

# Orientação a Objetos

## Paradigma Orientado a Objetos (POO)

---

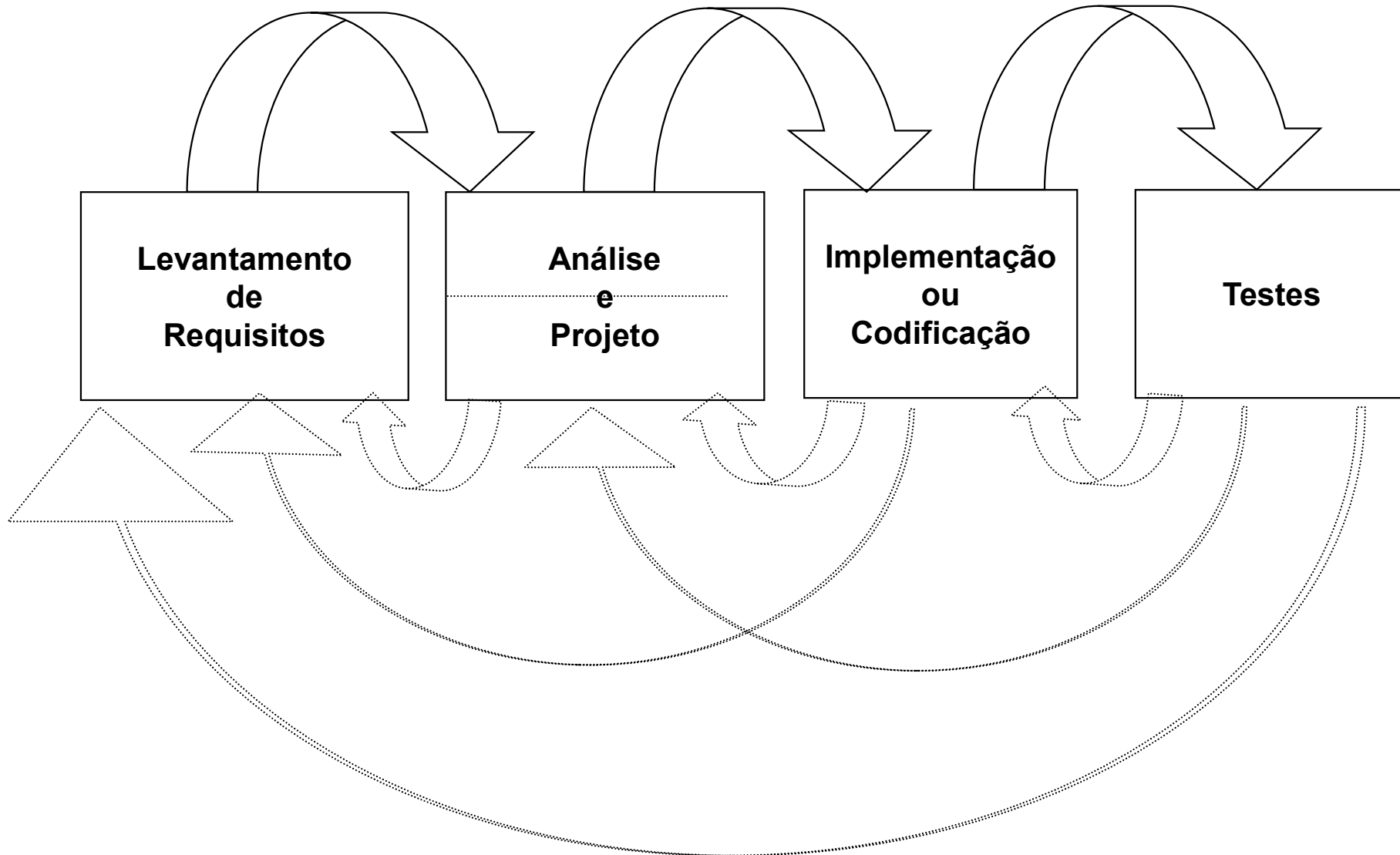
Slides 9: Modelagem Orientada a Objetos  
em *Unified Modeling Language* (UML)  
– Diagrama de Classes

---

Análise/Projeto em UML visando  
a Programação Orientada a Objetos (C++, ...)

