

# **Introdução ao Desenvolvimento Orientado a Objetos**

**Introdução à Orientação a  
Objetos**

# Programação Orientada a Objetos

- ▷ Programação orientada a objetos é um método de implementação no qual programas são organizados como coleções cooperativas de objetos, sendo que cada um desses objetos representa uma instância de uma classe. Essas classes são membros de uma hierarquia de classes unidas via relações de herança.

# Programação Orientada a Objetos

- ▷ Partes importantes da definição anterior:
  - Programação orientada a objetos usa objetos, não algoritmos, como peças fundamentais de construção;
  - Cada objeto é uma instância de alguma classe;
  - Classes podem estar relacionadas entre si através de relacionamentos de herança.
- ▷ Se em algum programa, falta qualquer um desses elementos, o mesmo não pode ser orientado a objetos.

# Projeto Orientado a Objetos

- ▷ Projeto orientado a objetos é um método de projeto que **engloba o processo de decomposição orientada a objetos** e uma notação descritiva tanto lógica quanto física bem como modelos estáticos e dinâmicos do sistema em desenvolvimento.

# Projeto Orientado a Objetos

- ▷ Partes importantes da definição anterior:
  - Projeto orientado a objetos leva a uma decomposição orientada a objetos;
  - Utiliza diferentes notações para expressar modelos diferentes da lógica de criação de um sistema, em adição aos aspectos estáticos e dinâmicos do sistema.
    - Modelos lógicos:
      - Estruturas de classes;
      - Estruturas de objetos.
    - Modelos físicos:
      - Arquitetura de modelos;
      - Arquitetura de processos.

# Projeto Orientado a Objetos

- ▷ O suporte para a decomposição orientada a objetos é o que faz o projeto orientado a objetos diferente do projeto estruturado.
- ▷ Para estruturar sistemas logicamente:
  - o projeto orientado a objetos utiliza abstrações de classes e objetos.
  - o projeto estruturado utiliza abstrações algorítmicas.

# Análise Orientada a Objetos

- ▷ A análise orientada a objetos é um método de análise que **examina requisitos pela perspectiva de classes e objetos** encontrados no vocabulário do domínio do problema.

# Análise, Projeto e Programação Orientada a Objetos

- ▷ Os produtos da análise orientada a objetos servem como modelos pelos quais é possível iniciar o projeto orientado a objetos;
- ▷ Os produtos do projeto orientado a objetos servem como plano para a implementação completa de um sistema utilizando a programação orientada a objetos.



# Orientação a Objetos

- ▷ A programação orientada a objetos se baseia na **composição e interação** entre diversas unidades de software chamadas **de objetos**.
- ▷ Nós humanos estamos sempre identificando objetos ao nosso redor. Nós damos nomes para esses objetos e, de acordo com as suas características, classificamos esses objetos em grupos, ou seja, classes.

# O que é isso?

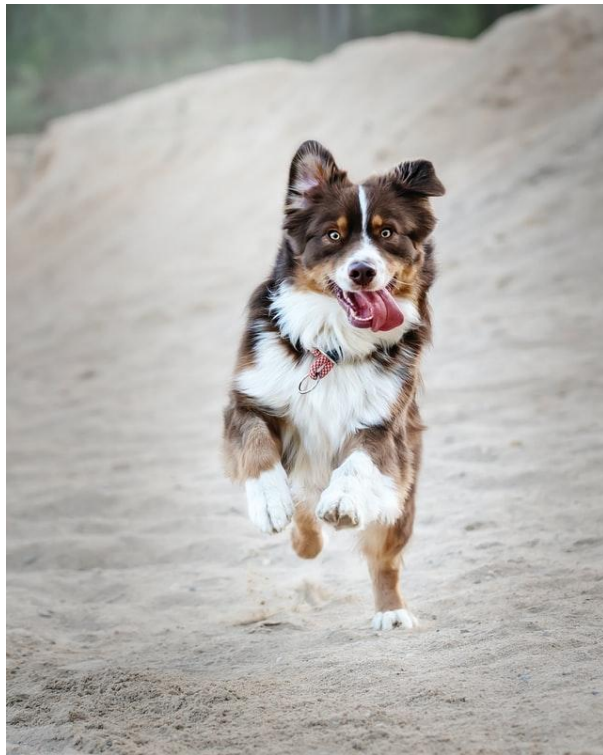


Photo by Joséphine Menge on Unsplash

# O que é isso?



Photo by Alexandru Rotariu on Unsplash

# O que é isso?



Photo by Joe Caione on Unsplash

# O que é isso?



Photo by Ярослав Алексеенко on Unsplash

# O que é isso?



Photo by Luke Stackpoole on Unsplash

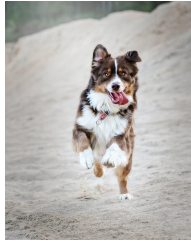


# O que é isso?



Photo by Jacques Bopp on Unsplash

# Cachorro



# Casa





# Orientação a Objetos

- ▷ A classe Cachorro define **atributos** e **comportamentos**. Como atributos da classe Cachorro podemos considerar, por exemplo: **nome**, **cor do pelo** e **raça**. Como comportamento podemos considerar: **latir** e **correr**.

