



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Princípios da Orientação a Objetos

QXD0007 - Programação Orientada a Objetos

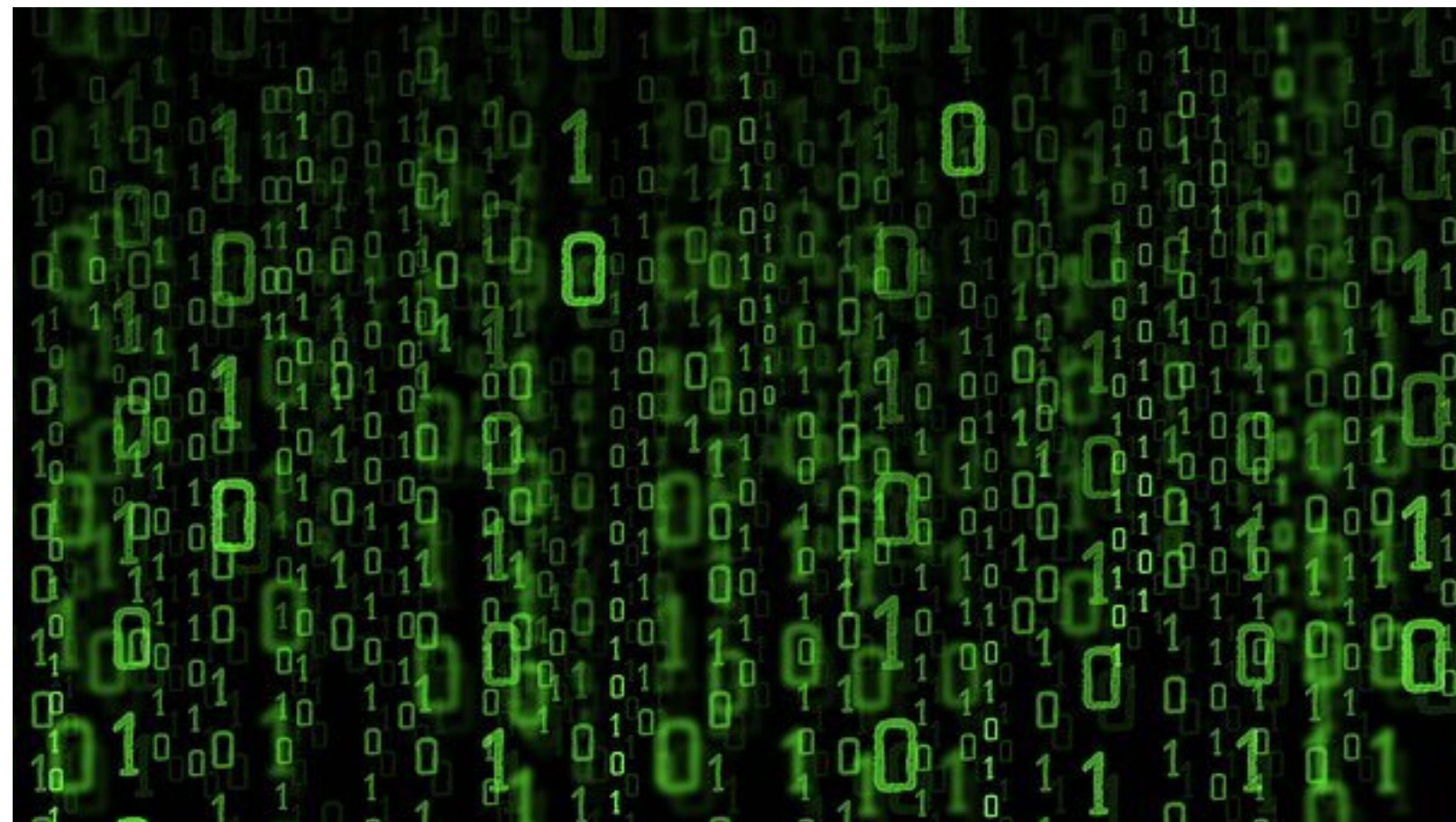
Prof. Bruno Góis Mateus (brunomateus@ufc.br)

Conteúdo

- Contextualização
- O paradigma da Orientação a Objetos
- Abstração
- Encapsulamento
- Polimorfismo
- Herança

Contextualização

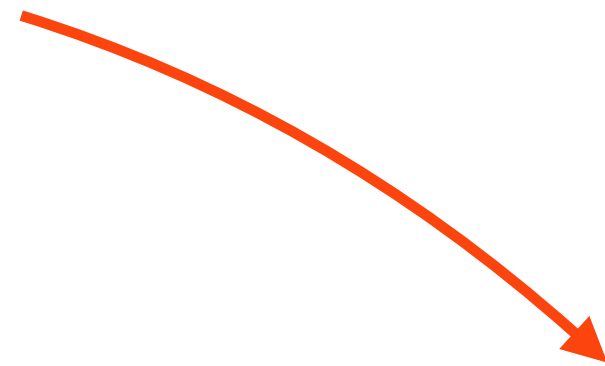
Contextualização



```
FE30- 20 B4 FC 90 F7 60 B1 3C
#FDEDL

FDEFD- 6C 36 00 JMP ($0036)
FDEF0- 09 00 00 CMP #BA0
FDEF2- 90 00 00 BCC $FDF6
FDEF4- 25 55 32 AND $32
FDEF6- 84 04 35 STY $35
FDEF8- 40 04 78 PHA
FDEF9- 20 04 78 JSR $FB78
FDEFD- 60 35 00 PLA
FDEF0- A4 35 00 LDY $35
FDEF2- 00 35 00 RTS
FDEF4- 06 34 00 DEC $34
FDEF6- F0 9F 00 BEQ $FDA3
FDEF8- 0A 16 00 BNE $FE1D
FDEF9- 09 0A 00 CMP #BA
FDEFB- 00 0B 00 BNE $FDC6
FDEFD- 85 31 00 STA $31
FDEF0- A5 3E 00 LDA $3E
FDEF2- 91 40 00 STA ($40),Y
FDEF4- E6 40 00 INC $40
*
```

Contextualização



<https://www.computerhistory.org/timeline/1933/>



Paradigmas de Programação:

A maneira em que organizamos nossas ideias para construir um abstração de um problema do mundo com intuito de utilizar a programação para resolvê-lo.

