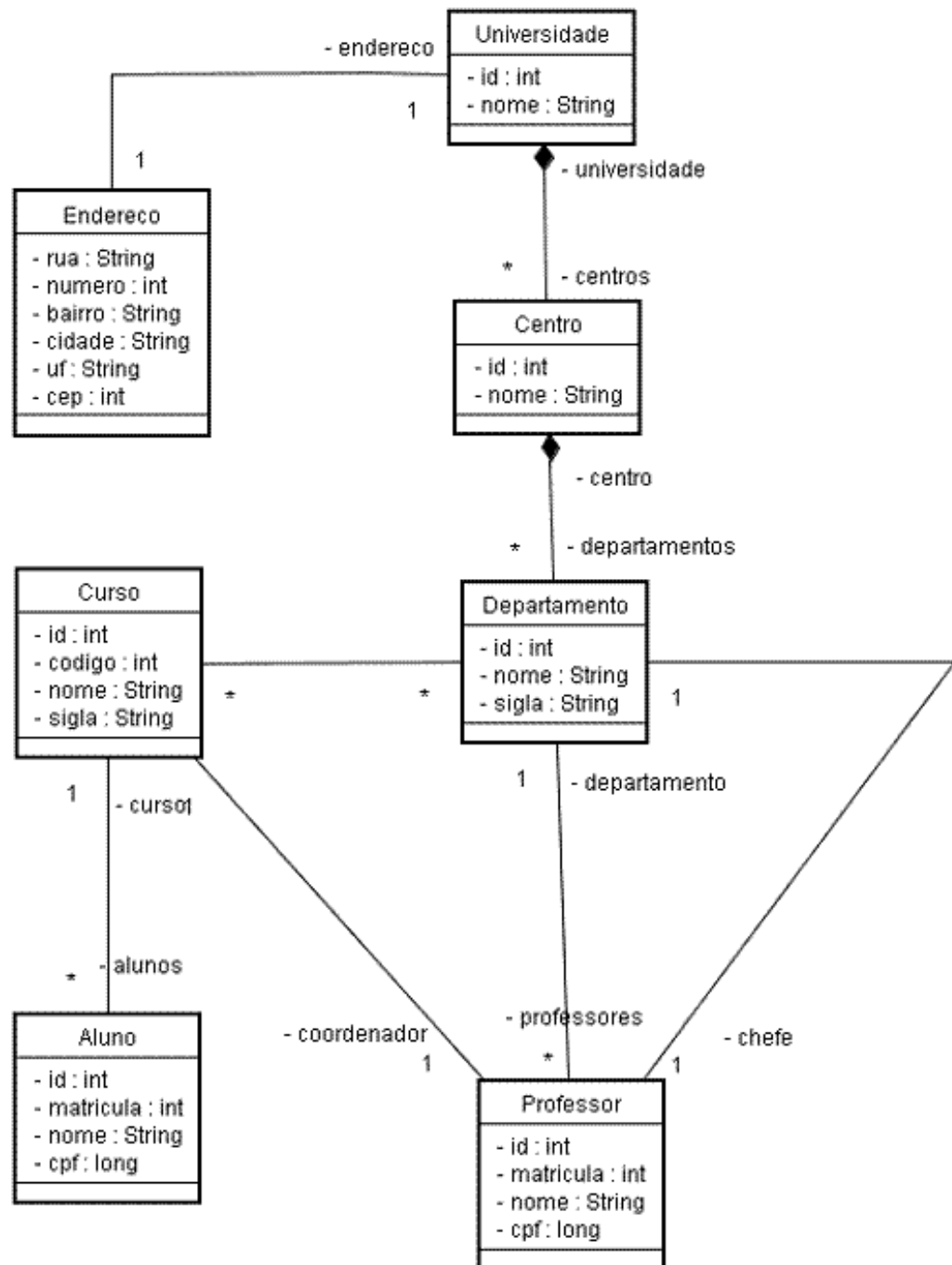
Nome da classe

Nome da classe

- Método
- Método

- Atributo
- Atributo
- Método ( )
- Método( )