# Orientação a Objetos

- ▷ Para não confundirmos o real significado da palavra comportamento, vamos evitá-la nesse primeiro momento. Na orientação a objetos estamos mais interessados em saber o estado (os atributos) de um objeto e enviar mensagens para esses objetos. Então vamos pensar que a classe cachorro define atributos e mensagens.

# Orientação a Objetos

- ▷ Logo, para facilitar o entendimento, vamos pensar em latir e correr como ordens e não como comportamento. Ao término dessa aula, vamos notar que mensagens não são somente ordens, mas para entendermos melhor os conceitos apresentados até este ponto, vamos considerar que essas mensagens são ordens dadas ao cachorro. Dessa forma, latir e correr são ordens.

# Orientação a Objetos

- ▷ Podemos melhorar o exemplo? Sim! Vamos continuar considerando os atributos nome, raça e cor do pêlo e, como mensagens, vamos considerar sentar, deitar e fingir de morto.
- ▷ Um objeto é capaz de armazenar estados, reagir a mensagens enviadas a ele mesmo e enviar mensagens a outros objetos. Um objeto armazena estados utilizando, para isso, os seus atributos.

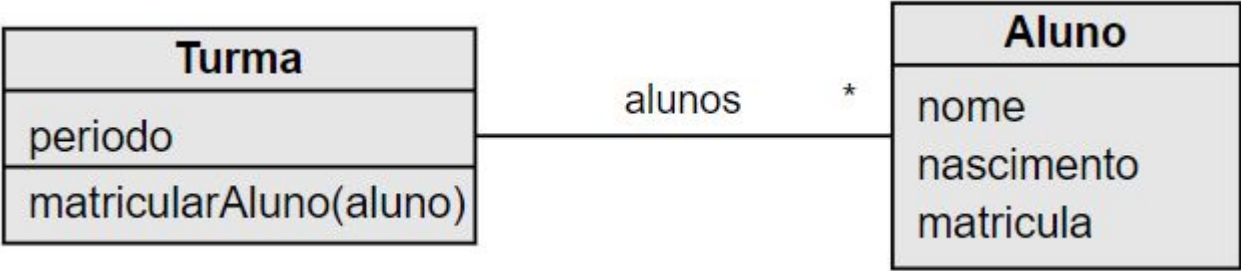
# Orientação a Objetos

- ▷ Agora saindo do mundo natural e pensando em nossos programas, poderíamos criar uma classe **Aluno** para representar um aluno em um sistema. Essa classe poderia ter uma série de **atributos**, dentre os quais destacamos **nome, data de nascimento e matrícula**.

# Orientação a Objetos

- ▶ Além de aluno, nós poderíamos criar uma **classe Turma**, que possuiria os **atributos período e alunos**. Nós poderíamos **matricular um aluno em uma Turma**, logo podemos ter a mensagem **MatricularAluno** na classe Turma. Ao enviar a mensagem MatricularAluno, uma instância da classe Aluno deve ser passada.
- ▶ Como podemos representar isso?

# Orientação a Objetos



# Classe

- ▷ Uma classe define um tipo de objeto.
- ▷ Uma classe é uma estrutura heterogênea, que contém:
  - Uma coleção de uma ou mais variáveis relacionadas (atributos).
  - Uma coleção de procedimentos ou funções (métodos).



# Objeto

- ▷ Um objeto:
  - É criado (instanciado) a partir de uma classe.
  - Contém uma área de memória que contempla o espaço reservado para os diferentes atributos definidos em sua classe.
  - Uma coleção de métodos que podem acessar os atributos definidos.

# Classes e Objetos

- ▷ Vamos melhorar o nosso exemplo incluindo mais classes e relacionamentos.

# Classe

## Aluno

- ▷ matricula
- ▷ nome
- ▷ nascimento

# Objetos

### Aluno

matricula: **123**  
nome: **Bruno**  
nascimento: **01/02/03**

### Aluno

matricula: **124**  
nome: **Daniela**  
nascimento: **02/03/03**

### Aluno

matricula: **125**  
nome: **Fábio**  
nascimento: **03/04/03**

# Classe

## Professor

- ▷ matricula
- ▷ nome
- ▷ ctps

# Objetos

## **Professor**

matricula: 1  
nome: **Cormen**  
ctps: **123456**

## **Professor**

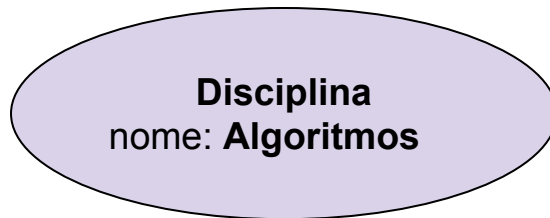
matricula: 3  
nome: **Lecum**  
ctps: **123458**