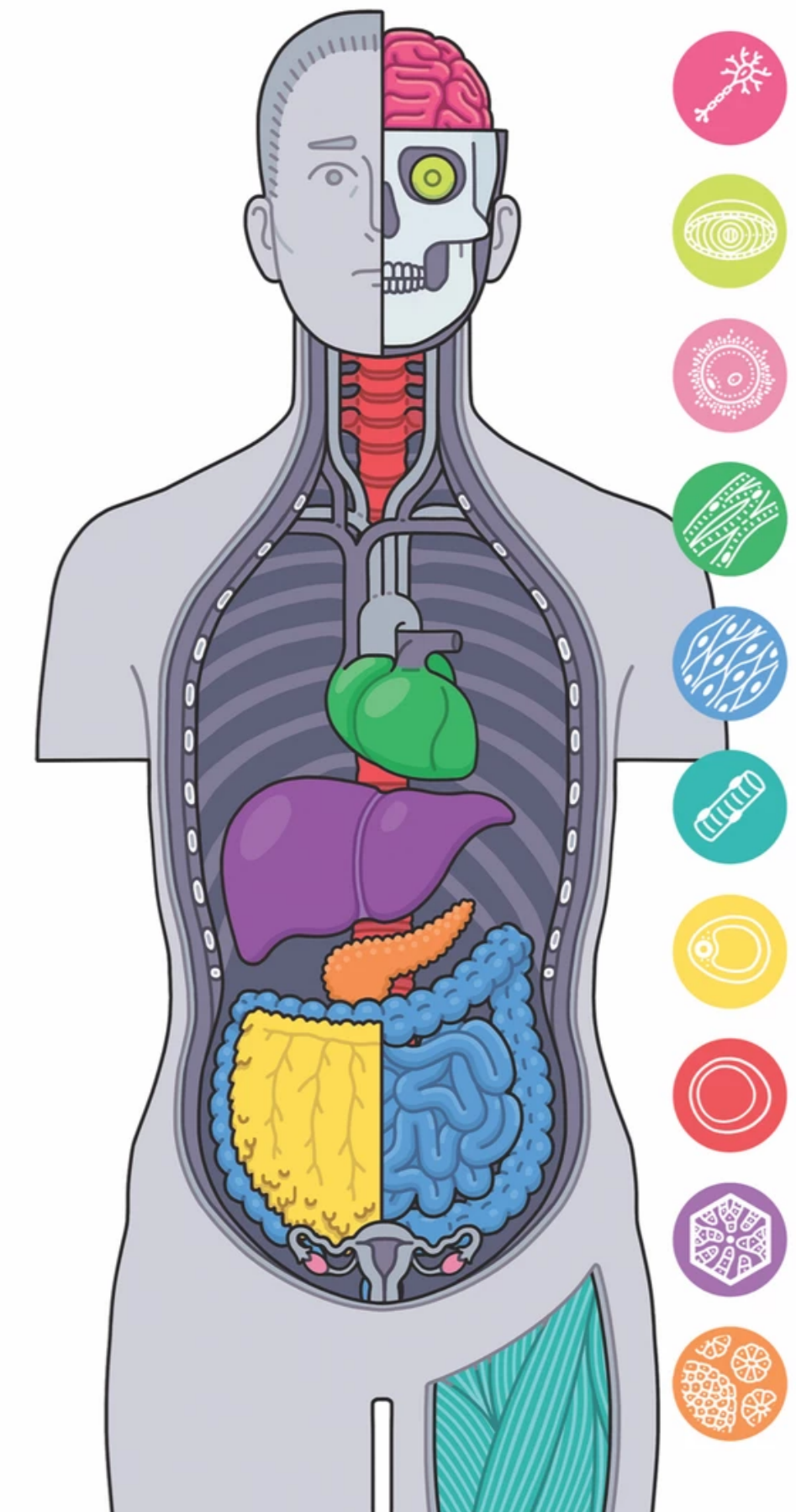
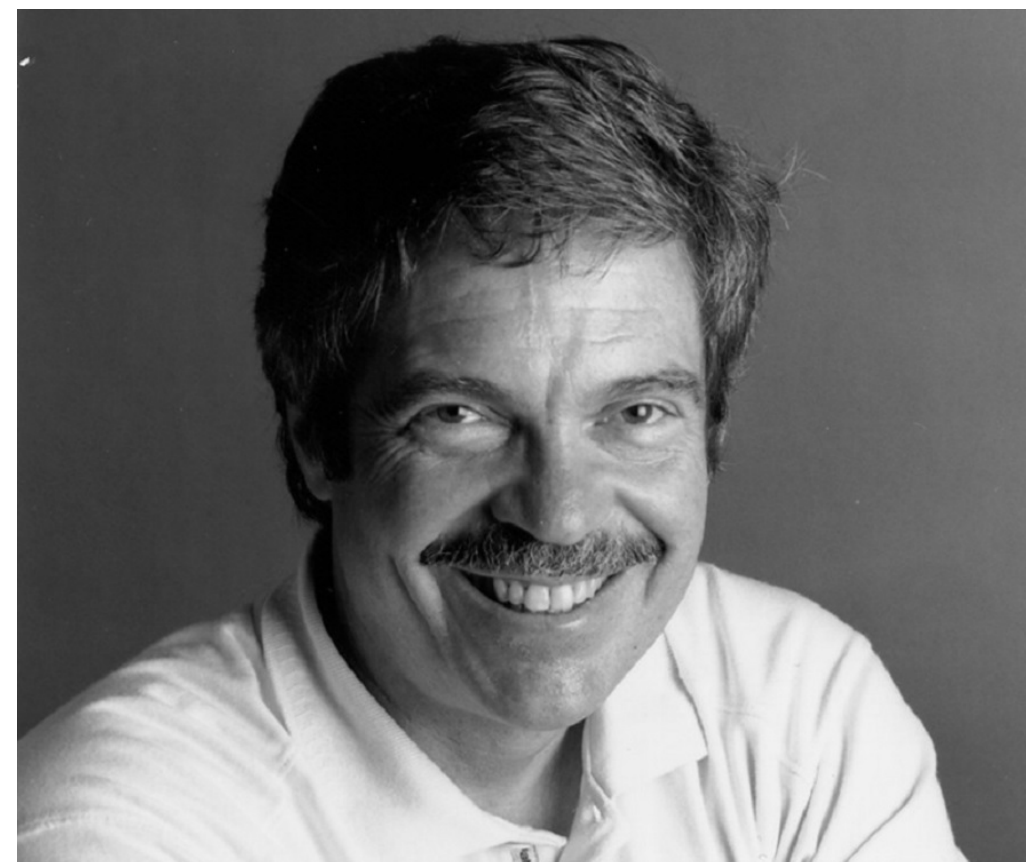
Robert W Floyd: [The Paradigms of Programming \(1978\)](#)

O paradigma da Orientação a Objetos

O paradigma da orientação a objetos

Alan Kay, um dos pais da OO

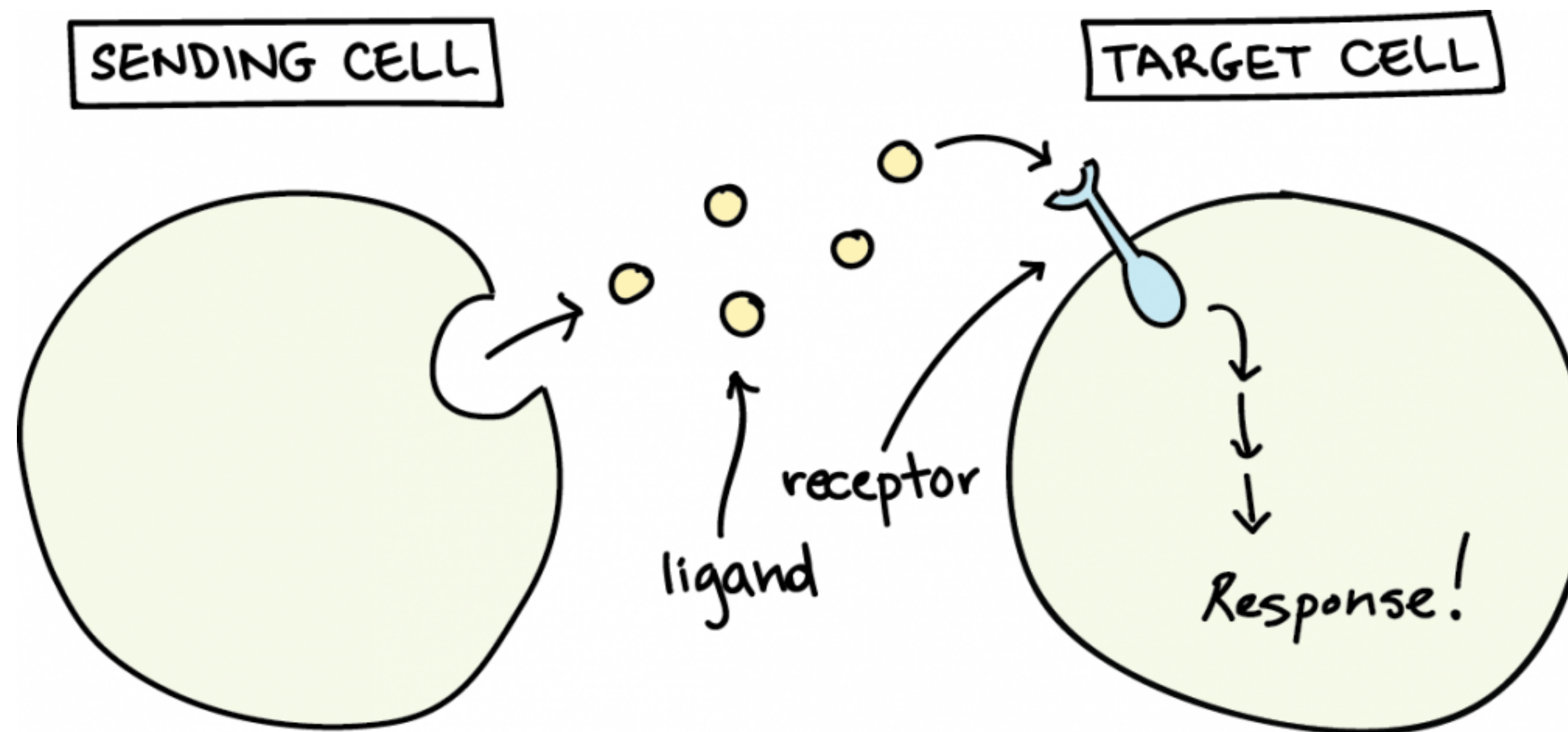
- “ I thought of **objects** being like biological **cells** and/or **individual computers** on a **network**, only able to **communicate with messages** (so messaging came at the very beginning -- it took a while to see how to do messaging in a programming language efficiently enough to be useful).”



