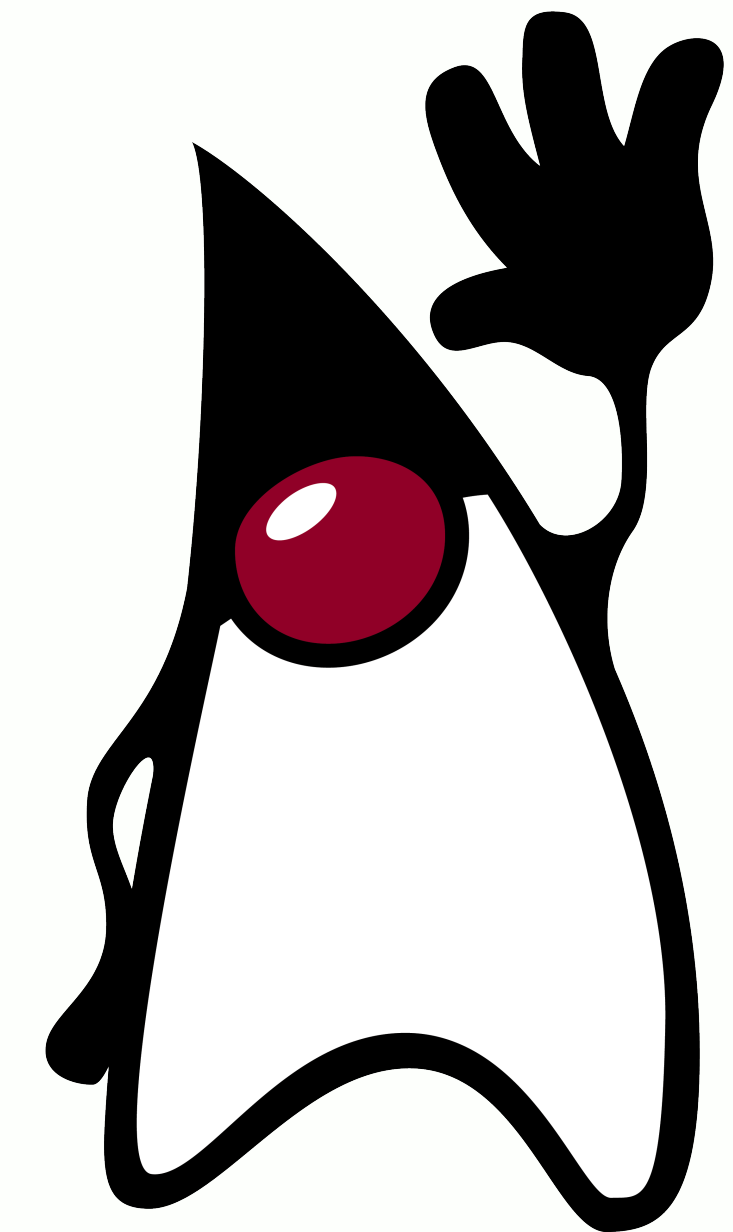




UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ

# Classes abstratas

QXD0007 - Programação Orientada a Objetos

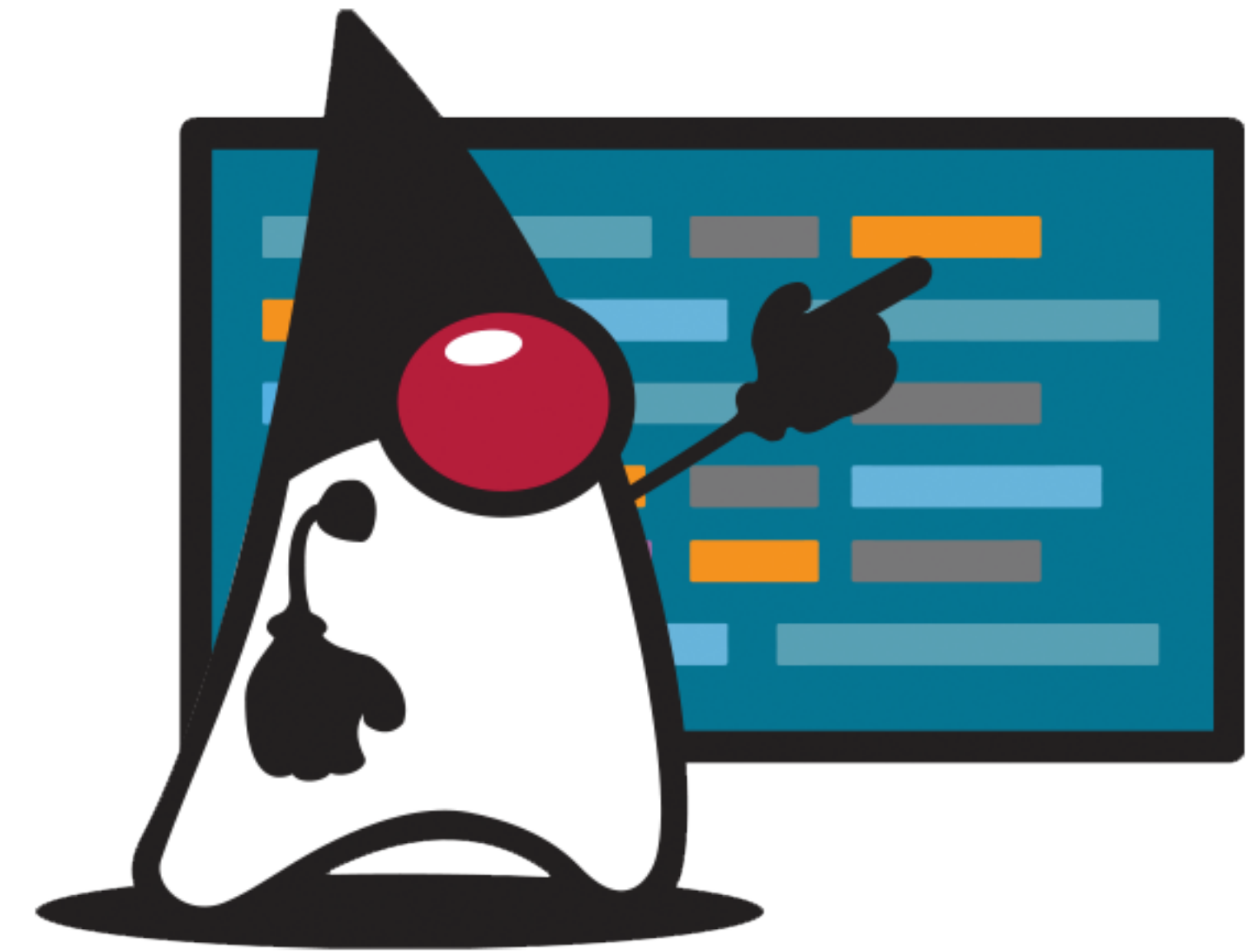


Prof. Bruno Góis Mateus ([brunomateus@ufc.br](mailto:brunomateus@ufc.br))

# Conteúdo

- Introdução
- Classe abstrata
- Capturando um Pokemon

# Introdução



# Introdução

- Desde os primórdios do desenvolvimento de software, o conceito de **reuso** foi reinventado várias vezes
- O paradigma OO nos oferece diversos mecanismo que facilitam o reuso
- Contudo, a utilidade e a reutilização do código dependem de quão bem ele foi projetado e implementado
- Uma parte software particularmente importante no contexto de reuso é o que nós chamamos de **contrato**

O termo **contrato** é amplamente usado em muitos aspectos dos negócios, incluindo desenvolvimento de software. Não confunda o conceito apresentado aqui com outros possíveis conceitos de **design de software**.

# Introdução

- Contrato
  - Um mecanismo que exige que o desenvolvedor cumpra especificações
  - Isso inclui seguir nomes de métodos, número de parâmetros e outra coisa mais
  - Padrões criados para facilitar as boas práticas de desenvolvimento

# Introdução

- Forçar o cumprimento do contrato é vital
  - Um desenvolvedor é sempre capaz de quebrar o contrato
  - Um desenvolvedor iniciante pode resolver reinventar a roda ao invés de utilizar o especificação
  - Um padrão deixa se ser benéfico caso ele não seja utilizado
-