# Classe

## Disciplina

▷ nome

# Objetos

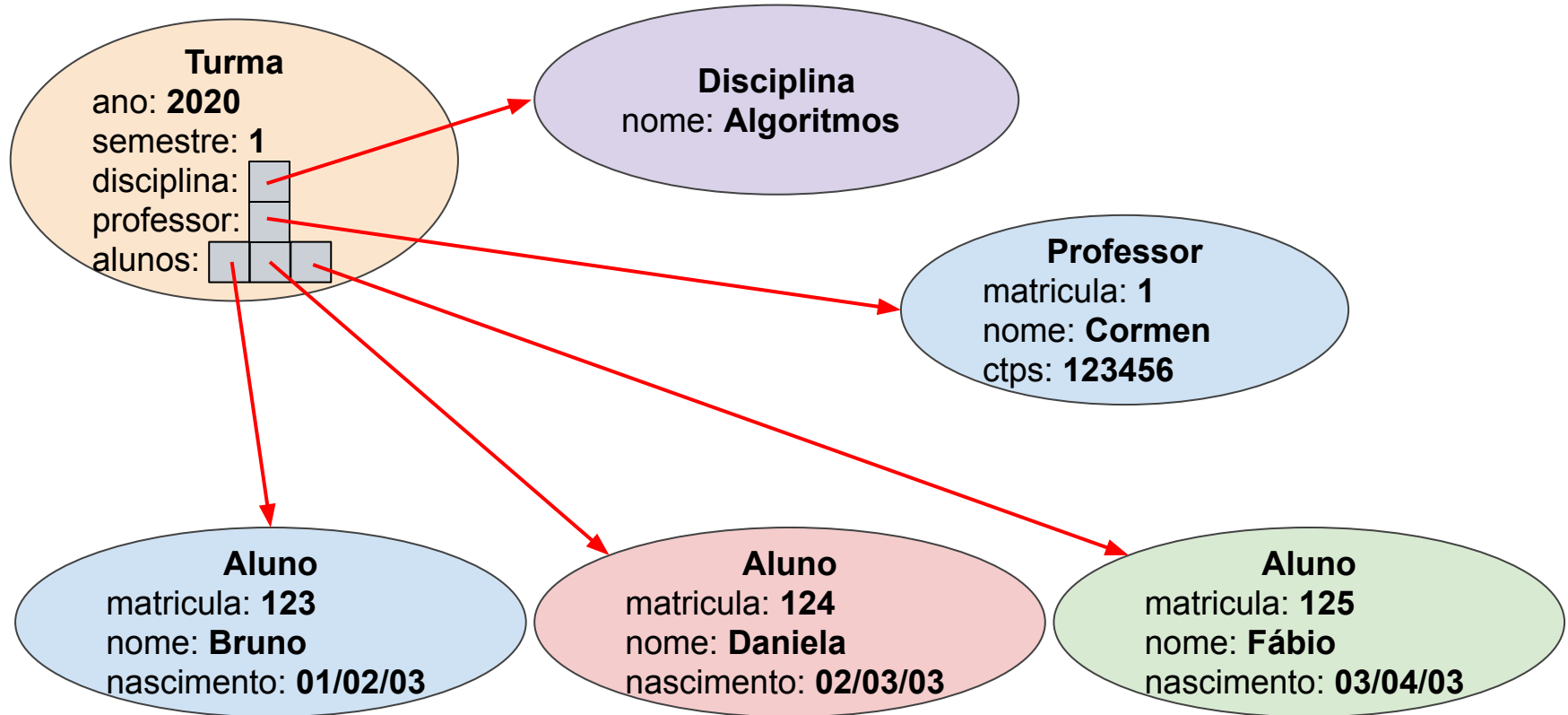


# Classe (Atributos)

## Turma

- ▷ ano
- ▷ semestre
- ▷ disciplina
- ▷ professor
- ▷ alunos

# Objetos



# Classe (Operações)

## Turma

- ▷ matricular(Aluno aluno)