Fonte: sciencefocus.com

O paradigma da orientação a objetos

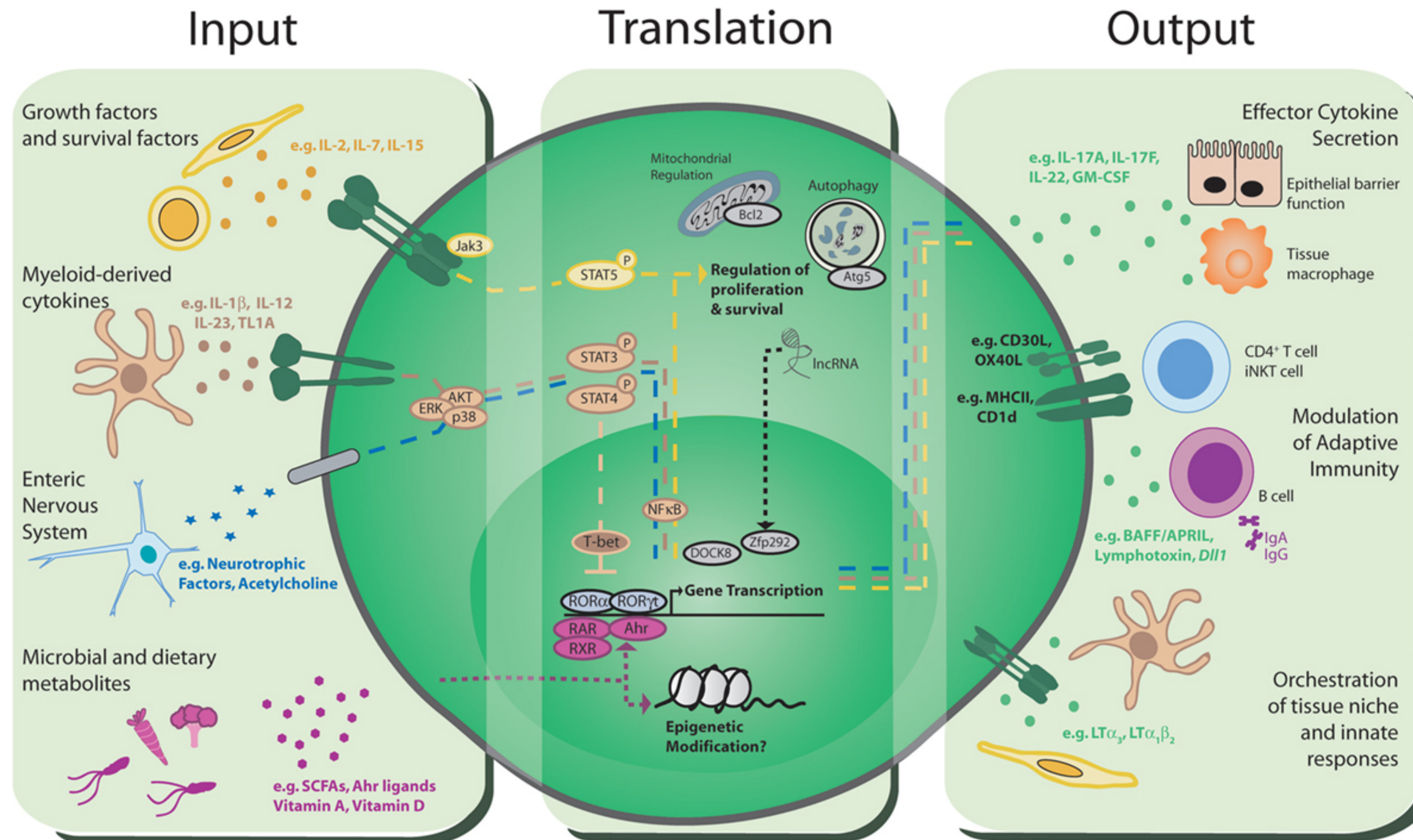
Comunicação via mensagens



Fonte: <https://organismalbio.biosci.gatech.edu/chemical-and-electrical-signals/intro-to-chemical-signaling-and-signal-transduction/>

O paradigma da orientação a objetos

Comunicação via mensagens



Fonte: <https://www.frontiersin.org/articles/10.3389/fimmu.2017.01298/full>



“ O paradigma da orientação a objetos visualiza um sistema de software como uma coleção de agentes interconectados chamados objetos. Cada objeto é responsável por realizar tarefas específicas. É através da interação entre objetos que uma tarefa computacional é realizada.”

Eduardo Bezerra, 2023

O paradigma da orientação a objetos

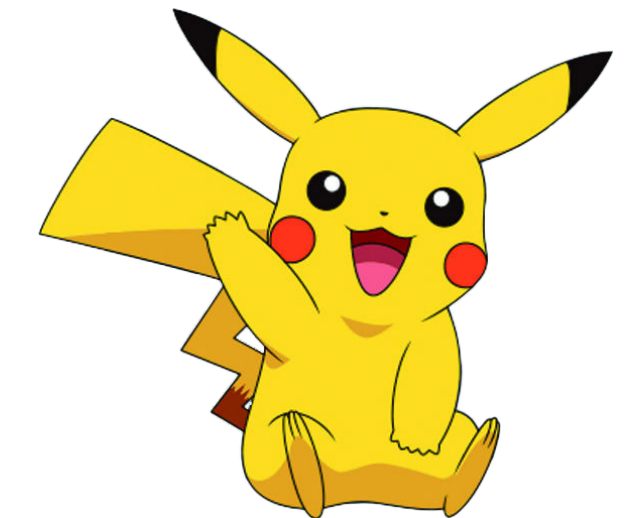
Princípios da Orientação a Objetos

1. Qualquer “*coisa*” é um *objeto*.
2. Objetos realizam *tarefas através da requisição de serviços* a outros objetos.
3. Cada *objeto pertence a uma determinada classe*. Uma classe agrupa objetos similares.
4. Classes são organizadas em *hierarquias*.