**Em **Java** temos duas opções  
de definir esses contratos:  
**Classes Abstratas e Interfaces****



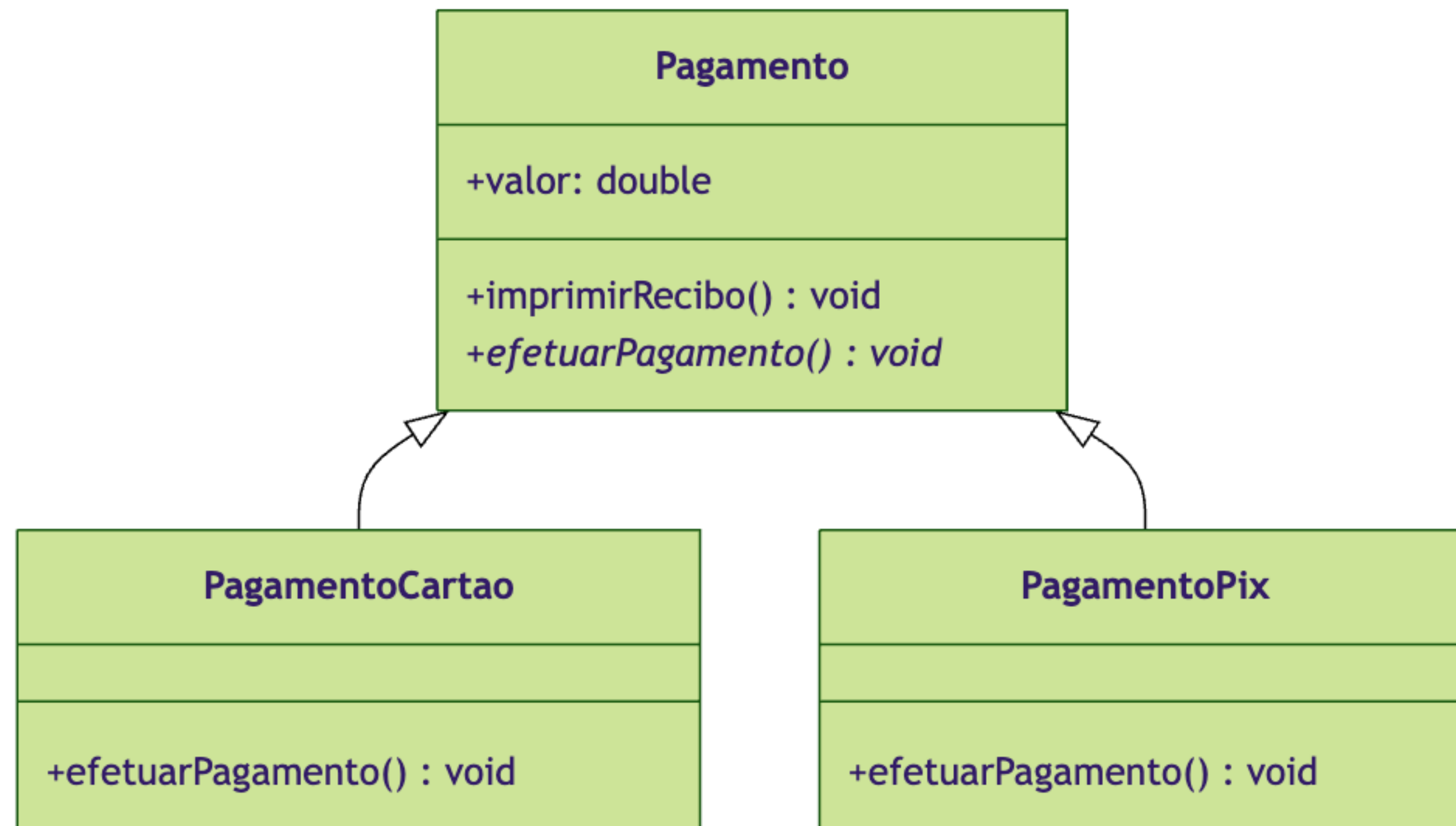
# Classes abstratas



# Classes abstratas

- Uma classe que pode conter um ou mais métodos sem implementação (**abstratos**) e métodos concretos (com implementação).
- Portanto, é uma classe que **não pode ser instanciada** diretamente
  - Mas podem ter construtores, para reuso
- Possibilitam herança de código **preservando comportamento (semântica)**
  - Métodos **abstratos** são **implementados nas subclasses**
- **Pode possuir métodos com implementação (concretos)**