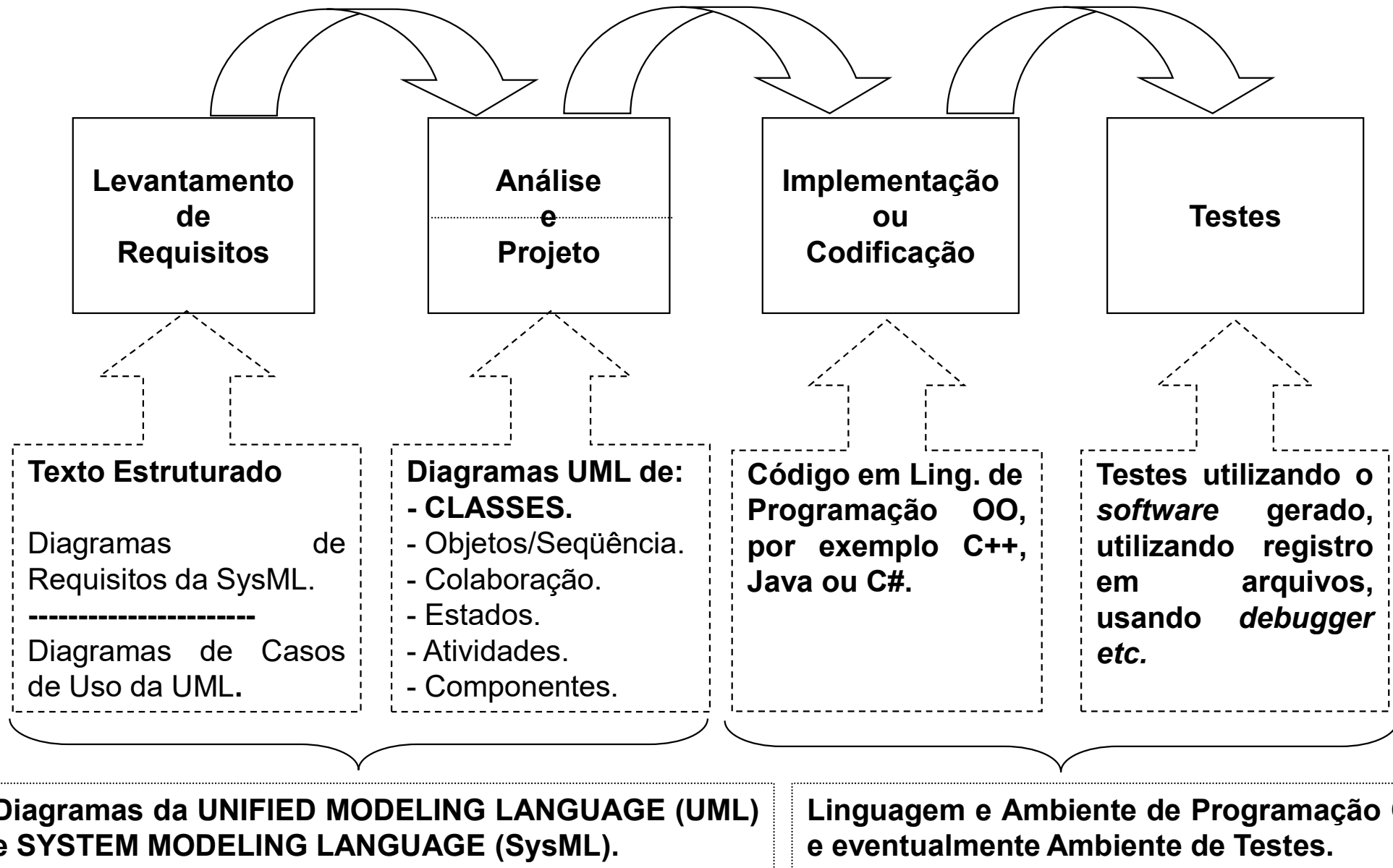
# Engenharia de Software

## Visão Clássica



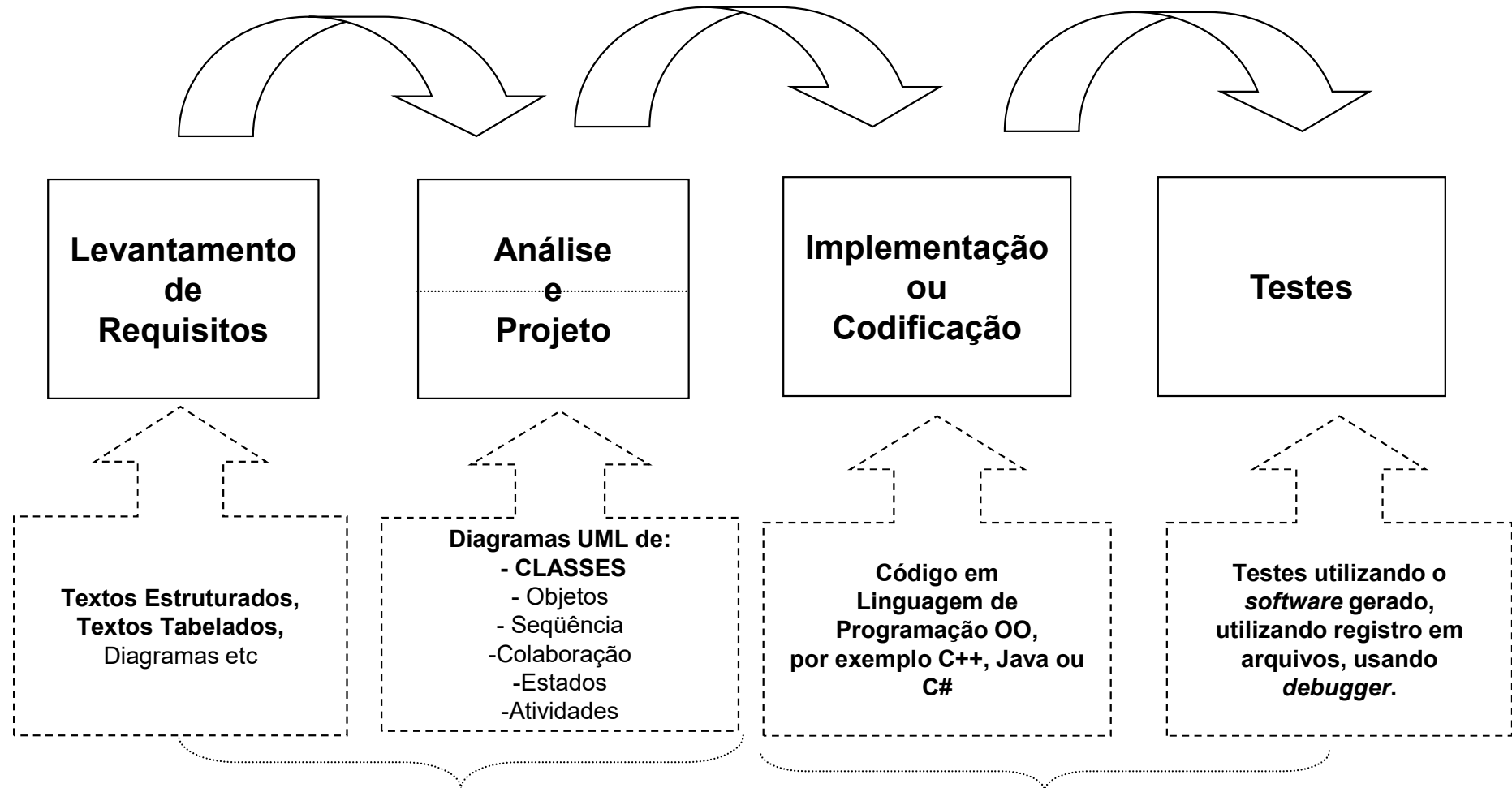
# Engenharia de Software OO

## Exemplos de Técnicas



# Engenharia de Software OO

## Exemplos de Ferramentas



**Ferramentas CASE: StarUML, Jude, Rational Rose, System Architect, Together, VisualParadigm, Rhapsody**  
...

**Ambiente de Programação OO** (integráveis as Ferramentas CASE): **Microsoft Visual Studio, Microsoft Visual C++ .net Express Edition**, Microsoft Visual C++ .net, Microsoft Visual C++, Borland Builder C++, Borland C++, CodeBlocks, Dev C++, G++.

# Bibliografias

---

## **Bibliografias:**

- Pressman, R. S. **Software Engineering – A Practitioner’s Approach**. 6th Edition McGraw Hill (Higher Education). 2005. ISBN 0-07-285318-2.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Software Development Process**. 1st Edition. Addison-Wesley. 2005. ISBN 0-201-57169-2.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. 2nd Edition. Addison-Wesley. 2005. ISBN 0-321-26797-4.
- Bezerra, E. **Princípios de Análise e Projeto de Sistemas com UML**. Editora Campus. 2003. ISBN 85-352-1032-6.

## **Outras bibliografias:**

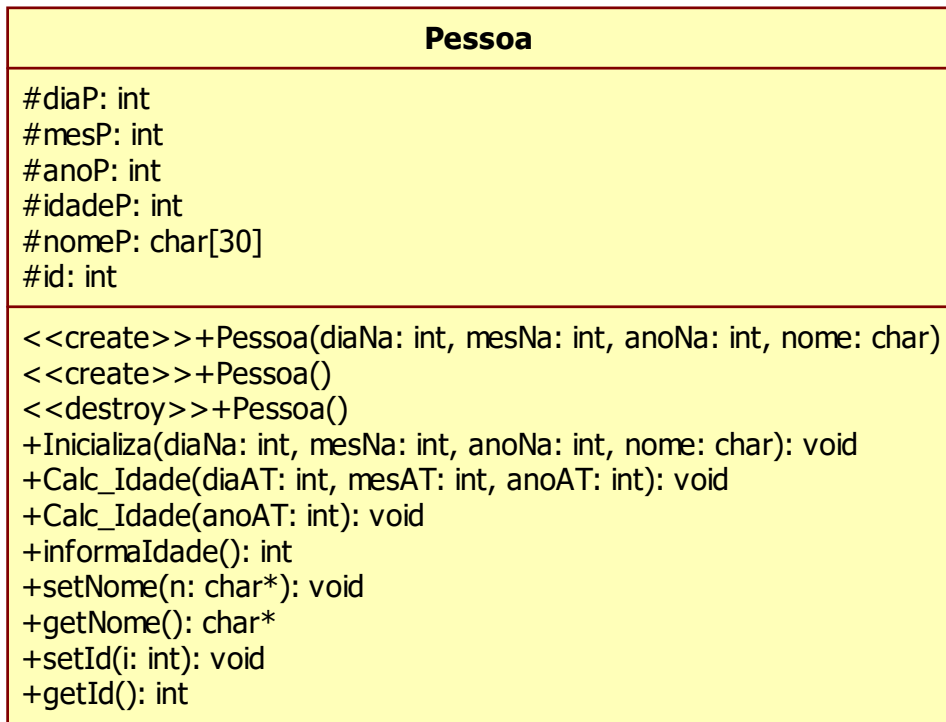
- ~~— GAMMA, E.; HELM, R.; Johnson, R.; Vlissides, J. **Design Patterns: Elements of Reusable Object-oriented Software**. Addison Wesley Longman, 1995.~~
- ~~— Largman, G. **Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design**. Prentice Hall. 1998. ISBN 0-13-748880-7.~~

# *Unified Modeling Language (UML)*

## Diagrama de Classe

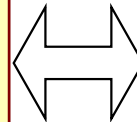
# Análise-Projeto / Implementação

Uma classe em UML

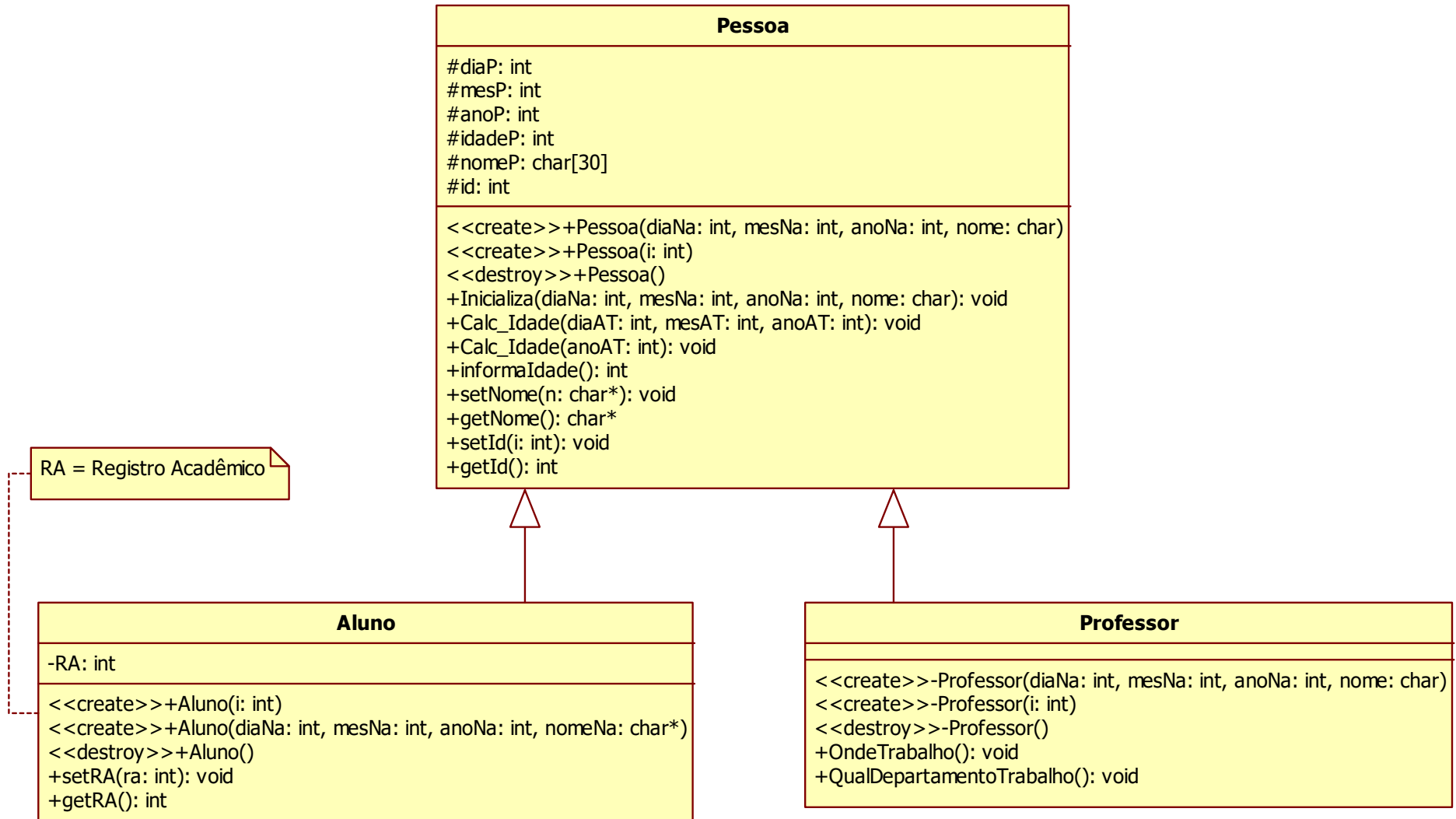


Uma classe em C++

```
#ifndef _PESSOA_H_
#define _PESSOA_H_
class Pessoa
{
protected:
    int diaP;
    int mesP;
    int anoP;
    int idadeP;
    char nomeP[ 30 ];
    int id;
public:
    Pessoa ( int diaNa, int mesNa, int anoNa, char* nome = "" );
    Pessoa ( );
    ~Pessoa ( );
    void Inicializa (int diaNa, int mesNa, int anoNa, char* nome = "" );
    void Calc_Idade (int diaAT, int mesAT, int anoAT );
    void Calc_Idade ( int anoAT );
    int informaIdade ( );
    void seld ( int i ) { id = i; }
    int getId ( ) { return id; }
    void seNome ( char* n ) { strcpy(nomeP, n); }
    char* getNome ( ) { return nome; }
};
#endif
```