O paradigma da orientação a objetos

Objeto

- Alguma coisa que faz sentido no domínio da sua aplicação
- Depende do problema que você está tentando resolver
- Deve favorecer a compreensão do problema
- Fornecer uma base real para a implementação



O paradigma da orientação a objetos

Objeto

- São descritos por meios de **atributos (estados)** e **comportamentos**



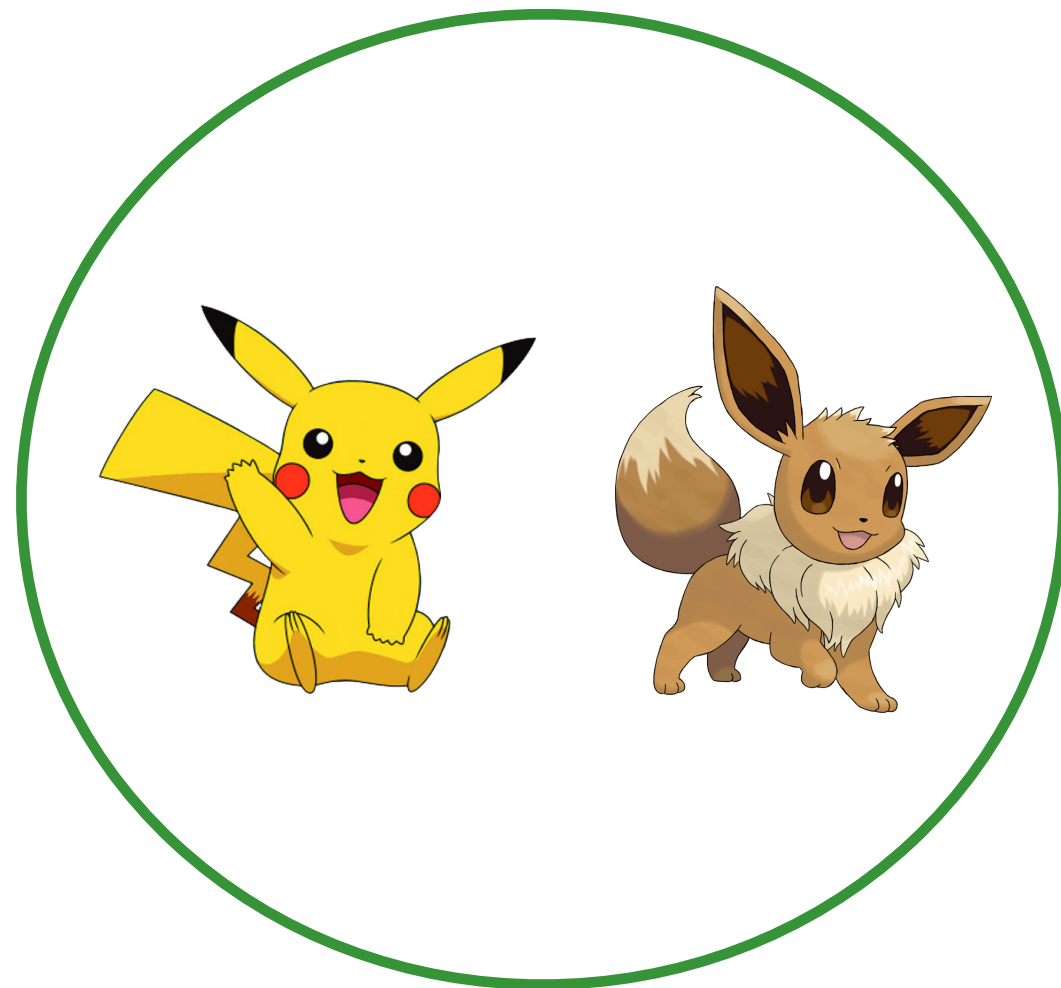


Classe: Outro conceito
inspirado na biologia.
**É uma categoria utilizada na
classificação científica dos
seres vivos, o sistema
taxonômico mais utilizado na
moderna biologia.**
Ex: Mamíferos, Aves e Répteis

O paradigma da orientação a objetos

Classe de objetos

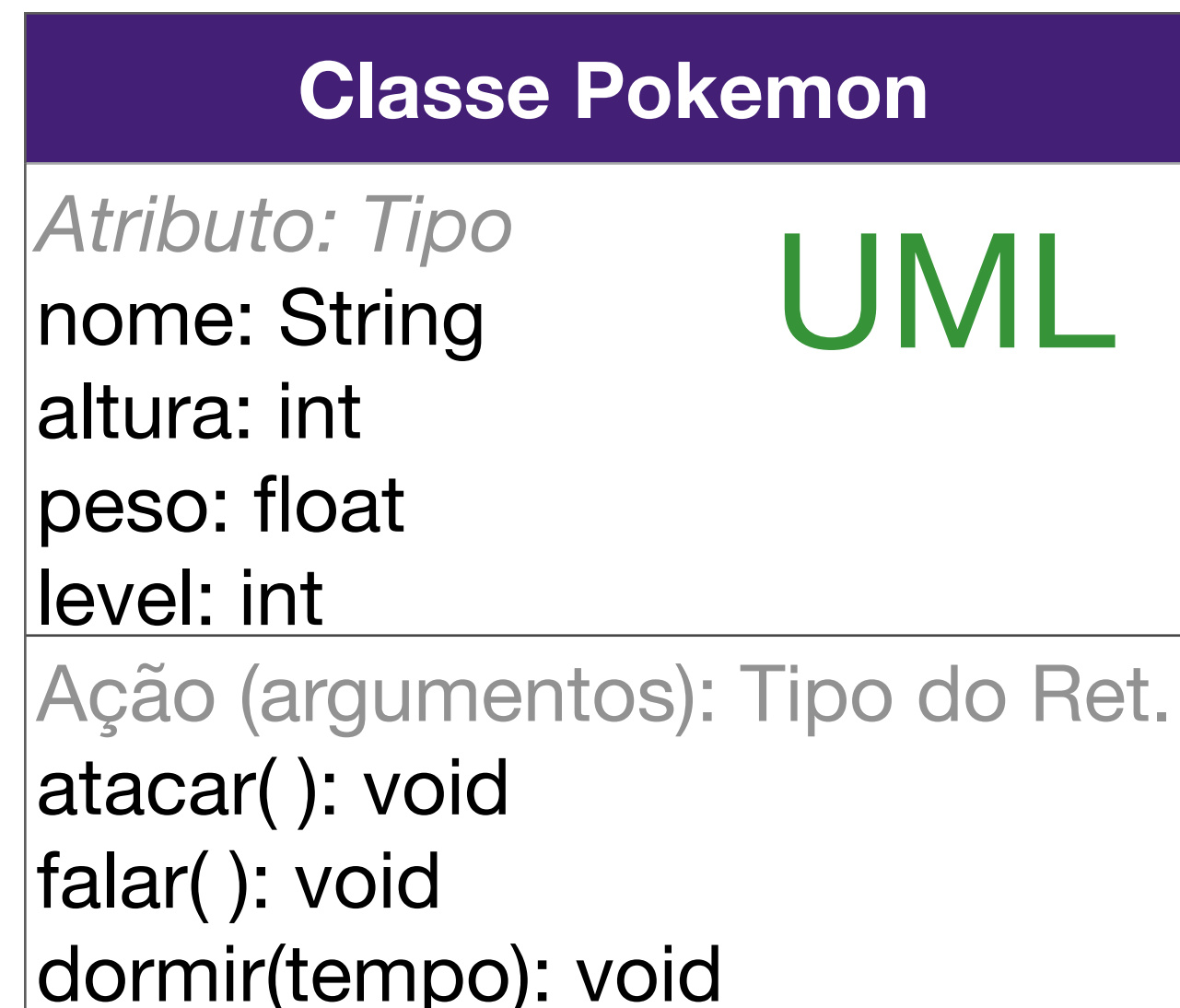
- Grupos de objetos com os mesmos atributos e os mesmos comportamentos