# Objetos

## Aluno

matricula: **123**

nome: **Bruno**

nascimento: **01/02/03**

## Aluno

matricula: **124**

nome: **Daniela**

nascimento: **02/03/03**

## Aluno

matricula: **125**

nome: **Fábio**

nascimento: **03/04/03**

## Turma

ano: **2020**

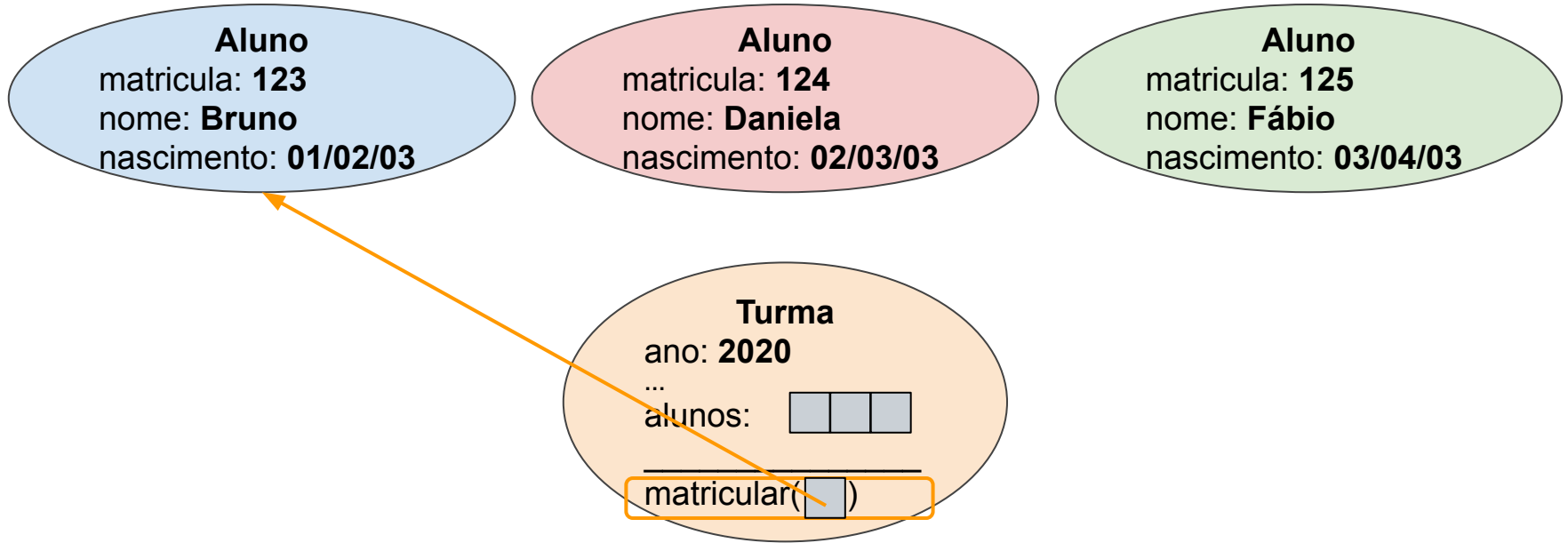
...

alunos:

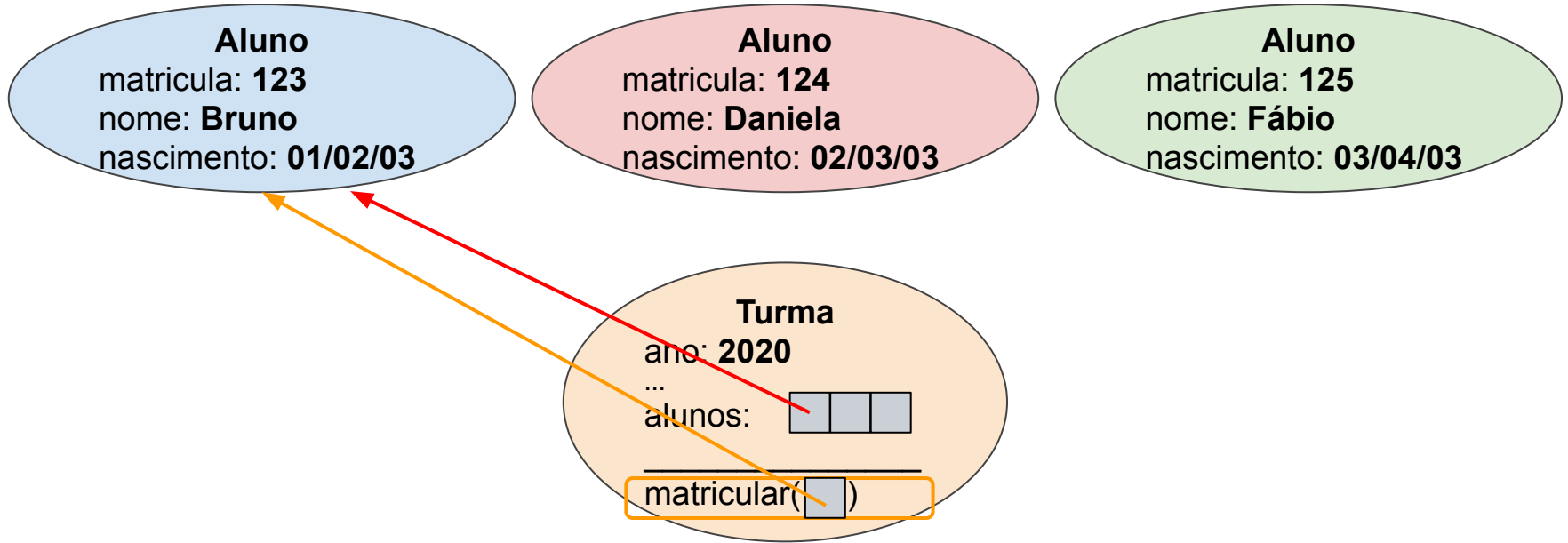
---

matricular()