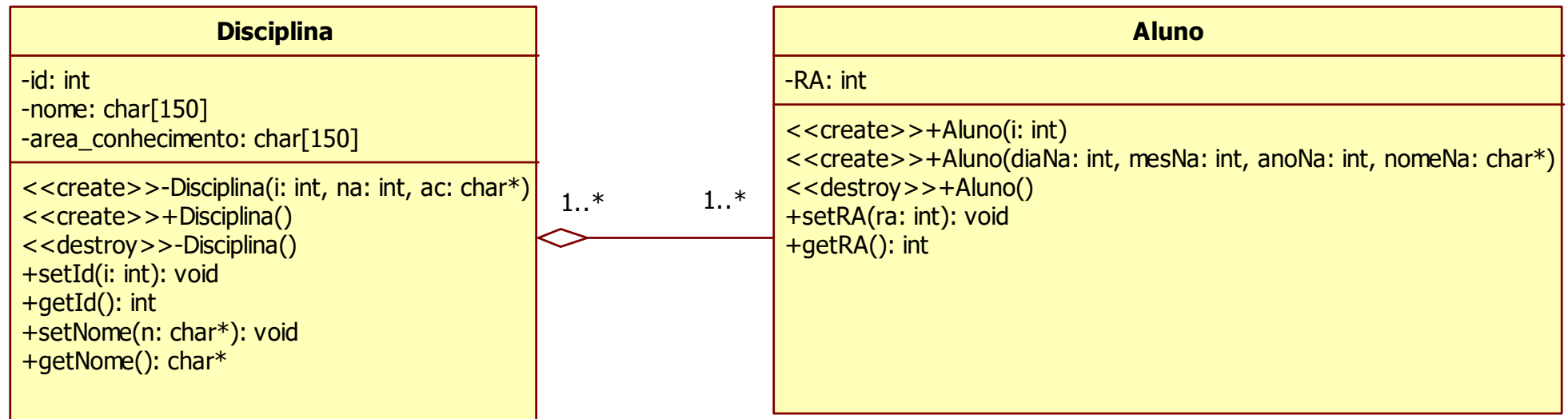
# Diagrama de Classe (de "Análise") - Herança





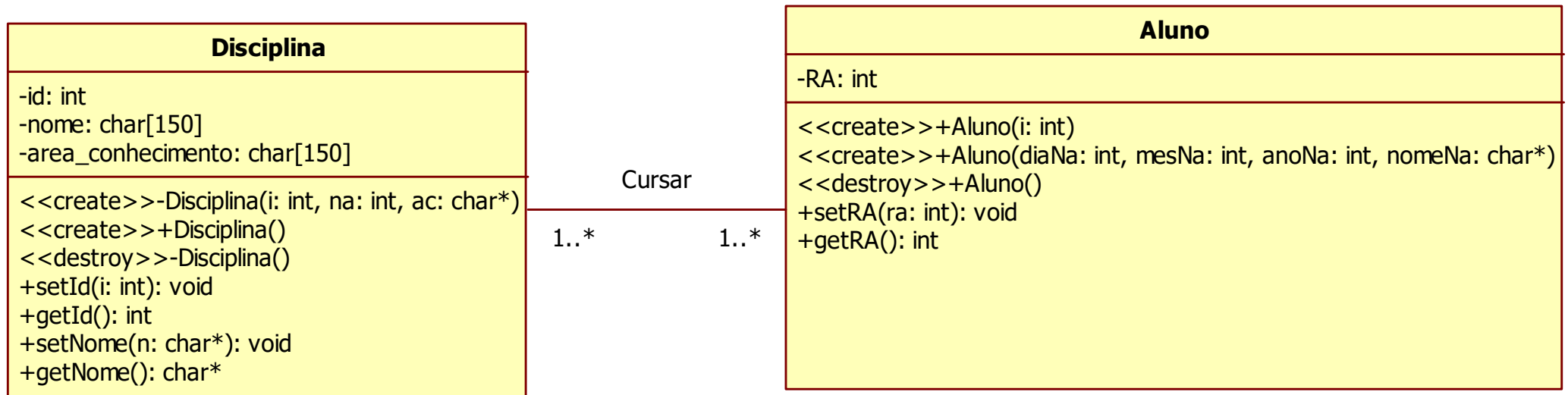
# Diag. Classes (“Análise”) Agregação (Fraca -Simple)

---



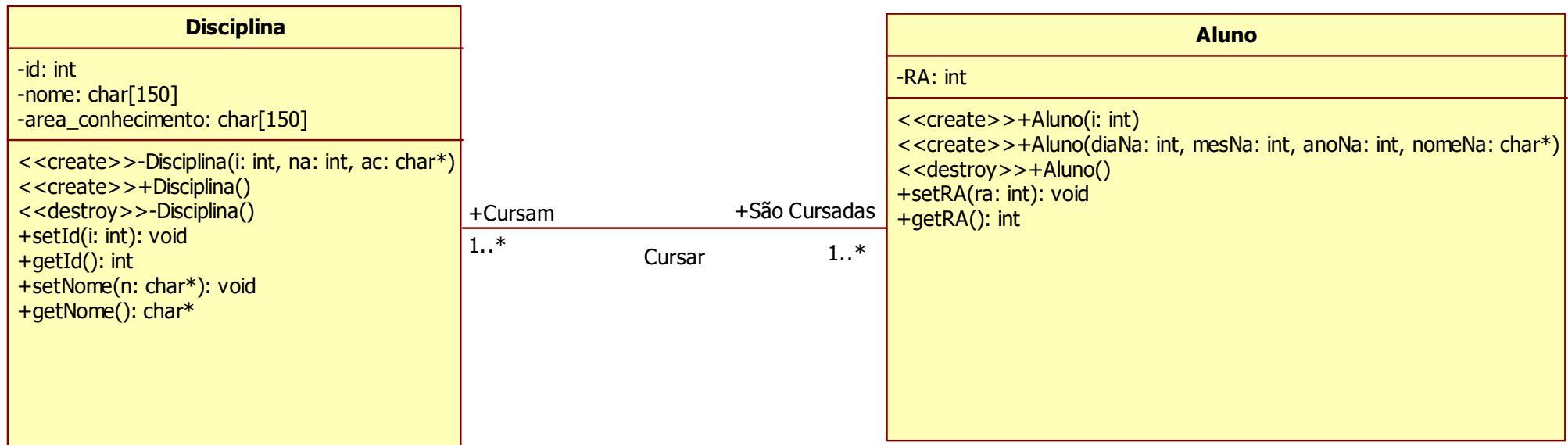
# Diagrama de Classes (“Análise”) - Associação