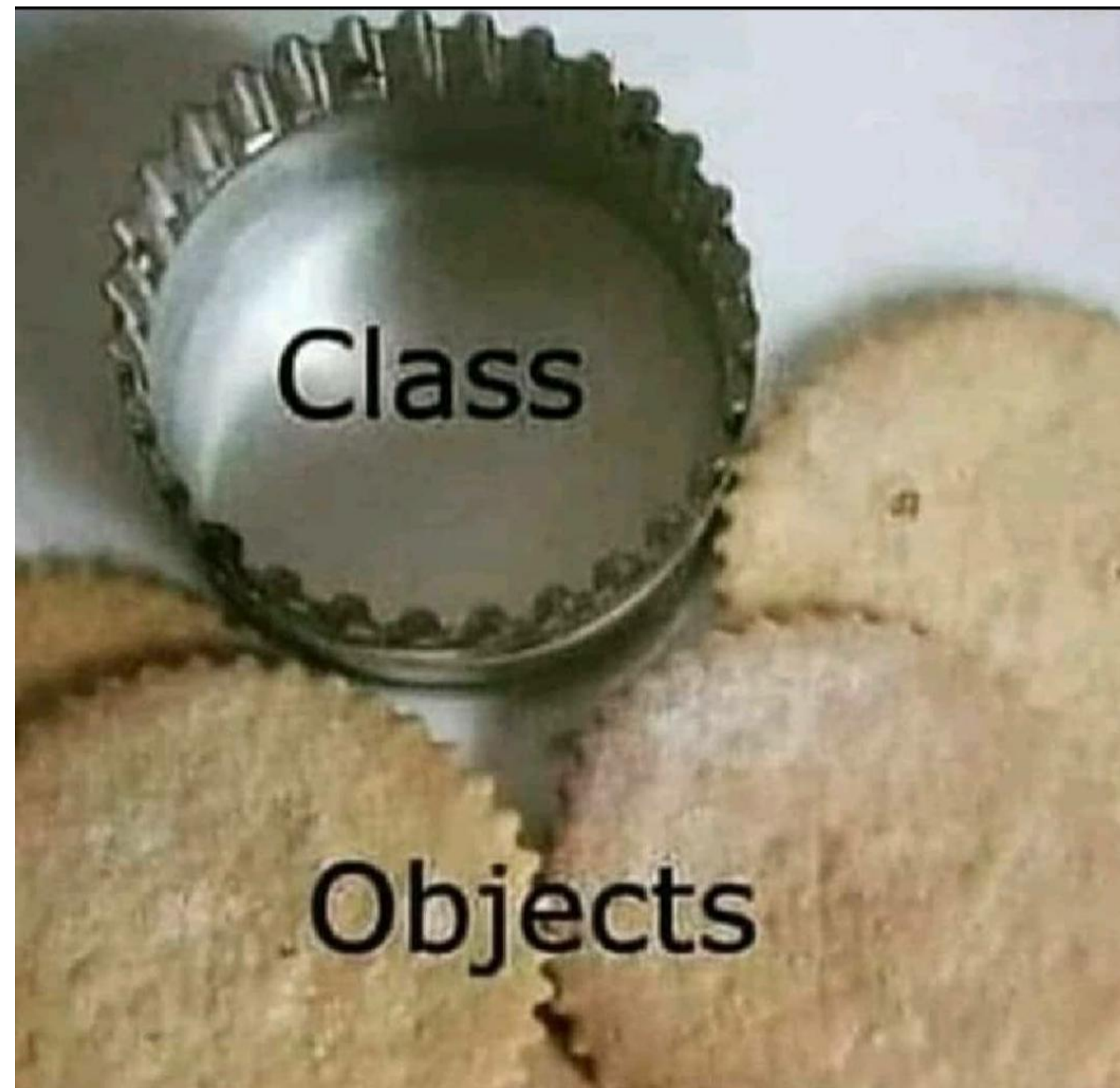
O paradigma da orientação a objetos

Classe de objetos

- São usadas para criar (*instanciar*) objetos



O paradigma da orientação a objetos

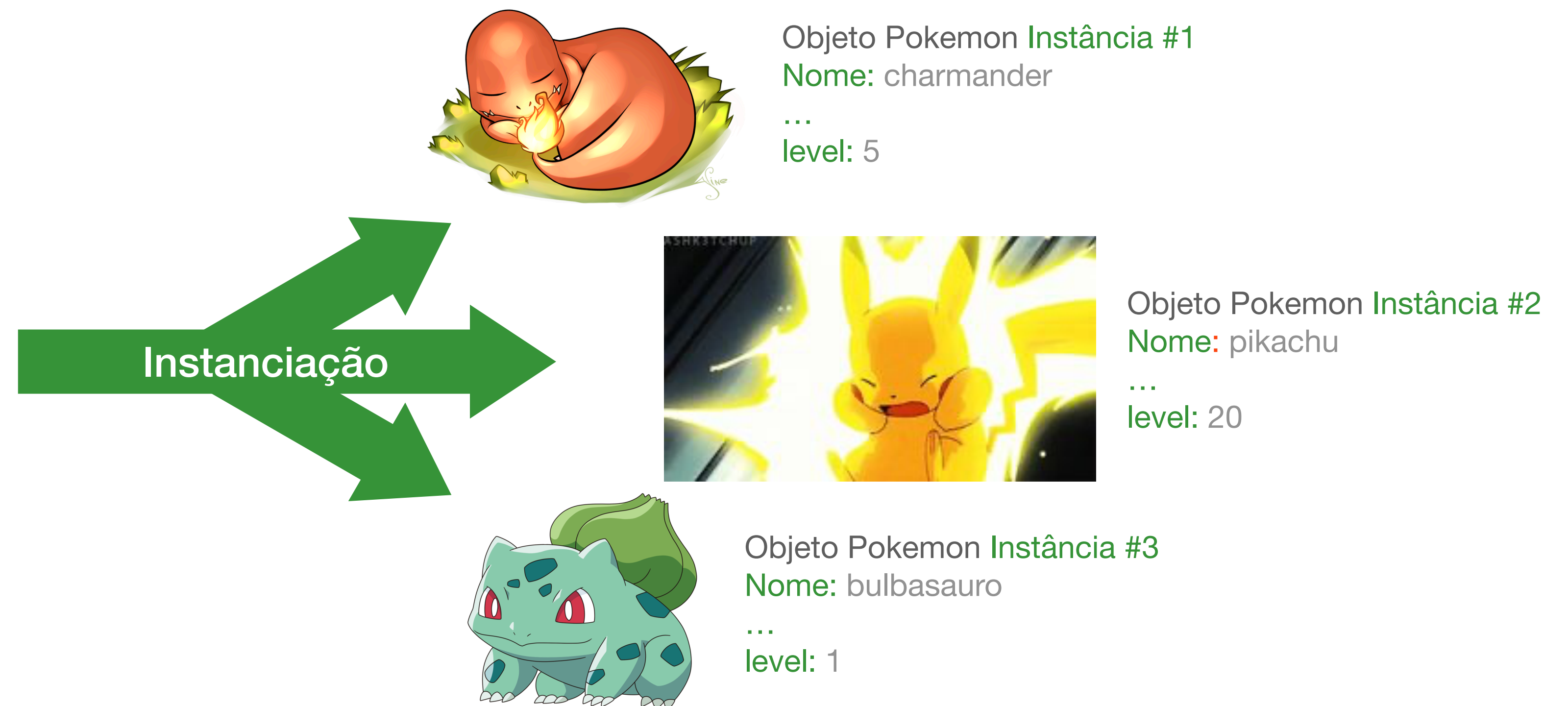


O paradigma da orientação a objetos

Em tempo de compilação

Classe Pokemon
<i>Atributo: Tipo</i> nome: String altura: int peso: float level: int
<i>Ação (argumentos): Tipo do Ret.</i> atacar(): void falar(): void dormir(tempo): void

Em tempo de execução



O paradigma da orientação a objetos

Classe x Objetos

- Os **objetos** representam as **coisas** do mundo real (domínio do problema)
- A descrição de um **grupo de objetos** é denominada **classe de objetos** ou simplesmente **classe**

O paradigma da orientação a objetos

Classe x Objetos

- Uma **classe** é uma **abstração** das **características relevantes** de um grupos de objetos (coisas) do mundo real
 - Na realidade em que vivemos, objetos são em geral bastante complexos

Classe Pokemon
nome: String
altura: int
peso: float
level: int
atacar(): void
falar(): void
dormir(tempo): void

O paradigma da orientação a objetos

Mensagens

- Para que um **objeto** realize alguma **tarefa**, deve haver um **estímulo** enviado a este objeto
- Lembre-se da analogia com as células do corpo humano
 - Respondem a estímulos
 - Estimulam outras células

O paradigma da orientação a objetos

Mensagens

- Independentemente da **origem do estímulo**, quando ele ocorre, diz se que o objeto em questão está recebendo uma **mensagem**.
- Uma **mensagem é uma requisição enviada de um objeto a outro** para que este último realize alguma **operação**



“In natural science, Nature has given us a world and we’re just to discover its laws. In computers, we can stuff laws into it and create a world.”