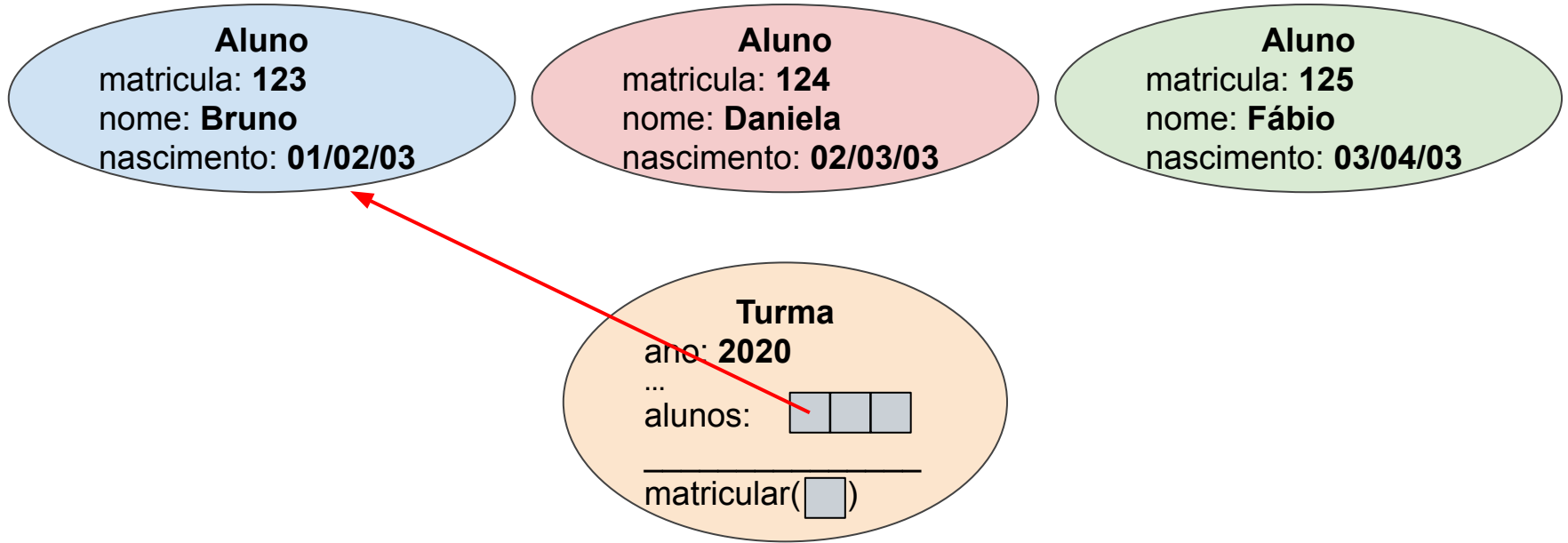
# Objetos



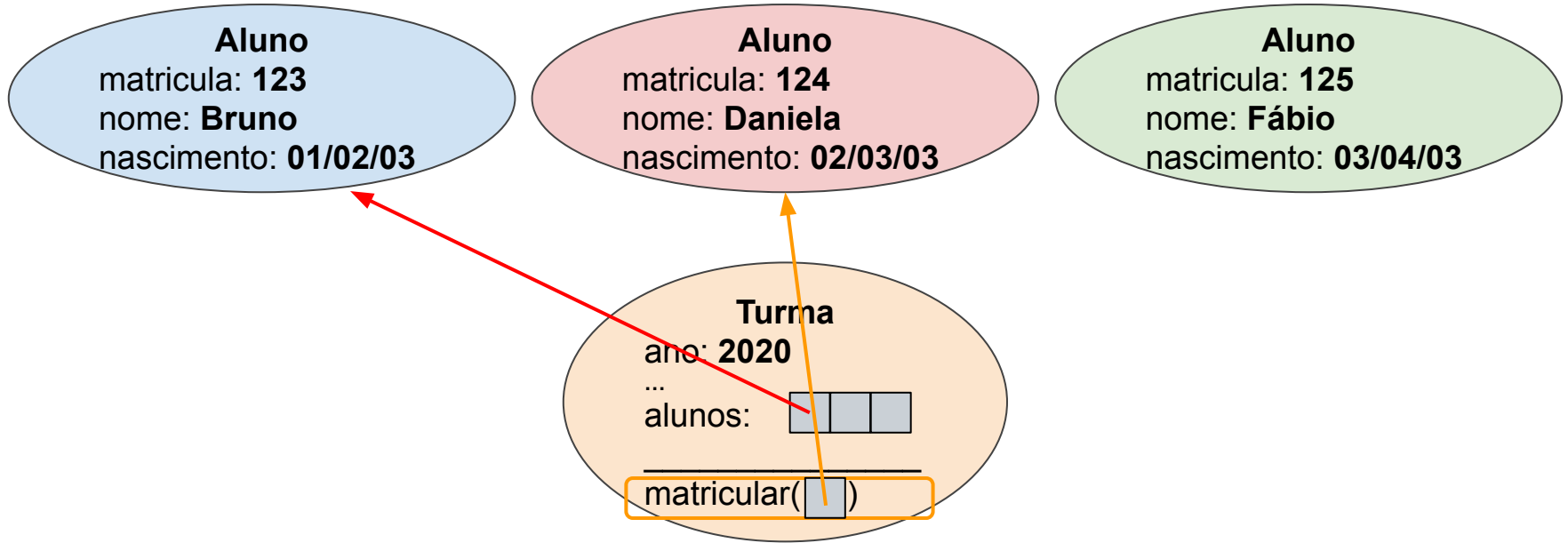