---

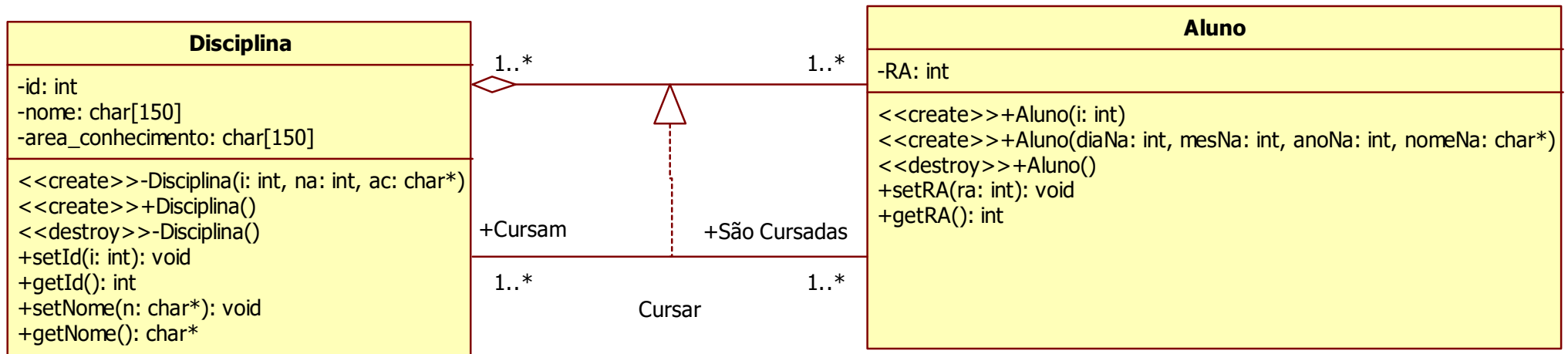


# Diagrama de Classes (“Análise”) – Papeis associativos

---

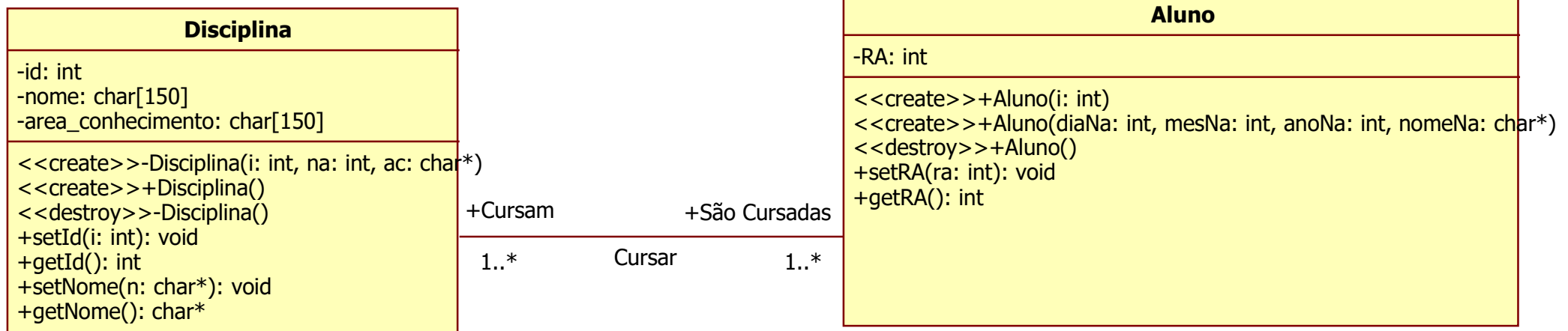


# Realização – Relação entre Relações



# Realização – Relação entre Relações

---



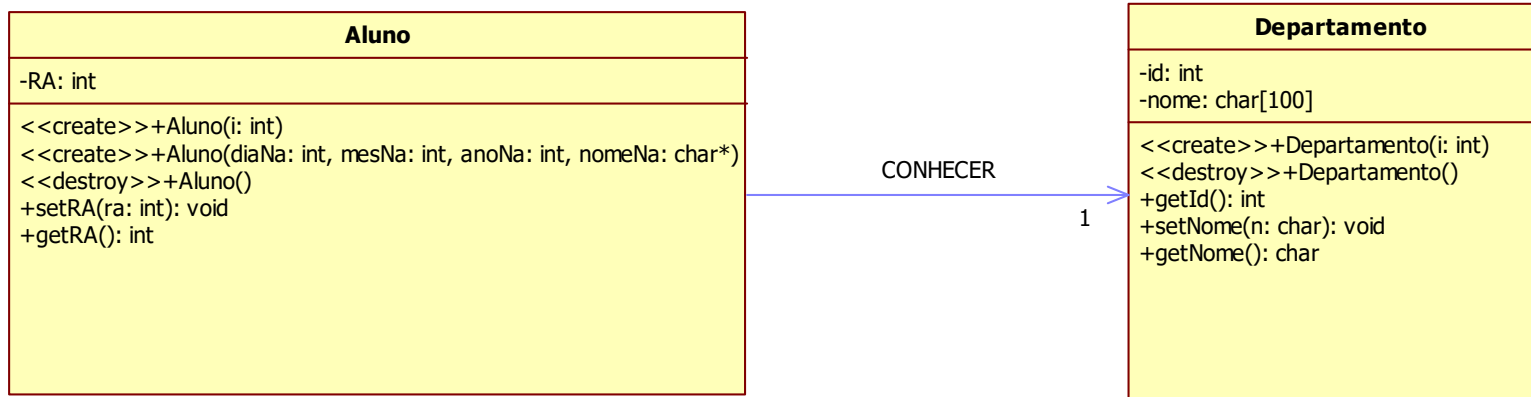
# *Unified Modeling Language (UML)*

Diagrama de Classe  
(de "análise" e de "projeto")

# Diagrama de Classe

## Associação Direcional

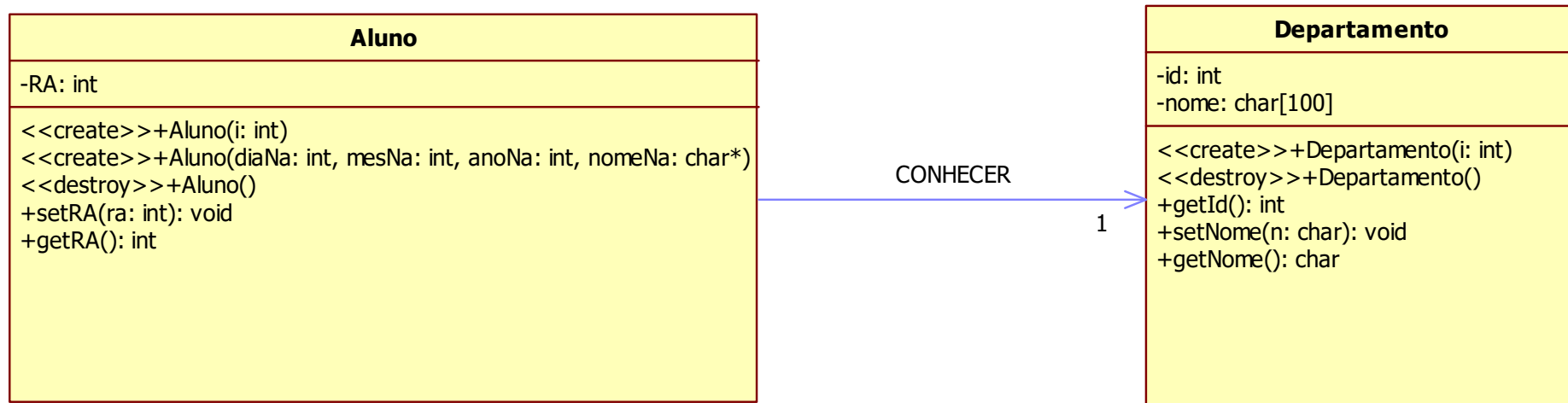
---



# Diagrama de Classe

## Associação Direcional

---



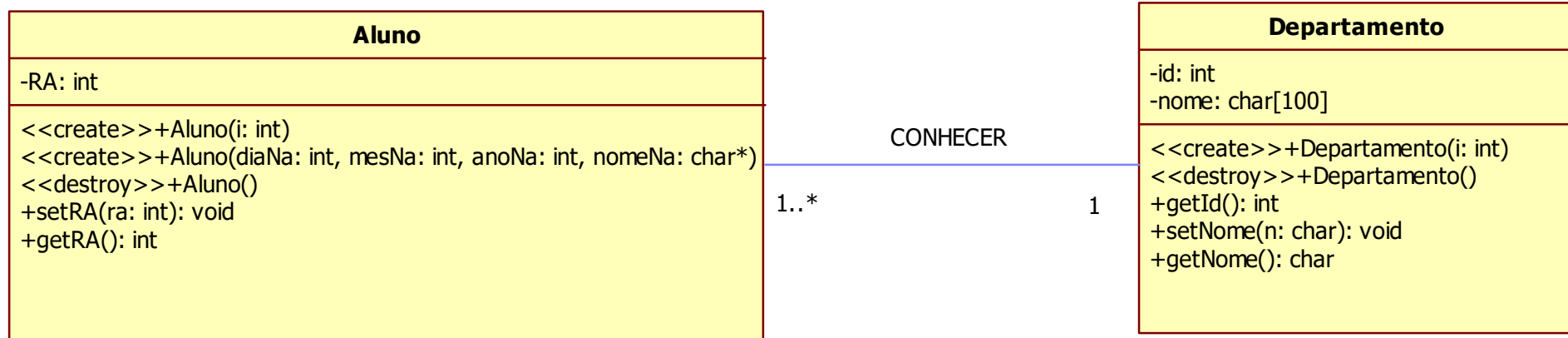
Aqui seria um diagrama de classes no escopo da 'análise', pois contém apenas as informações essenciais das classes e de seus relacionamentos.