— Alan Kay

“My message to the serious programmer is this: spend a part of your working day examining and refining your own methods. Even though programmers are always struggling to meet some future or past deadline, methodological abstraction is a wise long term investment.”

— Robert W Floyd

Abstração

Abstração

abs·tra·ir

verbo transitivo

1.Separar mentalmente uma parte de um todo para a considerar independente das outras partes.

2.Separar, apartar.

verbo intransitivo

3.Fazer abstração.

4.[Por extensão] Omitir.

verbo pronominal

5. Distrair-se.

6. Focar a sua atenção em. = CONCENTRAR-SE

- "abstrair", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2021, <https://dicionario.priberam.org/abstrair>

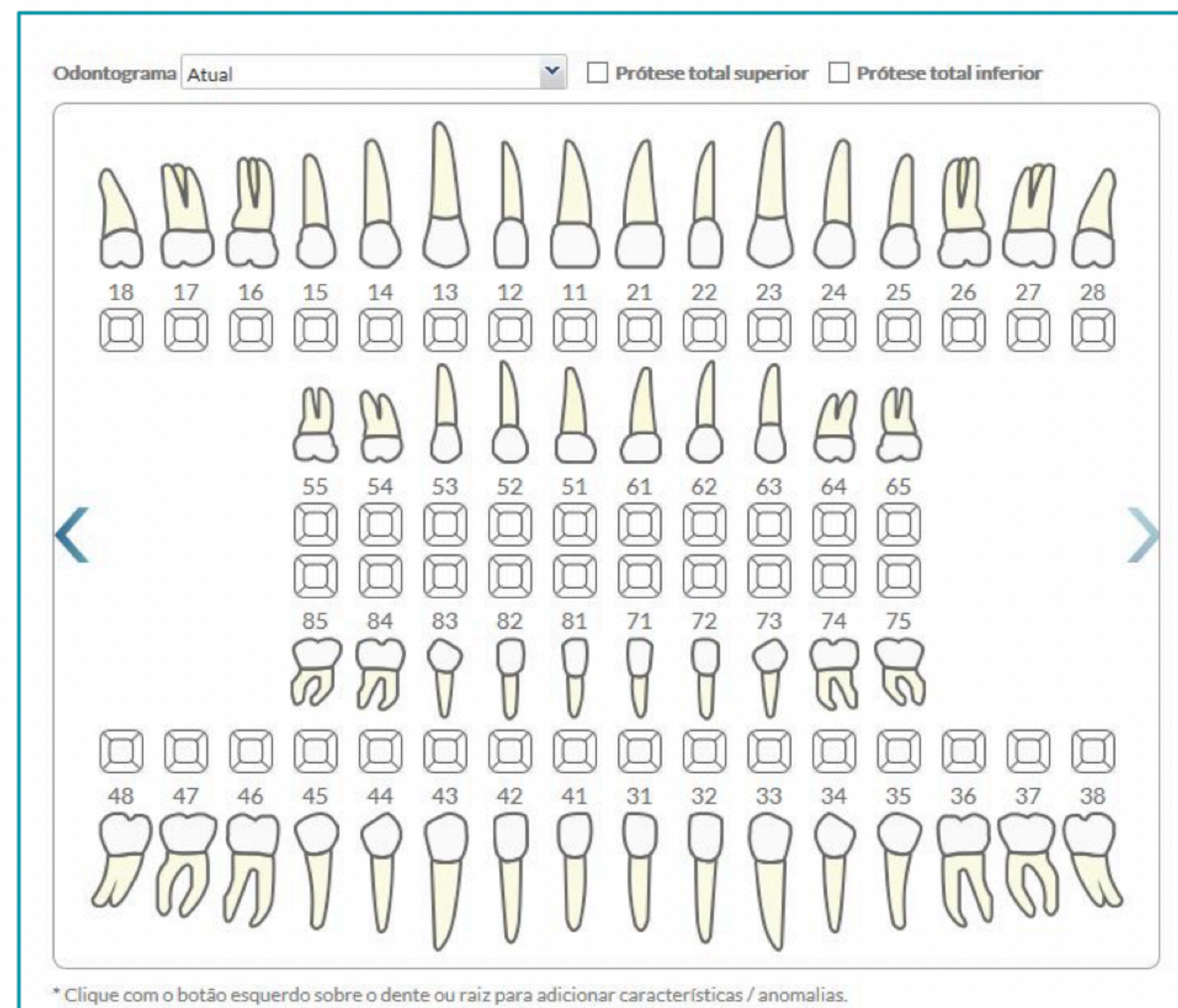
Abstração

- Uma **abstração** é qualquer **modelo** que inclui os **aspectos relevantes** de **alguma coisa**, ao mesmo tempo em que **ignora os menos importantes**.