# Classes abstratas

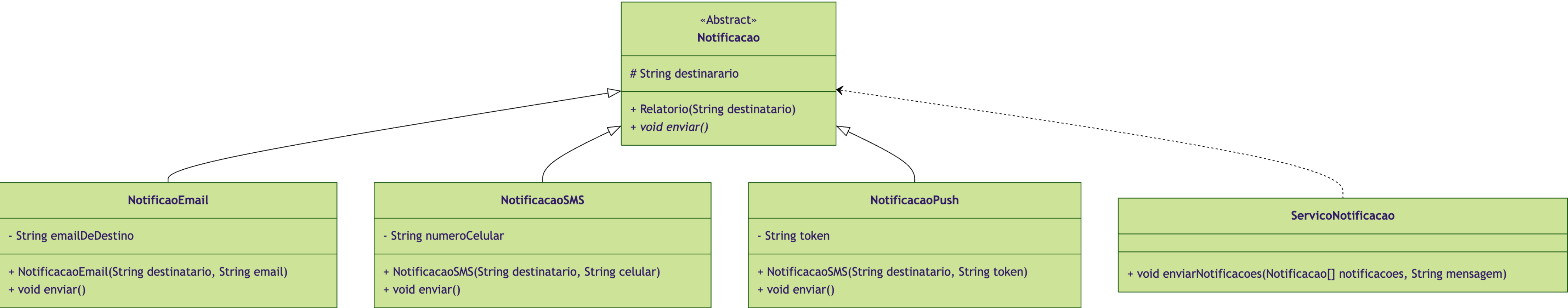


# Classes abstratas

## Quando utilizar?

- Quando **há comportamento base comum** entre subclasses
- Quando você precisa **forçar a implementação de métodos específicos**
- Quando deseja **evitar duplicação de código**

# Classes abstratas



# Capturando um pokemon



# Capturando um pokemon



## Pokédex data

National №	147
Type	DRAGON
Species	Dragon Pokémon
Height	1.8 m (5'11")
Weight	3.3 kg (7.3 lbs)

## Training

EV yield	1 Attack
Catch rate	45 (5.9% with PokéBall, full HP)
Base Friendship	35 (lower than normal)
Base Exp.	60
Growth Rate	Slow

## Base stats

HP	41		192	286
Attack	64		119	249
Defense	45		85	207
Sp. Atk	50		94	218
Sp. Def	50		94	218
Speed	50		94	218
Total	300		Min	Max

[1, 255]

$$X = \max\left(\frac{(3 * HP_{max} - 2 * HP_{atual}) * TAXA_{modificada}}{3 * HP_{max}}, 1\right) + BONUS_{estado}$$

# Capturando um pokemon



## Training

EV yield	1 Attack
Catch rate	45 (5.9% with PokéBall, full HP)

## Base stats

HP	41	<div></div>
Attack	64	<div></div>
Defense	45	<div></div>
Sp. Atk	50	<div></div>
Sp. Def	50	<div></div>
Speed	50	<div></div>
Total	300	

[1, 255]

$$X = \frac{3 * HP_{max} - 2 * HP_{atual}}{3 * HP_{max}} * catch_{rate} * bonus_{ball} * bonus_{status}$$

$$X = \frac{3 * HP_{max} - 2 * HP_{max}}{3 * HP_{max}} * catch_{rate} * bonus_{ball} * bonus_{status}$$

$$X = \frac{HP_{max}}{3 * HP_{max}} * catch_{rate} * bonus_{ball} * bonus_{status}$$

$$X = \frac{catch_{rate} * bonus_{ball} * bonus_{status}}{3}$$

$$X = 15$$



# Capturando um pokemon



## Training

EV yield	1 Attack
Catch rate	45 (5.9% with PokéBall, full HP)
Base Friendship	35 (lower than normal)
Base Exp.	60
Growth Rate	Slow

$$TAXA_{modificada} = f(x)$$



$$f(x) = x$$



$$f(x) = 1.5 * x$$



$$f(x) = 2 * x$$



$$f(x) = \begin{cases} 4 * x \\ x \end{cases}$$

Se speed do Pokemon > 100  
caso contrário



$$f(x) = \begin{cases} 8 * x \\ x \end{cases}$$

Se o Pokemon é do sexo oposto e da mesma espécie  
caso contrário



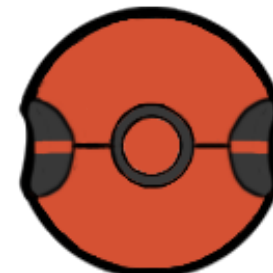
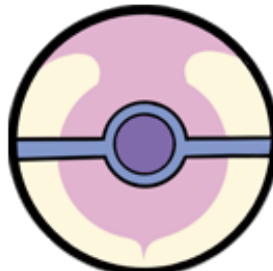
$$f(x) = \begin{cases} -20 \\ x \\ +20 \\ +30 \end{cases}$$