# Diagrama de Classe (de “Análise”)

## Associação Bidirecional

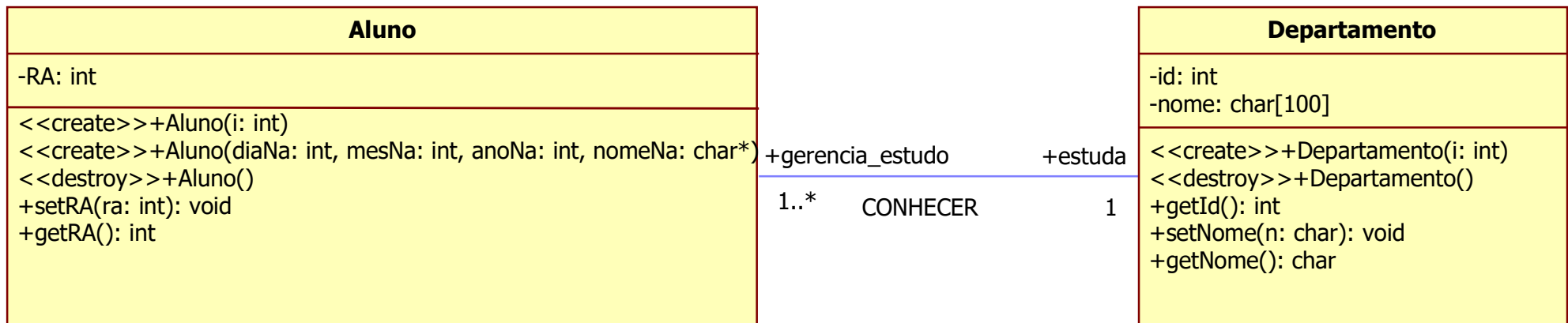
---



# Diagrama de Classe (de “Análise”)

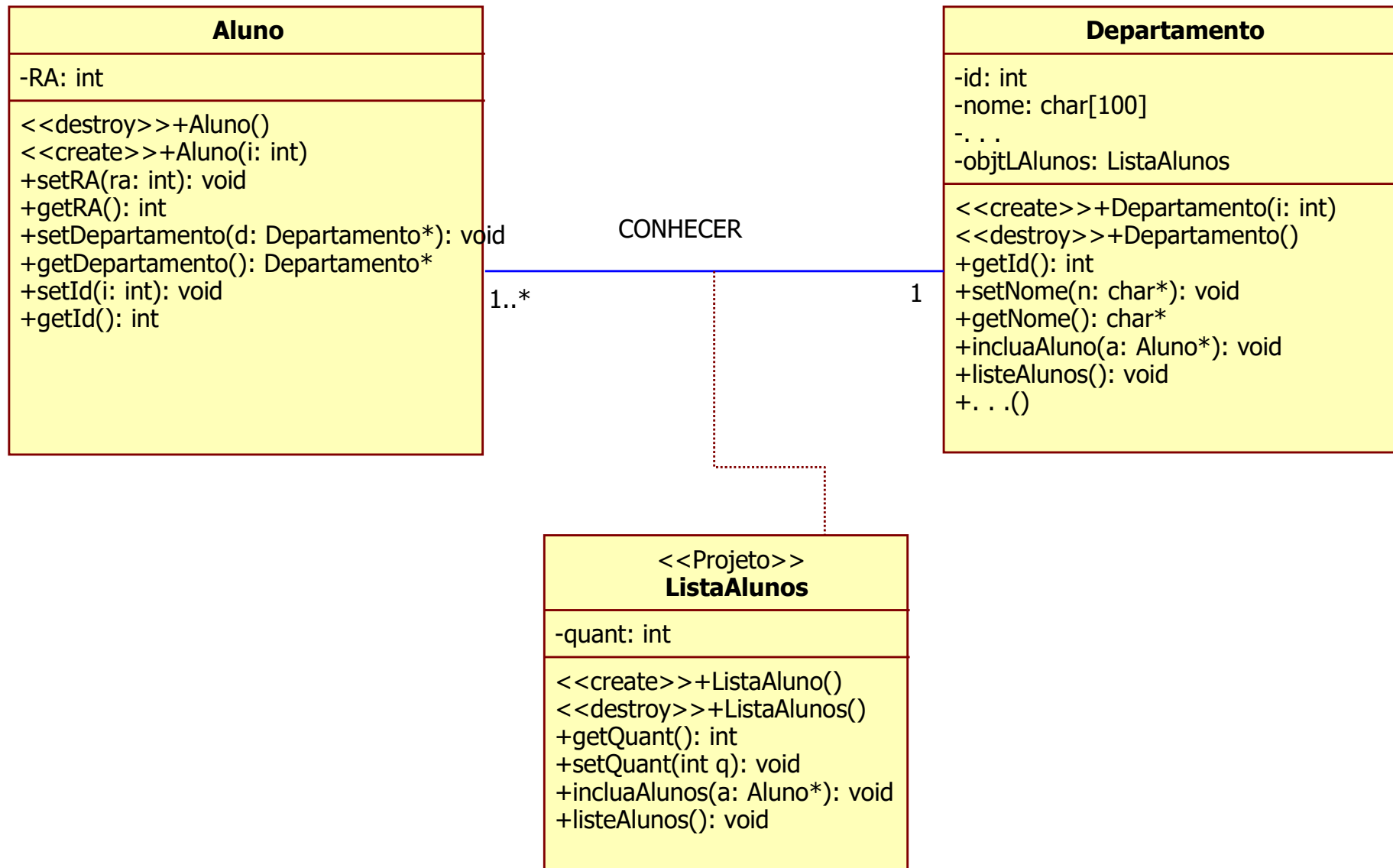
## Associação Bidirecional com Papeis

---



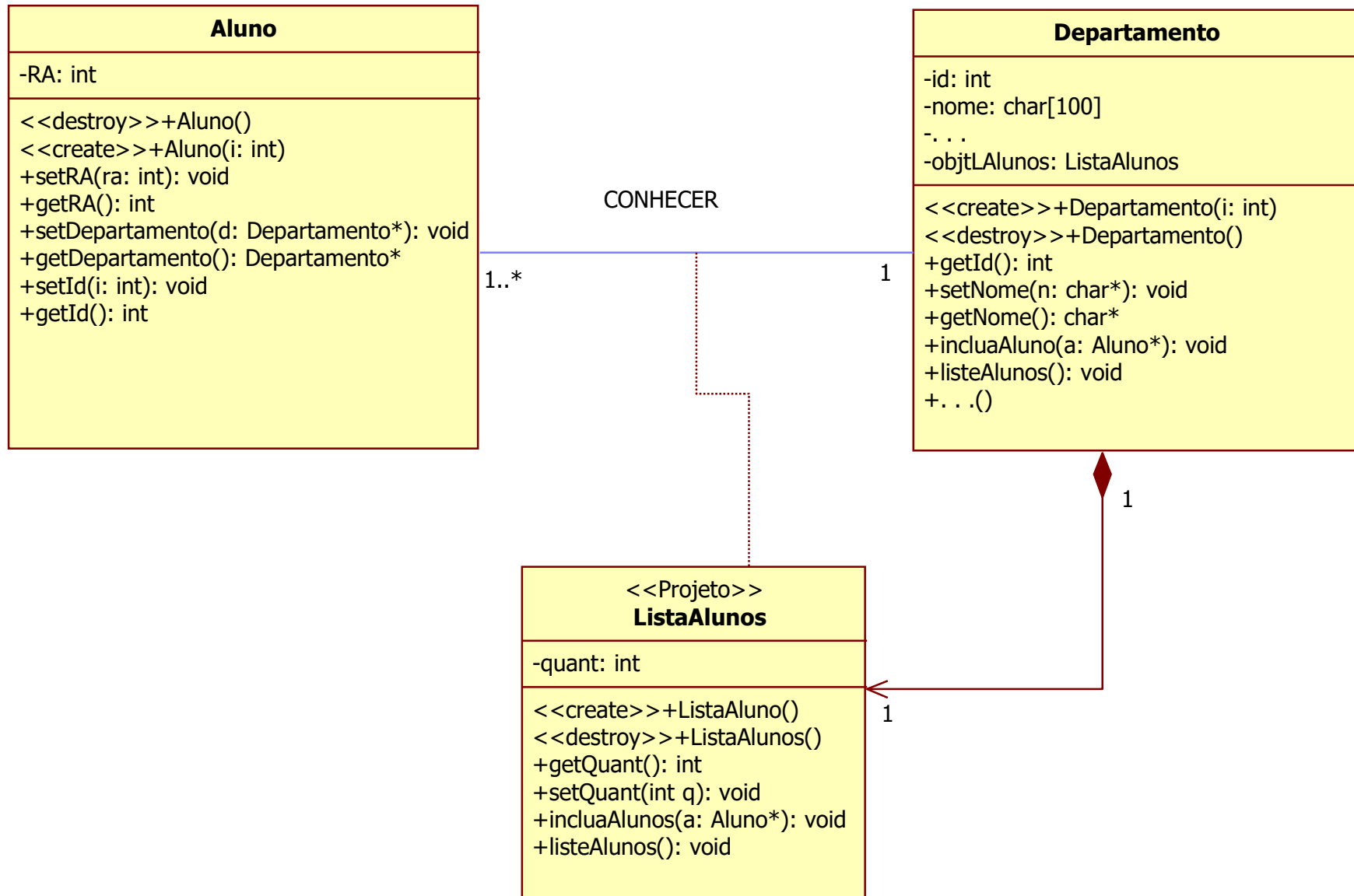
# Diagrama de Classe (de “Projeto”)