# Objetos



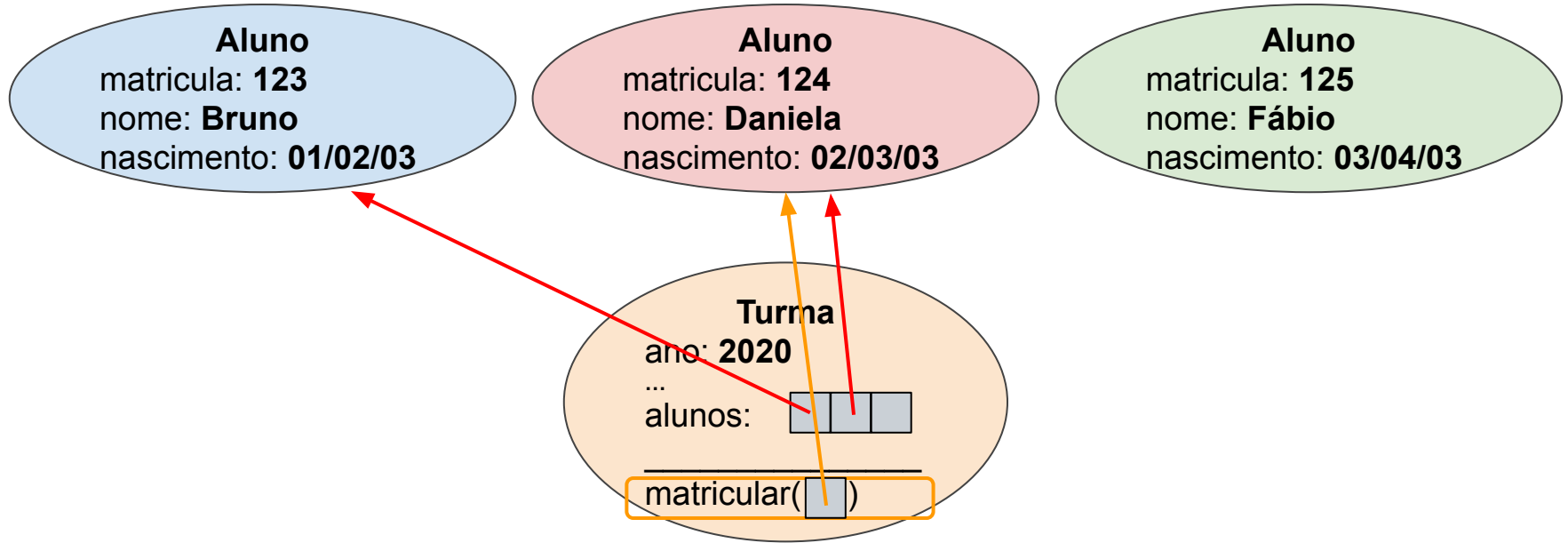
# Objetos



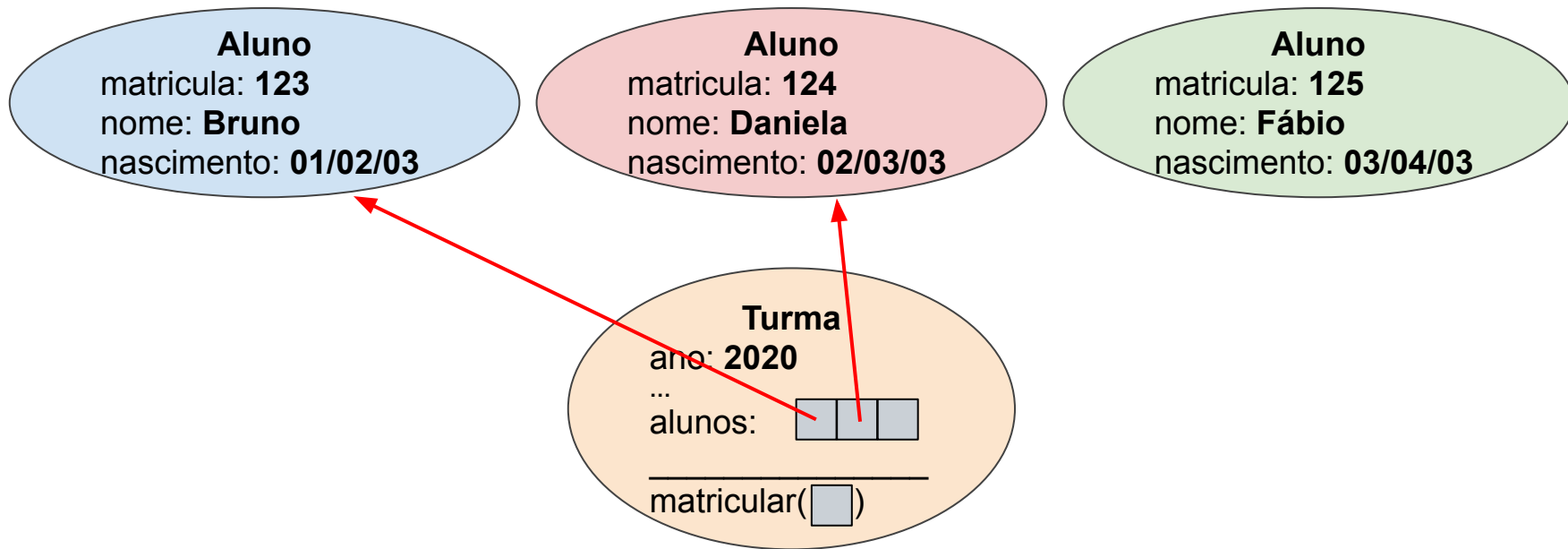