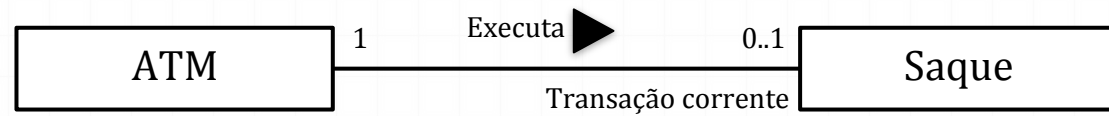


# Exemplo sistema ATM

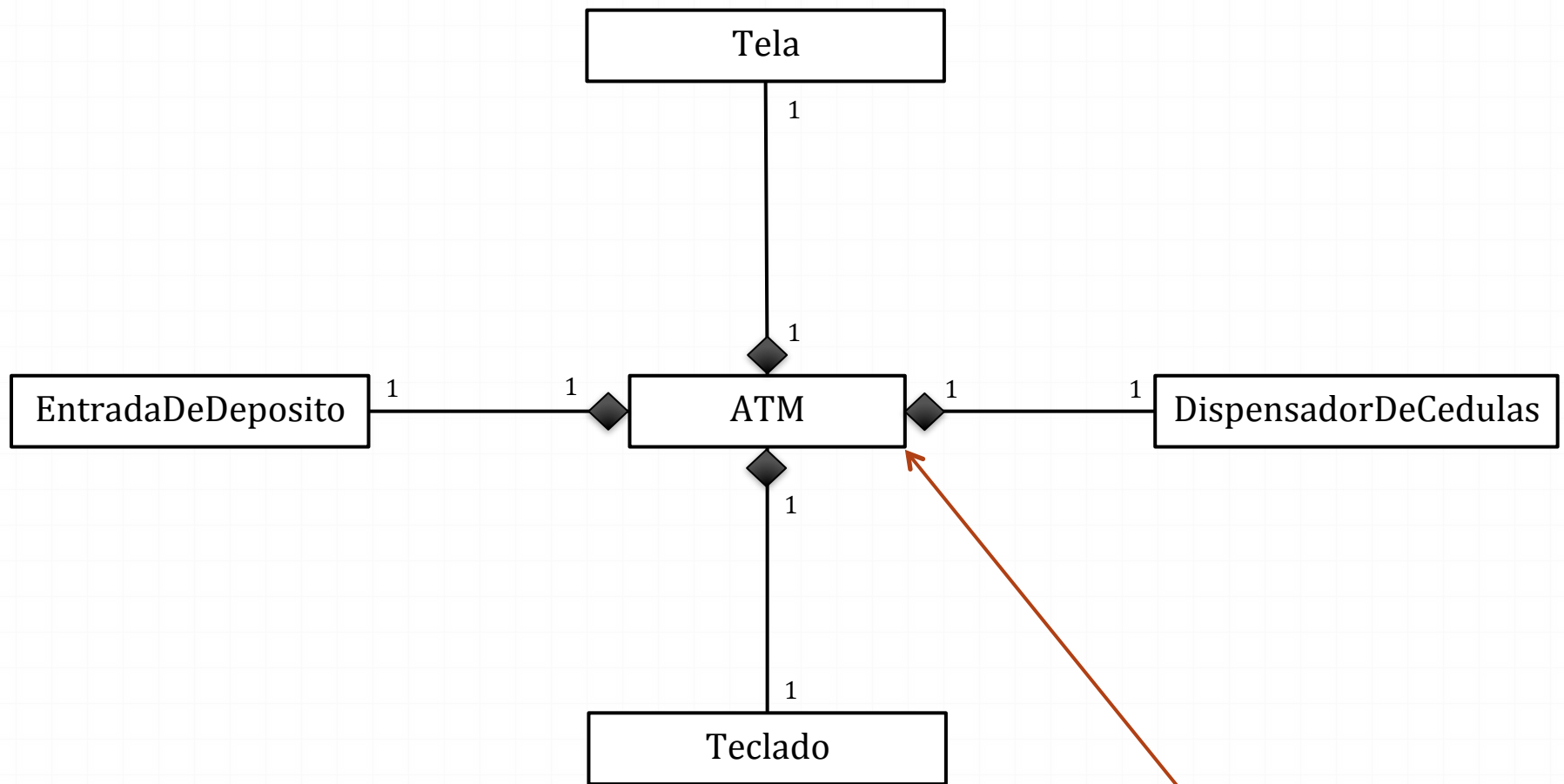
## 0 Identificando as classes (documento de requisitos)