## Classe de Associação

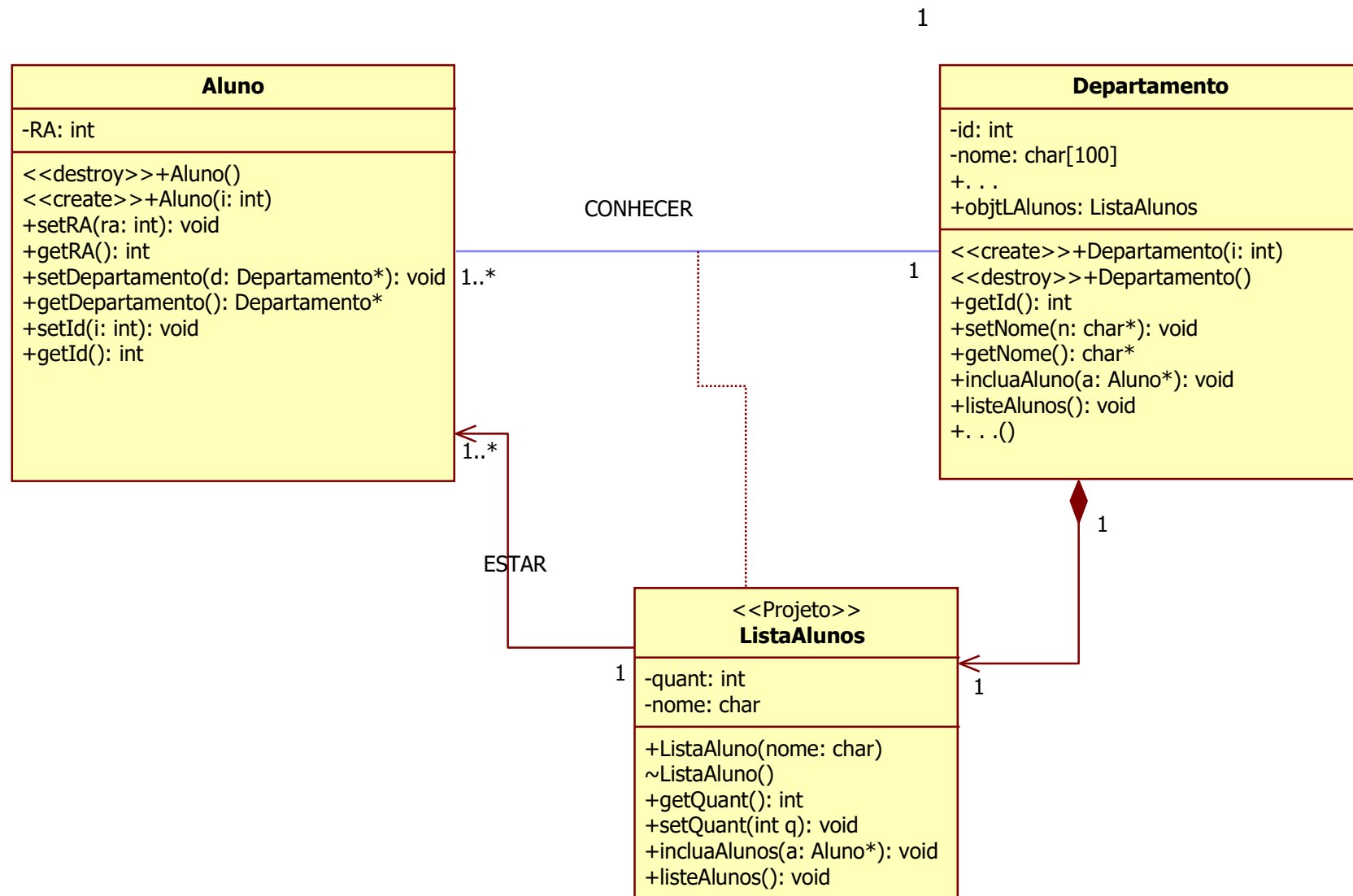


# Diagrama de Classe (de “Projeto”)

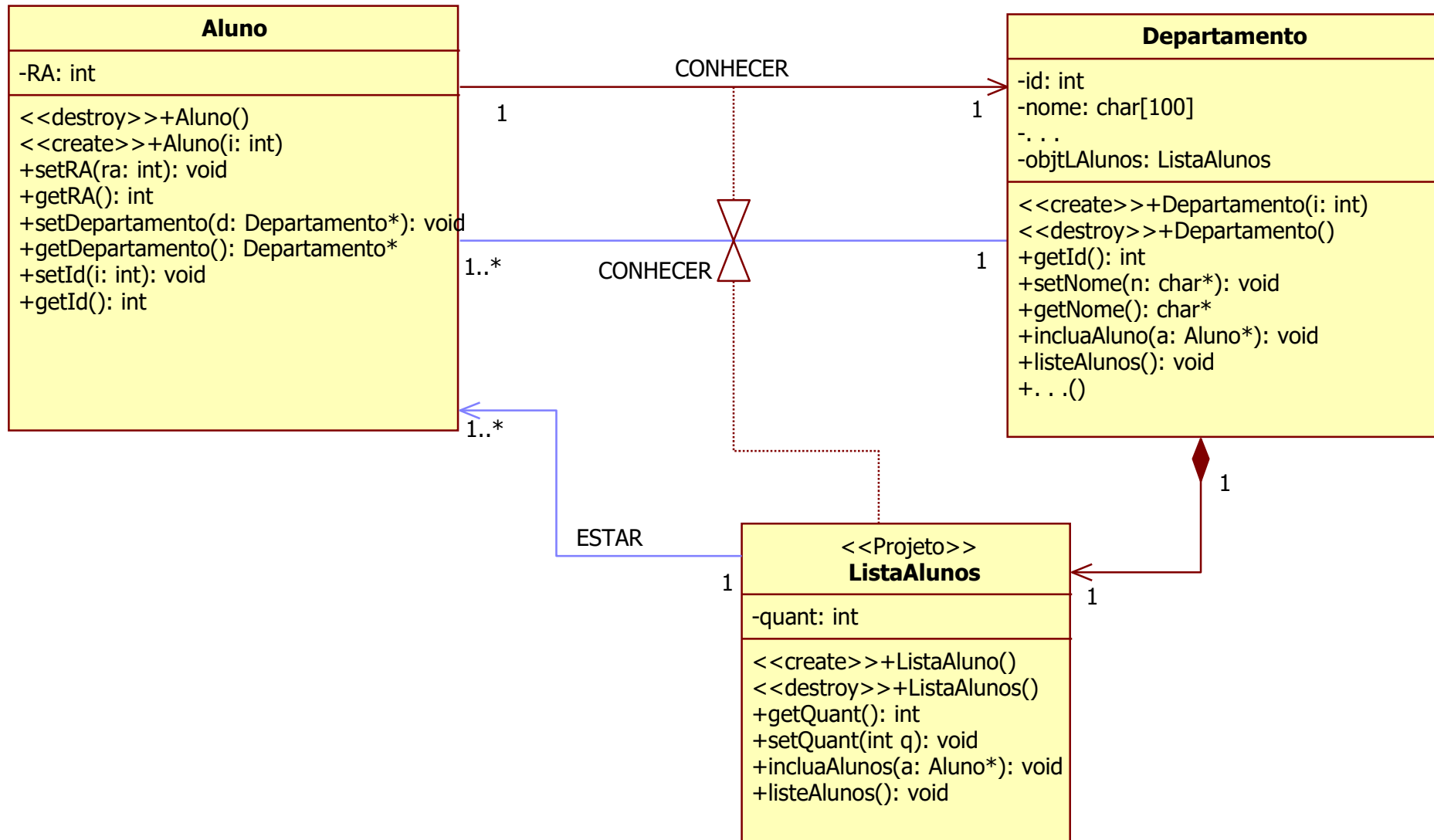
## Composição (*Agregação Forte*)



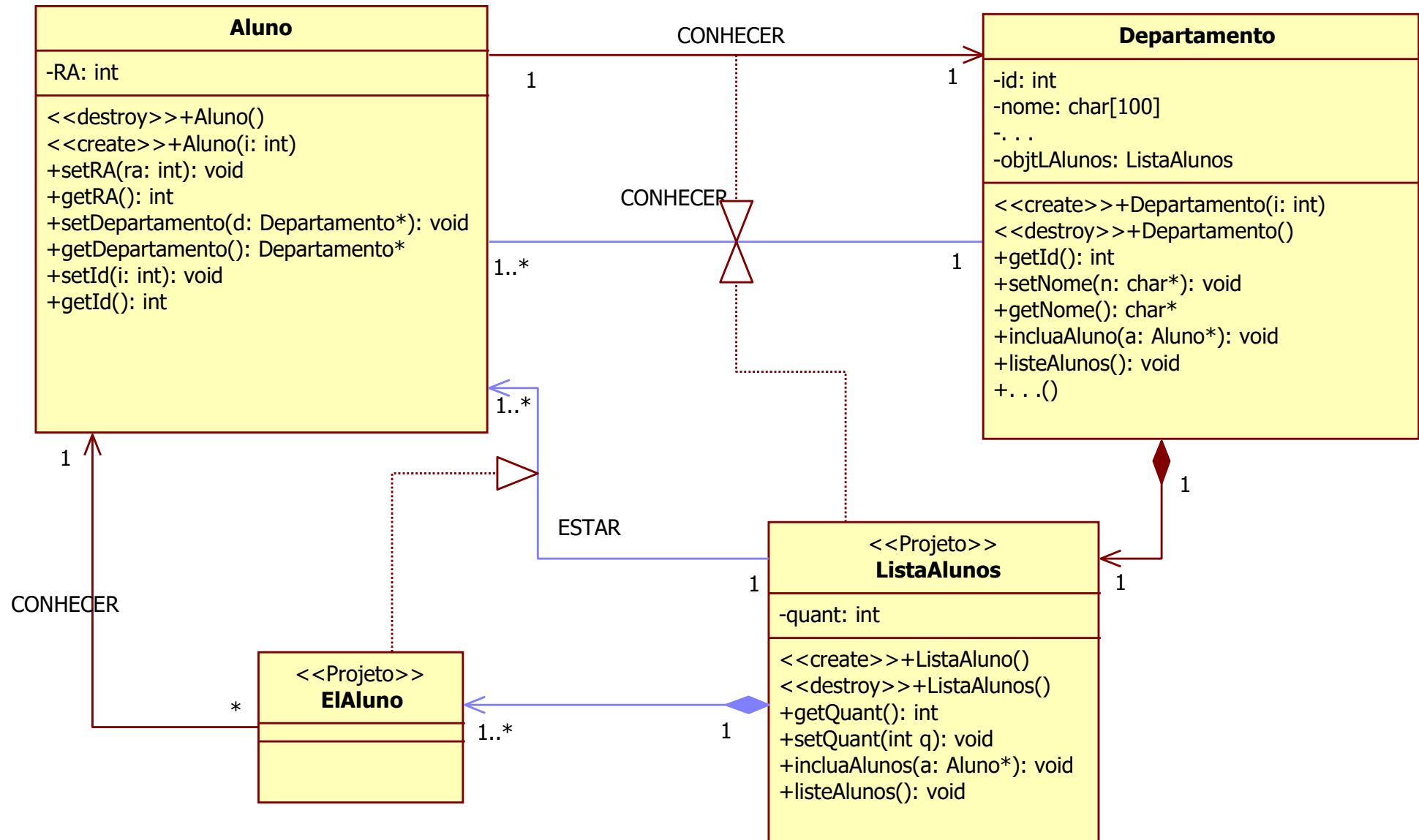
# Diagrama de Classe (de “Projeto”) Estereótipo