# Objetos



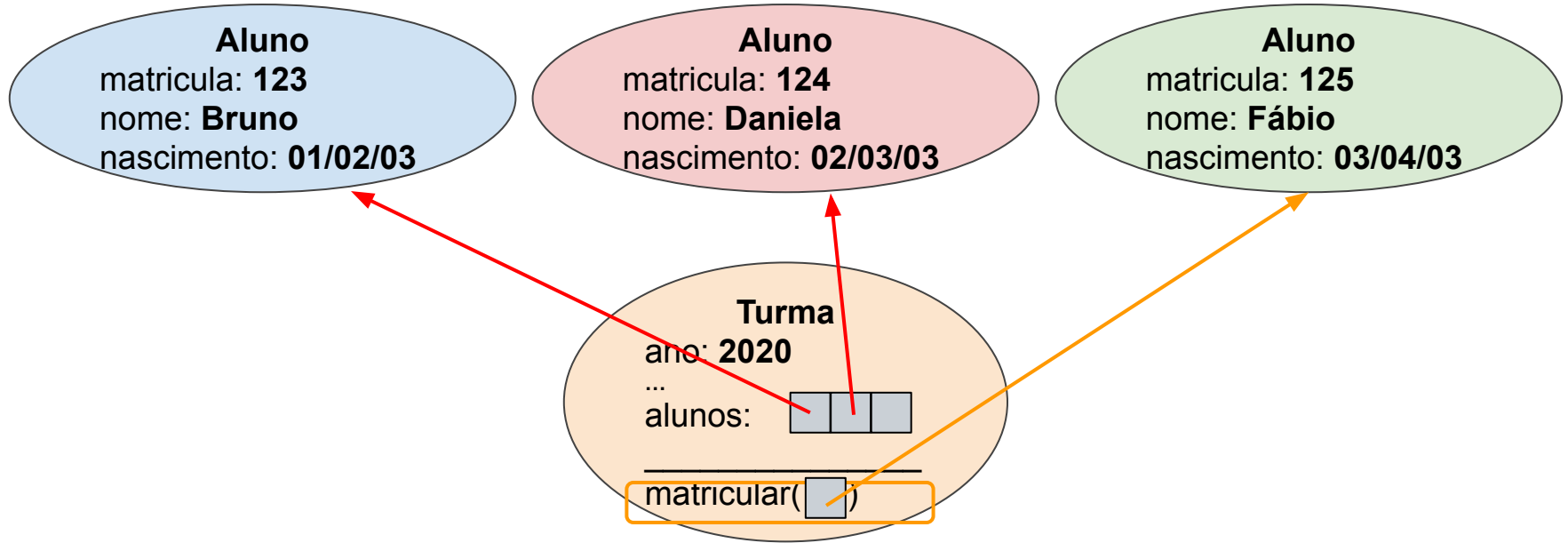
# Objetos



# Objetos

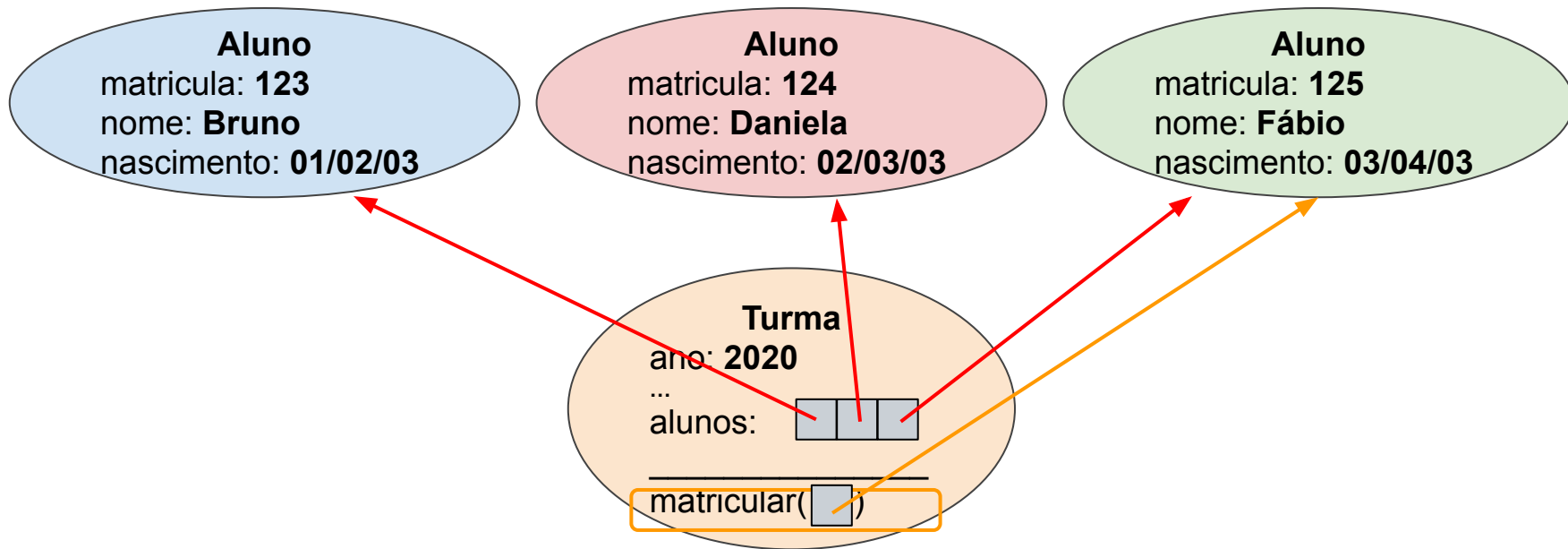