Abstração

Capacidade de abstração x Experiência



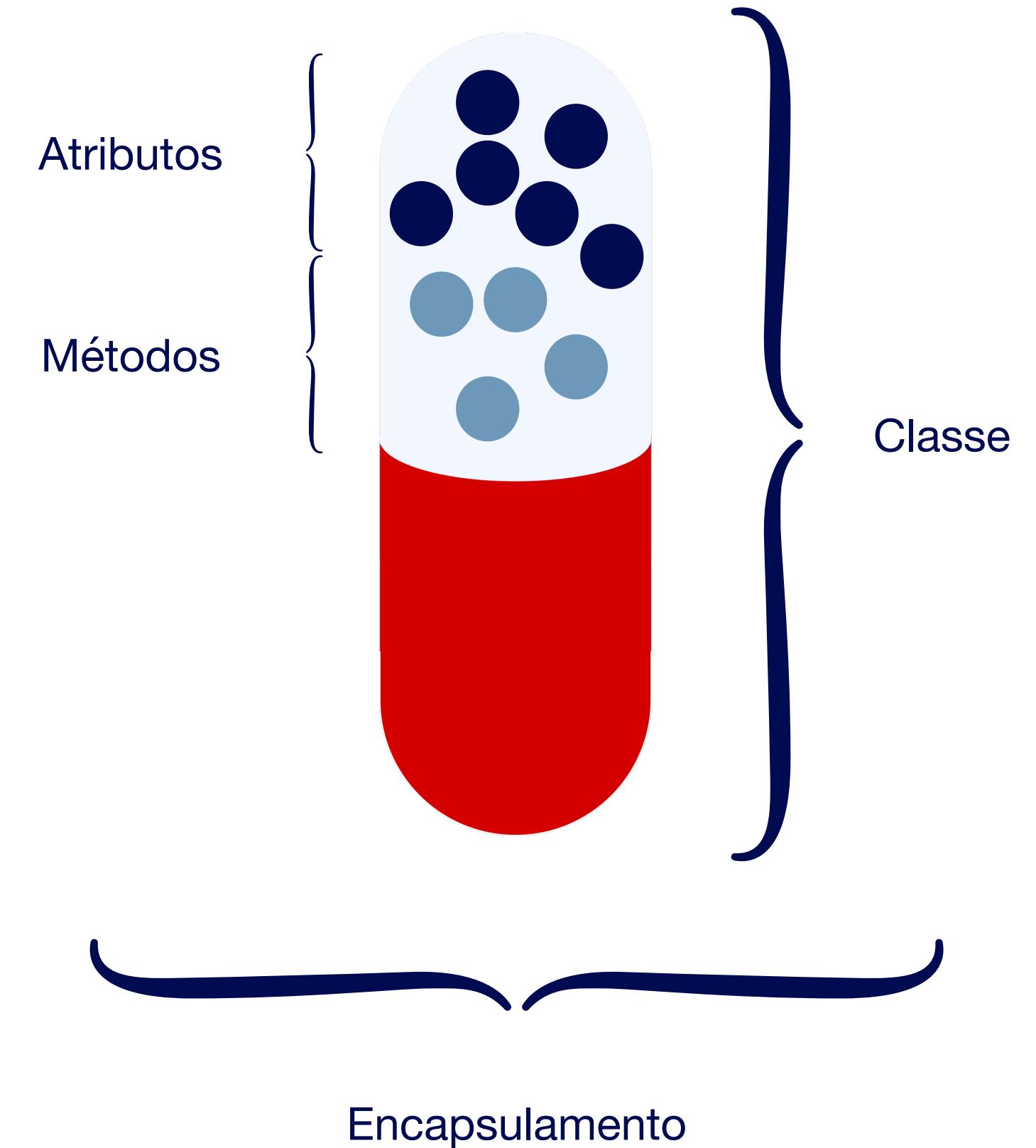


Encapsulamento

Encapsulamento

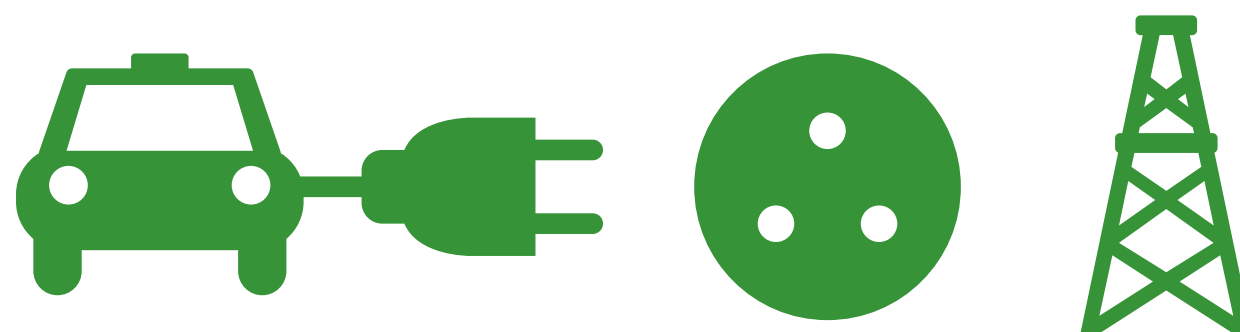
Objeto

- Nossa **unidade modular** é composto:
 - Estado
 - Valores das variáveis internas
 - Comportamento
 - Ações que são realizados pelo objeto
 - Modo como elas são realizadas



Encapsulamento

- É uma forma de restringir o acesso ao estado interno do objeto
- Permite a colaboração apenas via mensagens (requisitante/requisitado)
- A maneira (implementação) que o objeto requisitado realiza a tarefa não é exposta/conhecida pelo objeto requisitante

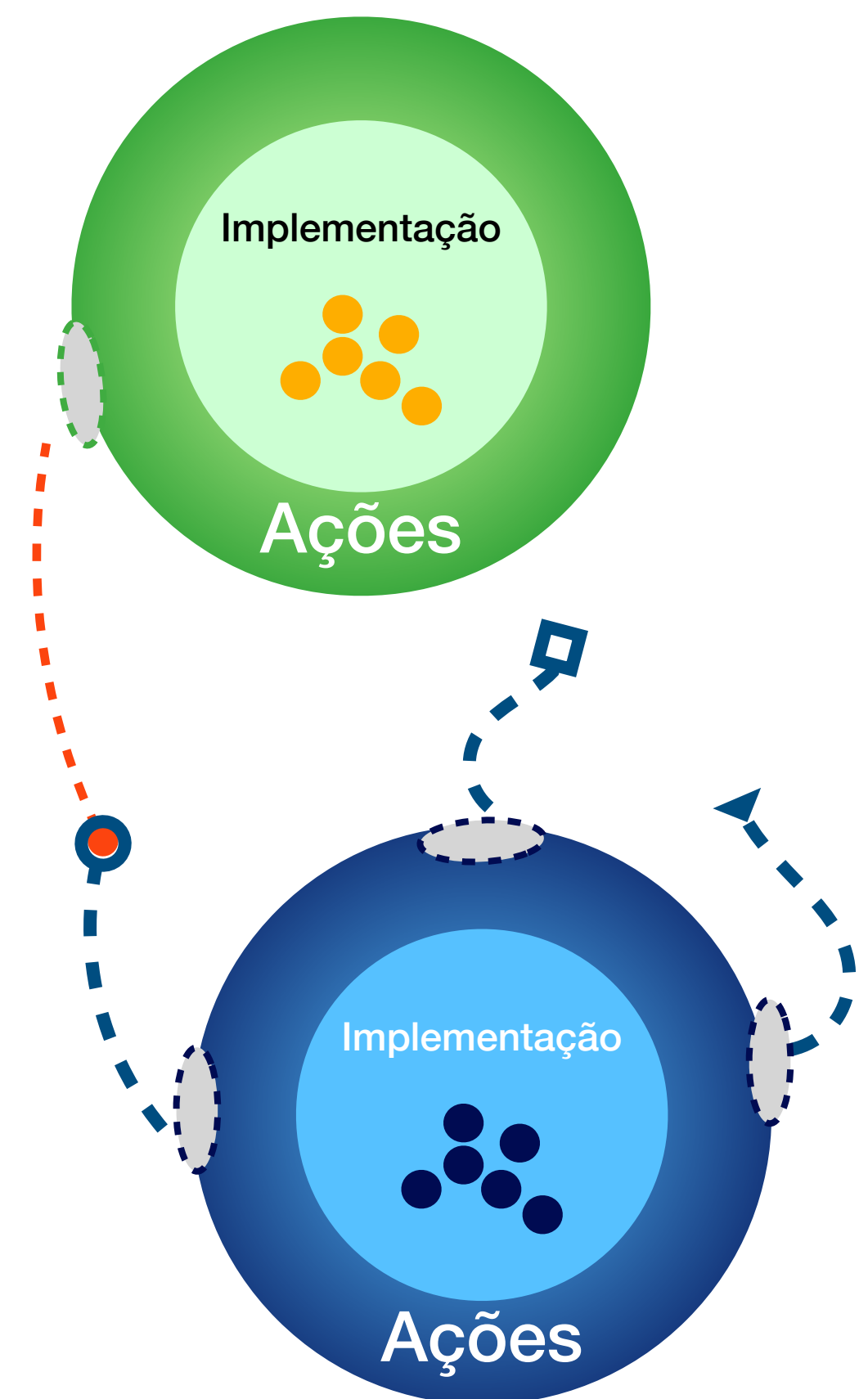


Encapsulamento

Interface do objeto

- Informa o que um objeto **conhece** e o que ele é capaz de **fazer**
- Define os **serviços/ações** que um objeto pode realizar
 - Logo, as mensagens que ele recebe