# Diagrama de Classe (de “Projeto”) Realização

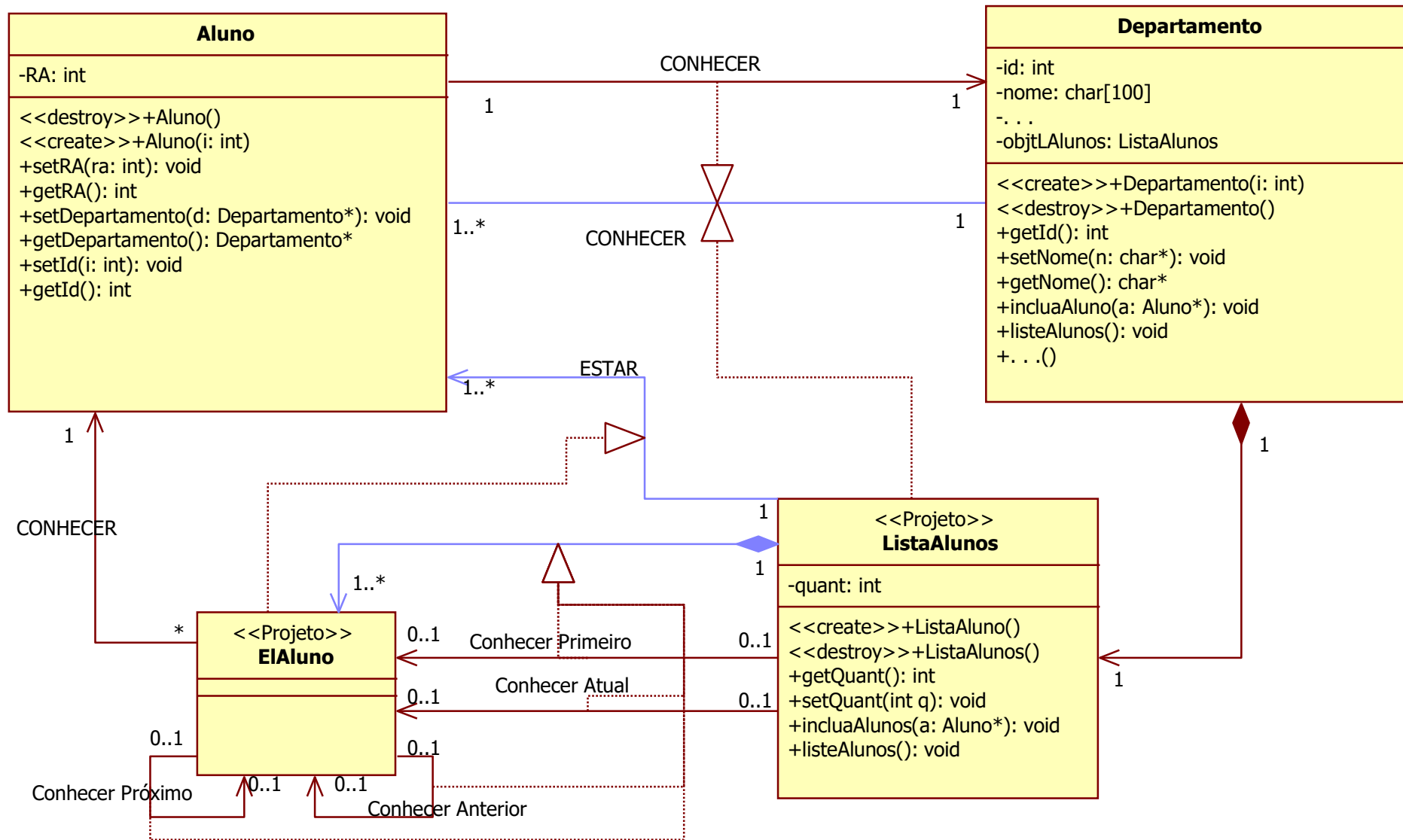


# Diagrama de Classe (de “Projeto”) Realização



# Diagrama de Classe (de “Projeto”)

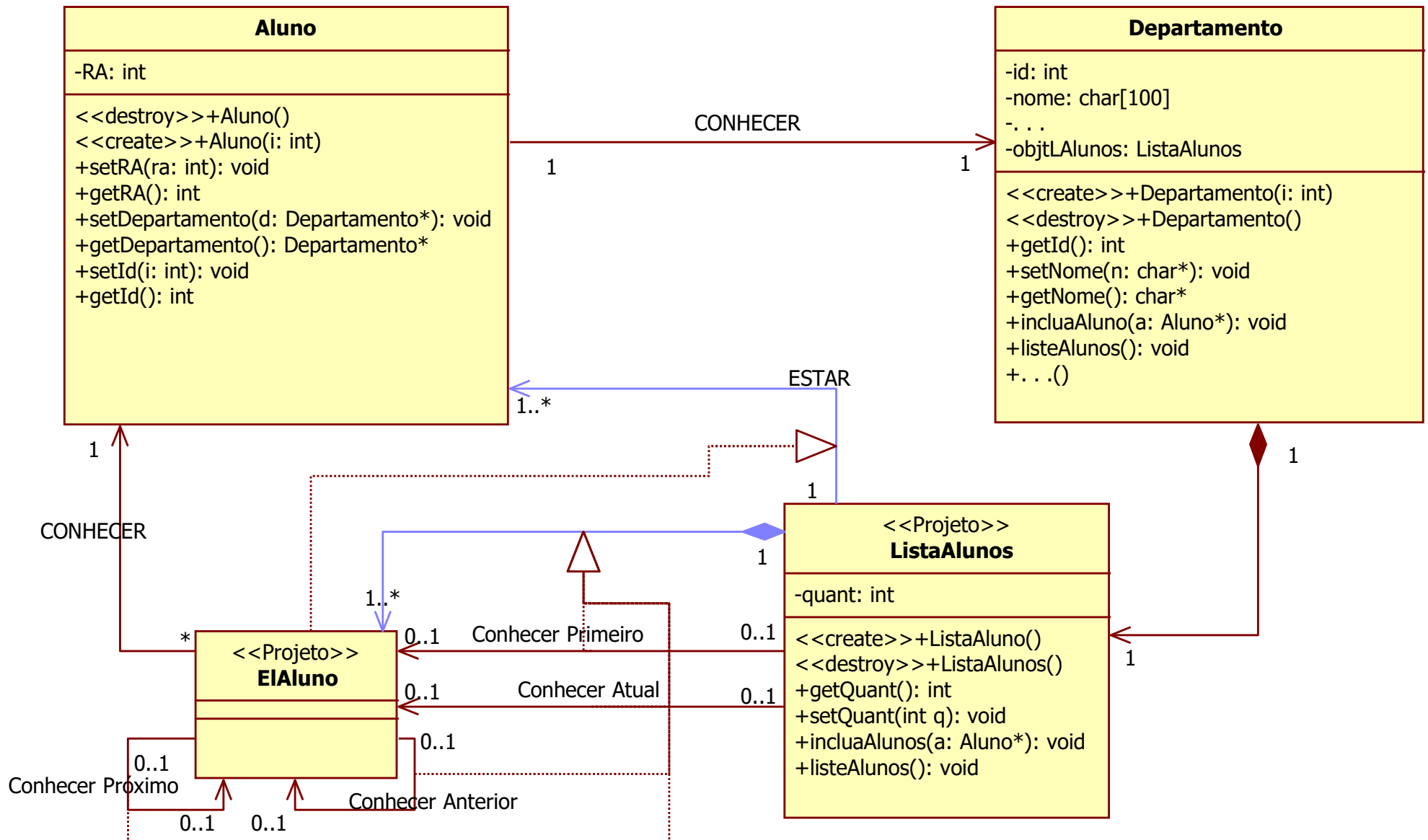
## “Refinando Realização” – “Relação Reflexiva”





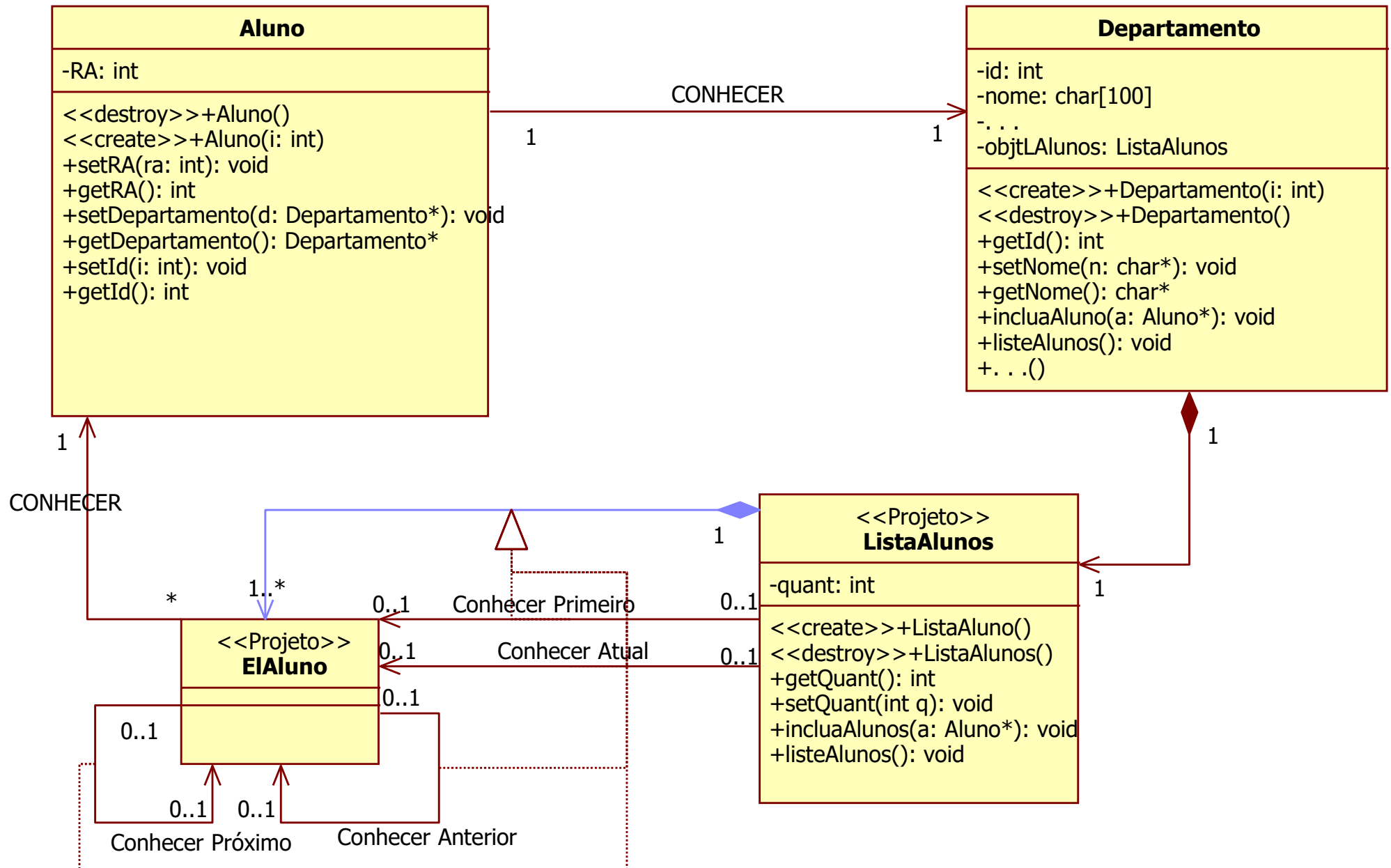
# Diagrama de Classe (de “Projeto”)