# Objetos

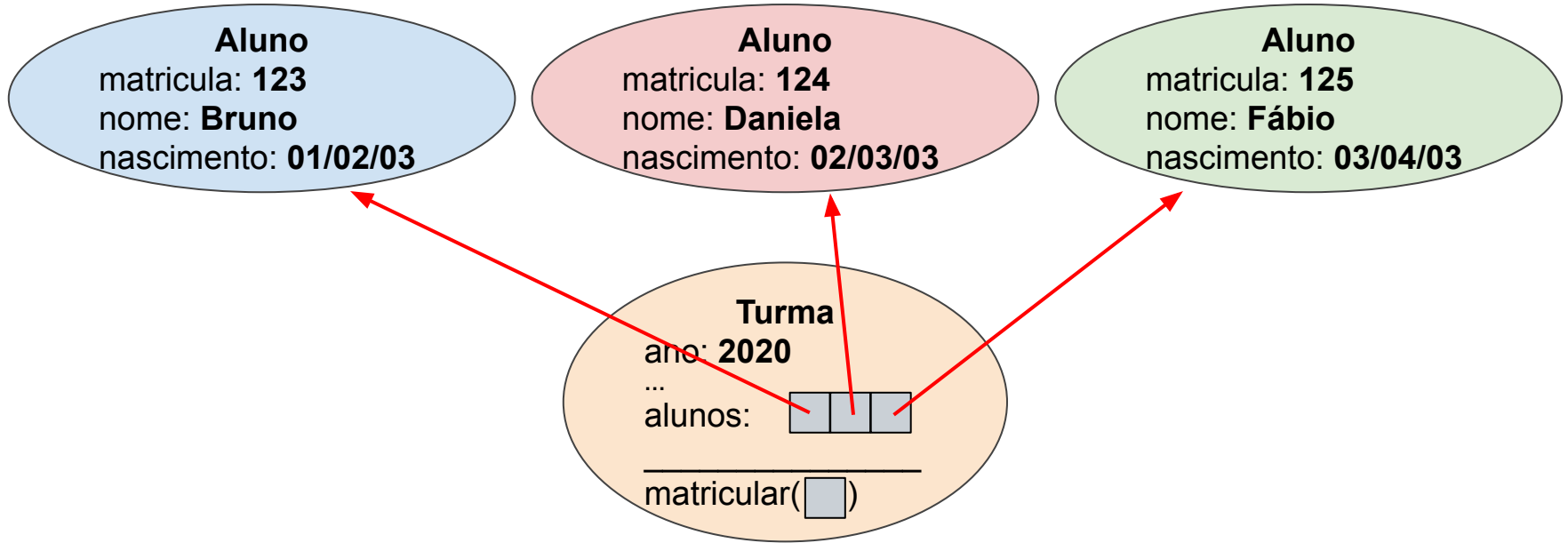




# Objetos



# Objetos



# Fim de material

## Dúvidas?