Data Hiding





Herança

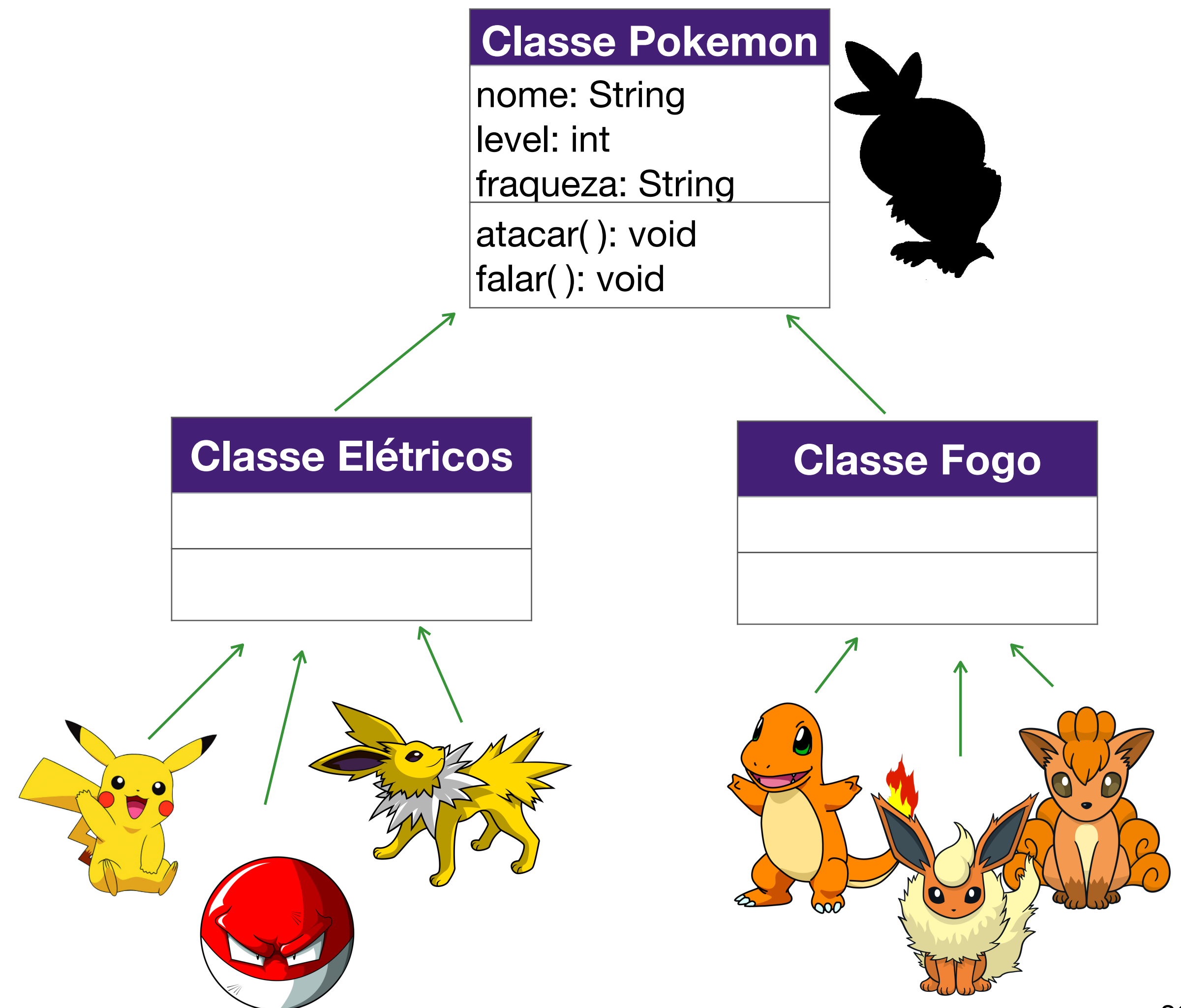
Herança

Mais um camada de abstração

- Por meio da herança, **classes semelhantes são agrupadas em hierarquias**
- Permitem que as **classes herdem atributos e comportamentos** de outras classes de **níveis acima**
- Cada nível da hierarquia pode ser visto com um nível de abstração

Herança

- Facilita o **compartilhamento de comportamentos**
- As diferenças ou variações em uma classe particular podem ser organizadas de forma mais clara





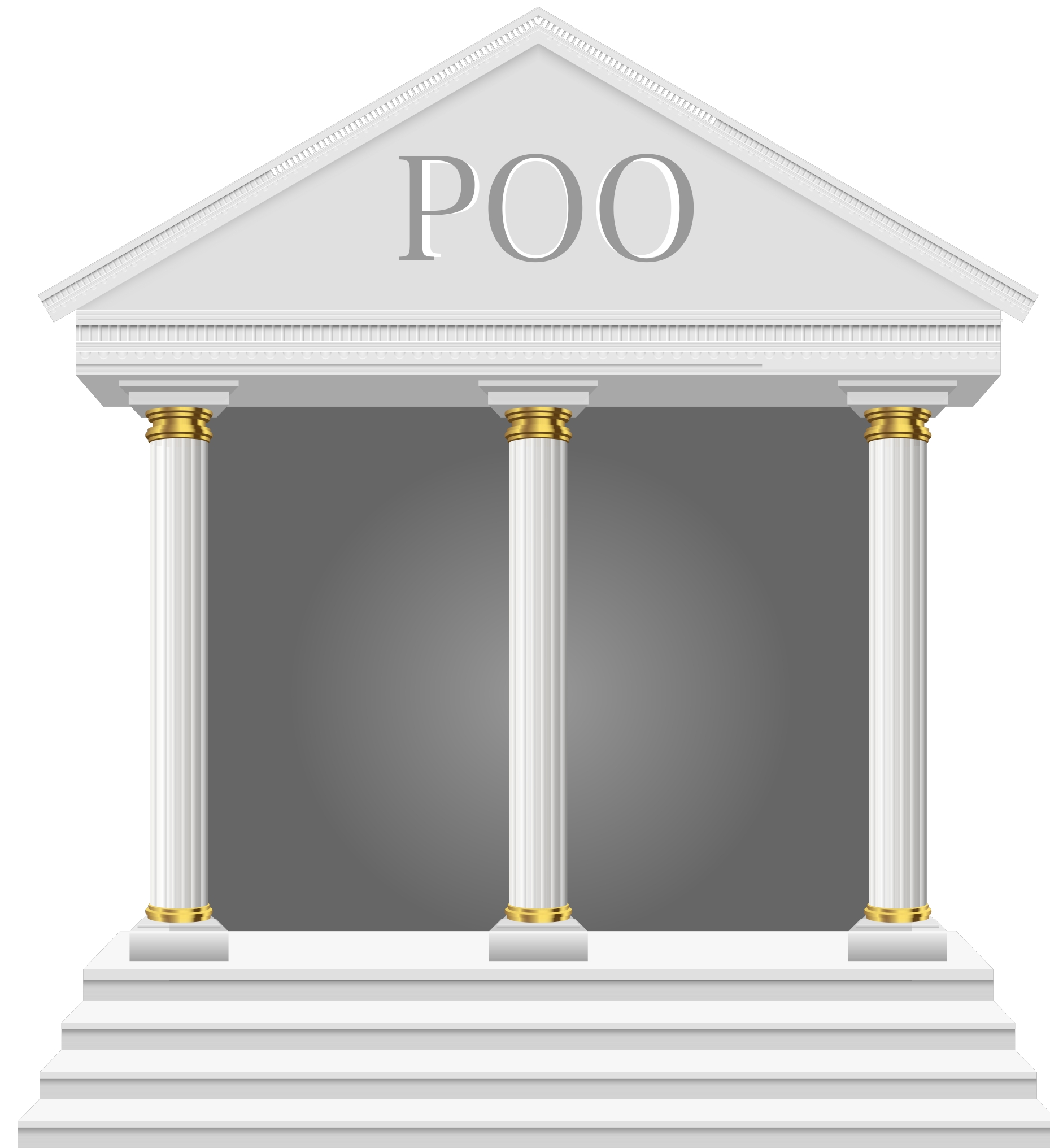
Polimorfismo

Polimorfismo

- É um termo originário do grego e significa “**muitas formas**” (poli = muitas e morphos = formas)
- É a capacidade de um método poder ser implementado de **diferentes formas** ou de realizar “**coisas**” **diferentes**
- A decisão sobre qual o método será executado é tomada em tempo de execução, através do mecanismo de ligação tardia (late binding)

Polimorfismo





Por hoje é só