## “Projeto – ‘Eliminando’ Análise” - 1



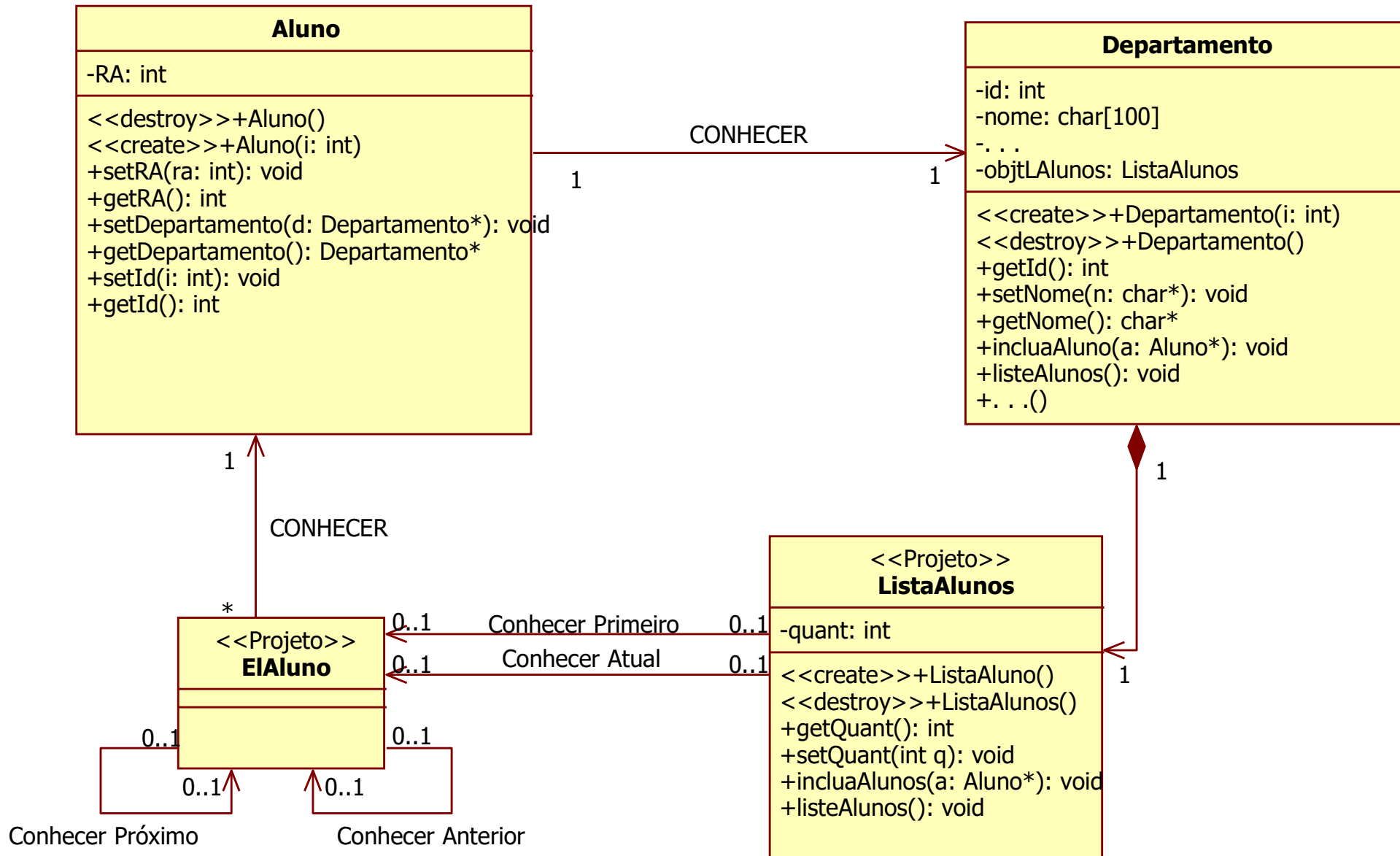
# Diagrama de Classe (de “Projeto”)

## “Projeto – ‘Eliminando’ Análise” - 2



# Diagrama de Classe (de “Projeto”)

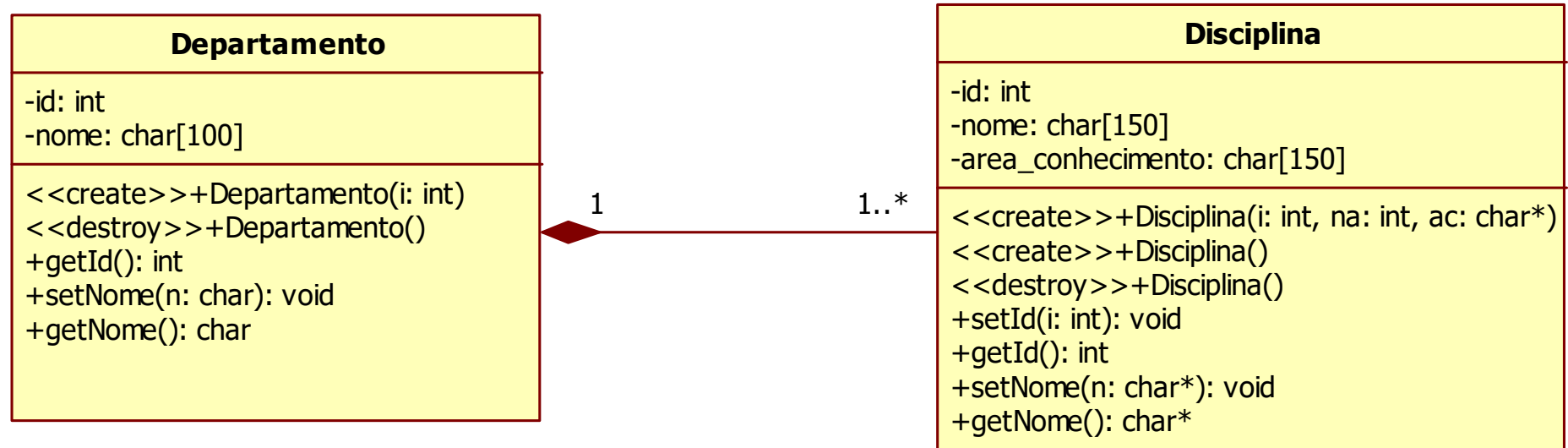
## “Projeto – ‘Eliminando’ Análise” - 3



# Diagrama de Classe (de “Análise”)

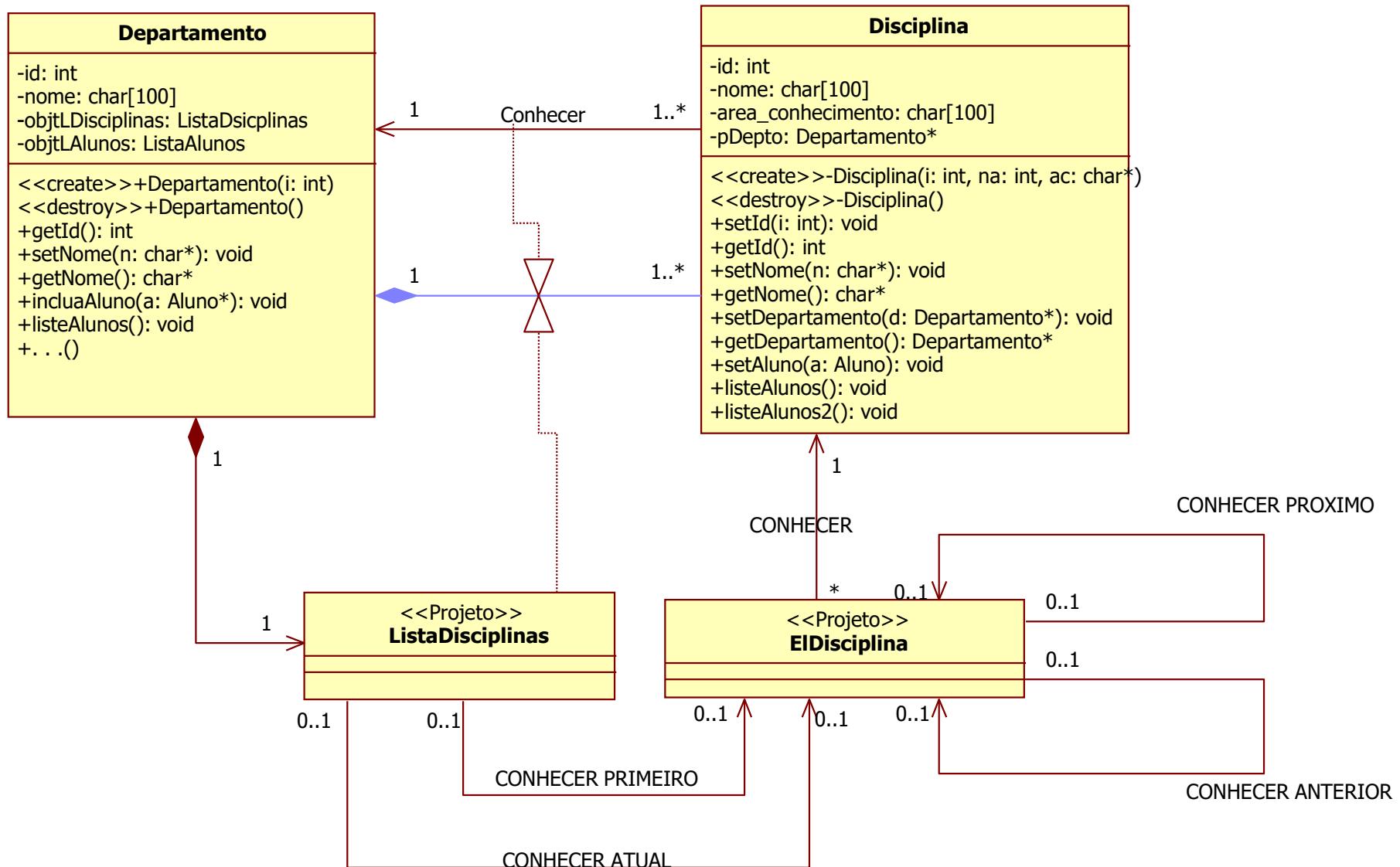
## Composição (*Aggregação Forte*)

---



# Diagrama de Classe (de “Projeto”)

## Composição (Agregação Forte)



# StarUML

- Versão usada da Ferramenta CASE

## STAR UML:

– Versão/Version 5.0.2.1570

- <https://softchamp.com/download-staruml/download>
- \* <https://softchamp.com/download-staruml>
- <https://www.updatestar.com/pt/directdownload/staruml/2509899>
- <https://staruml.informer.com/5.0/>