- 0 Atm
- 0 Tela
- 0 Teclado
- 0 Dispensador de cédulas
- 0 Entrada de depósito
- 0 Conta
- 0 Banco de dados do banco
- 0 Pesquisa do saldo
- 0 Retirada/saque
- 0 Depósito

ATM

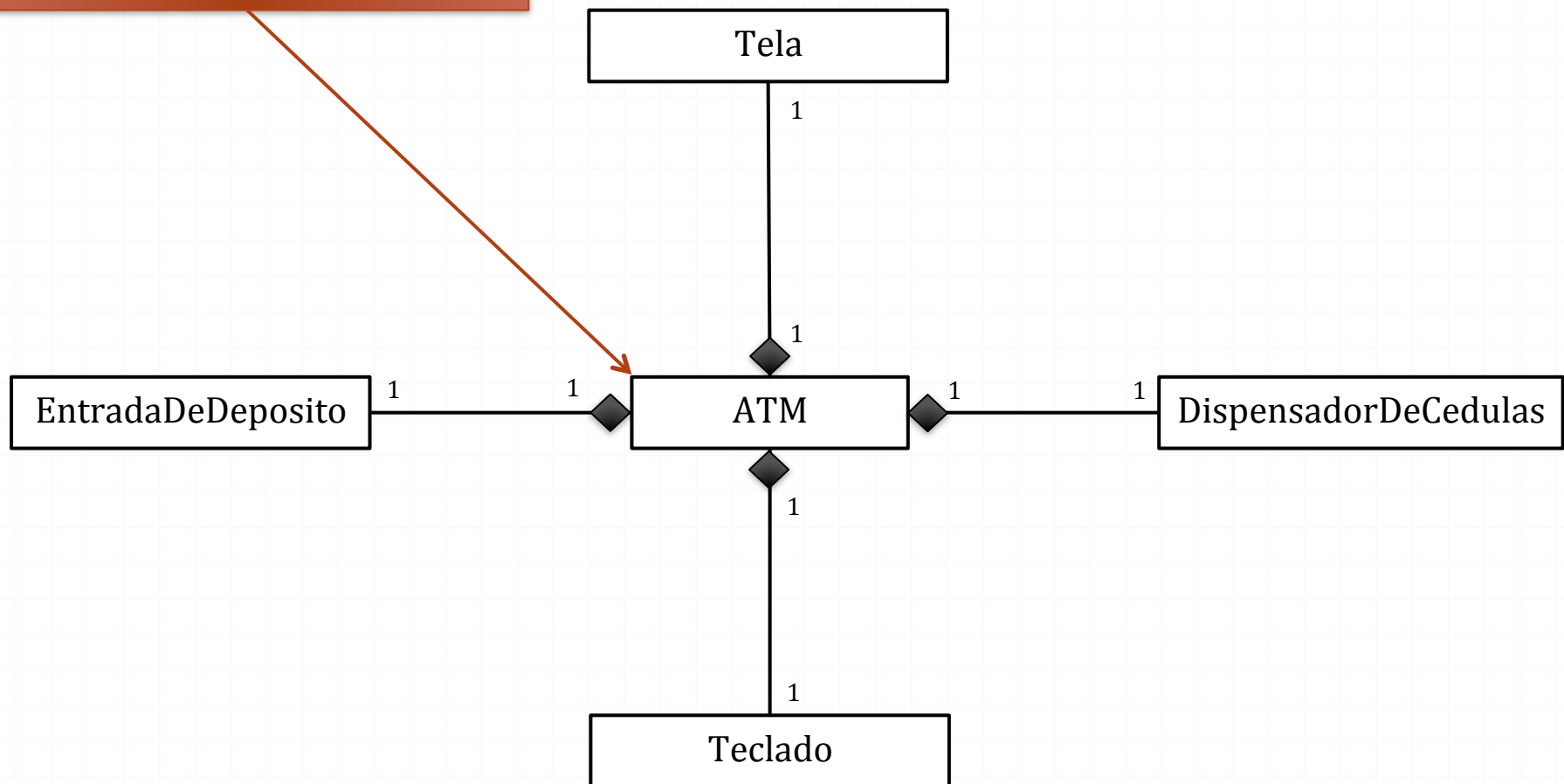


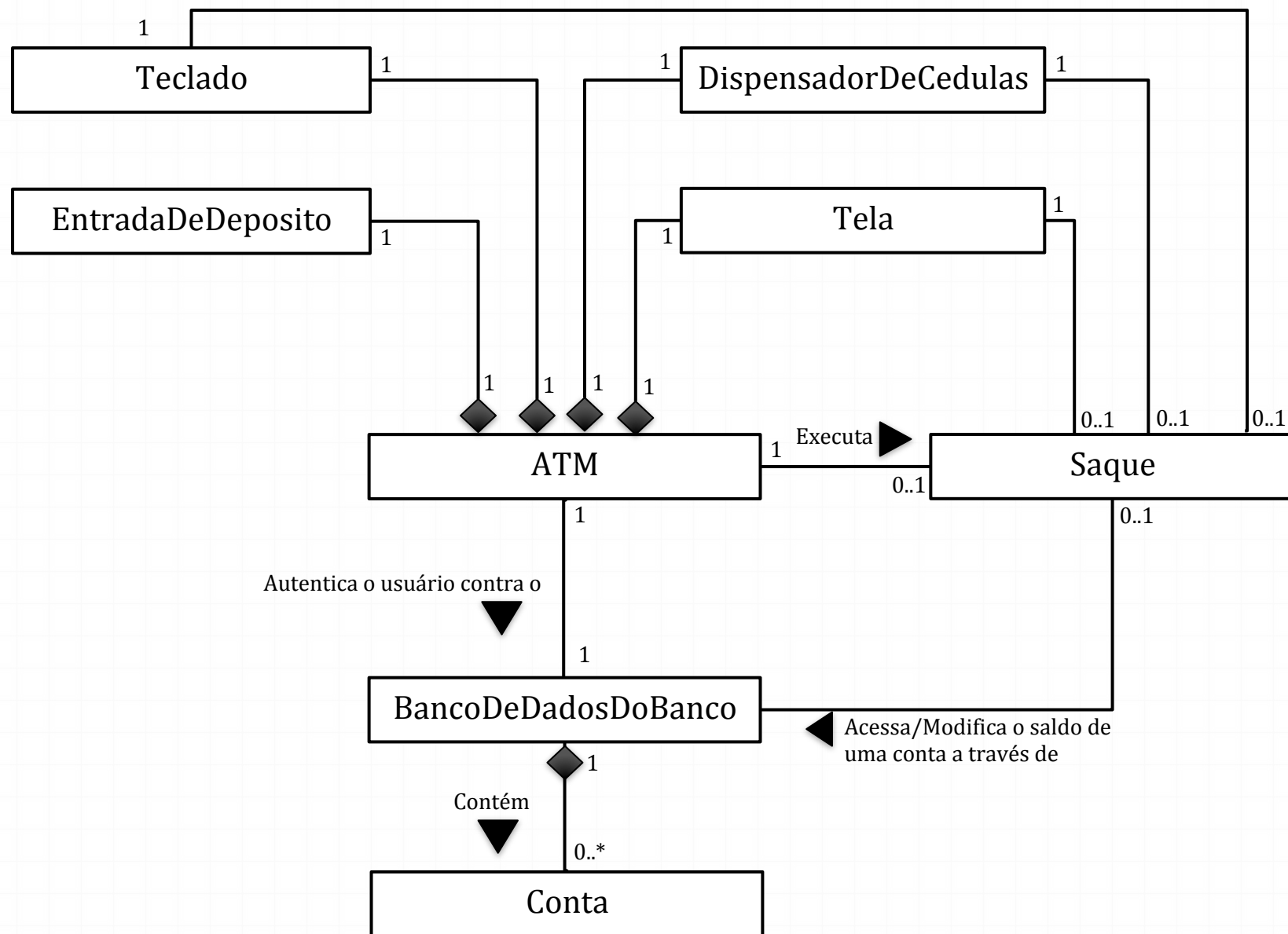
Nome de papel: Identifica o papel que o objeto *Saque* desempenha em seu relacionamento com o ATM



A classe ATM é o todo e as outras classes são as partes.

Em composição as partes só existem quando o todo existir





# Atributos

## 0 Palavras e frases descritivas do documento de requisitos do ATM

Classe	Palavras e frases
ATM	usuário é autenticado
PesquisaDeSaldo	número de conta
Saque	número de conta valor
Deposito	número de conta valor
Conta	número de conta PIN saldo
DispensadorDeCedulas	inicia cada dia carregado com 500 cédulas de 20.00



# Atributos

ATM
usuarioAutenticado: Boolean = false

PesquisaDeSaldo
numeroConta: Integer

Saque
numeroConta: Integer quantidade: Double

Deposito
numeroConta: Integer quantidade: Double

BancoDeDadosDoBanco

Conta
numeroConta: Integer pin: Integer saldoDisponivel: Double saldoTotal: Double

Tela

Teclado

DispensadorDeCedulas
conta: Integer = 500

EntradaDeDeposito

# Operações

ATM
usuarioAutenticado: Boolean = false

PesquisaDeSaldo
numeroConta: Integer
executar( )

Saque
numeroConta: Integer
quantidade: Double
executar( )

Deposito
numeroConta: Integer
quantidade: Double
executar( )

BancoDeDadosDoBanco
autenticarUsuario( ): Boolean obterSaldoDisponivel( ): Double obterSaldoTotal( ): Double creditar( ) debitar( )

Conta
numeroConta: Integer pin: Integer saldoDisponivel: Double saldoTotal: Double
validarPIN( ): Boolean obterSaldoDisponivel( ): Double obterSaldoTotal( ): Double creditar( ) debitar( )

Tela
mostrarMensagem( )

Teclado
obterEntrada( ): Integer

DispensadorDeCedulas
conta: Integer = 500
dispensarCedulas( ) haSuficienteCedulasDisponiveis: Boolean

EntradaDeDeposito
envelopeFoiRecebido( ): Boolean

# Operações

ATM
usuarioAutenticado: Boolean = false

PesquisaDeSaldo
numeroConta: Integer
executar( )

Saque
numeroConta: Integer
quantidade: Double
executar( )

Deposito
numeroConta: Integer
quantidade: Double
executar( )

BancoDeDadosDoBanco
autenticarUsuario( ): Boolean
obterSaldoDisponivel( ): Double
obterSaldoTotal( ): Double
creditar( )
debitar( )

Conta
numeroConta: Integer
pin: Integer
saldoDisponivel: Double
saldoTotal: Double
validarPIN( ): Boolean
obterSaldoDisponivel( ): Double
obterSaldoTotal( ): Double
creditar( )
debitar( )

Tela
mostrarMensagem( )

Teclado
obterEntrada( ): Integer

DispensadorDeCedulas
conta: Integer = 500
dispensarCedulas( )
haSuficienteCedulasDisponiveis: Boolean

EntradaDeDeposito
envelopeFoiRecebido( ): Boolean

Parâmetros

# Atividade

- 0 Criar um diagrama de classe (de 5 a 8 classes) de um sistema qualquer contendo
  - 0 Atributos
  - 0 Operações
  - 0 Navegabilidade
  - 0 Multiplicidade
  - 0 Os diferentes tipos de visibilidade
- 0 Exemplos de sistemas:
  - 0 Sistema de matrículas
  - 0 Sistema de pagamento
  - 0 Sistema de empréstimo de livros
  - 0 Sistema de evento (artigos)