Se peso ≤ 99  
Se peso ≥ 100 e ≤ 199.9  
Se peso ≥ 200 e ≤ 299.9  
Se peso ≥ 300



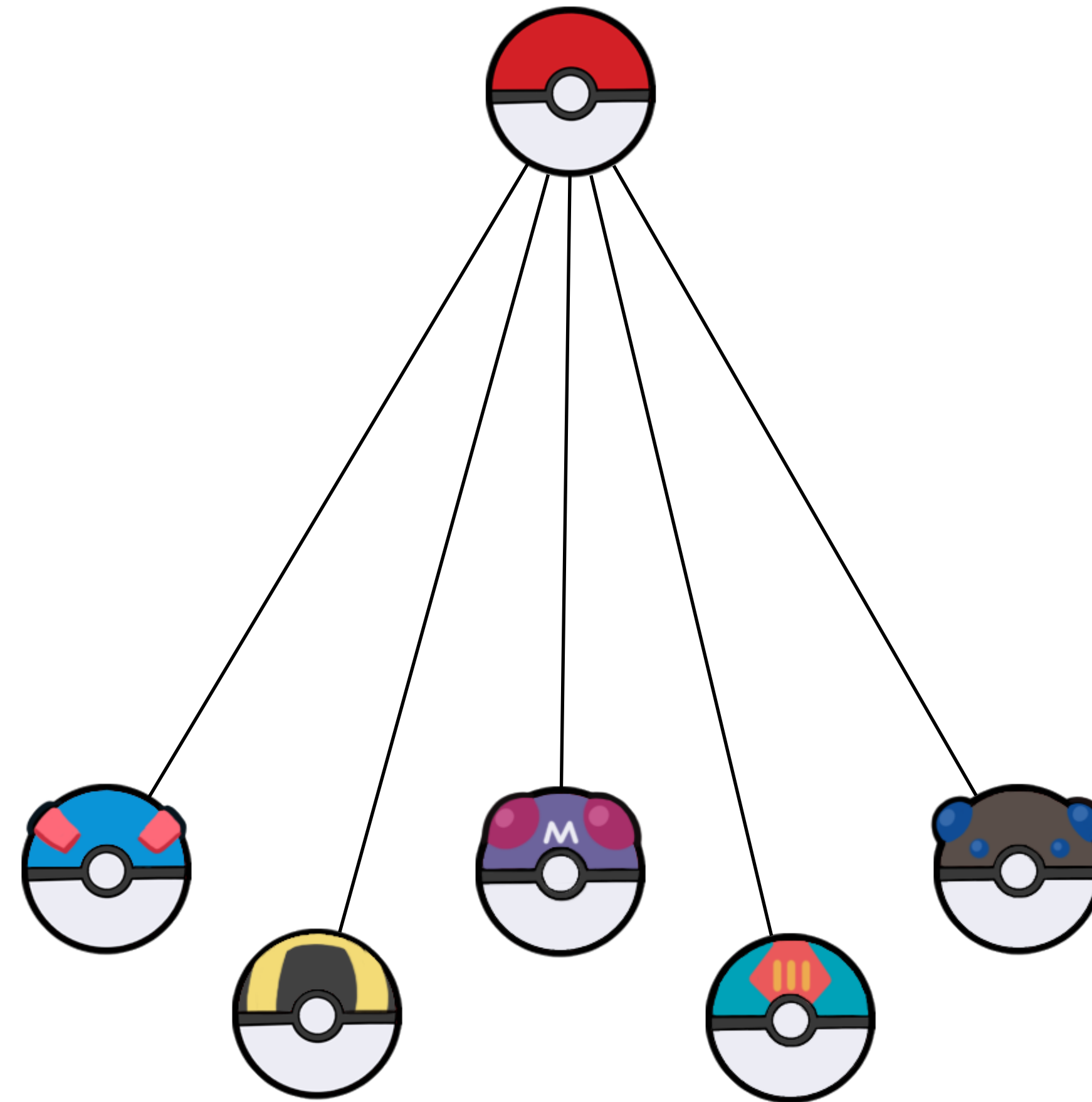
$$f(x) = \begin{cases} 5 * x \\ x \end{cases}$$

Se o pokemon foi pescado  
caso contrário



# Capturando um pokemon

Reuso sem subtipos



# Capturando um pokemon



## Training

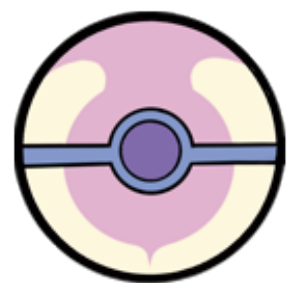
EV yield	1 Attack
Catch rate	45 (5.9% with PokéBall, full HP)

## Base stats

HP	41	<div></div>
Attack	64	<div></div>
Defense	45	<div></div>
Sp. Atk	50	<div></div>
Sp. Def	50	<div></div>
Speed	50	<div></div>
Total	300	



Altera os pontos de amizade do pokemon capturado para 200

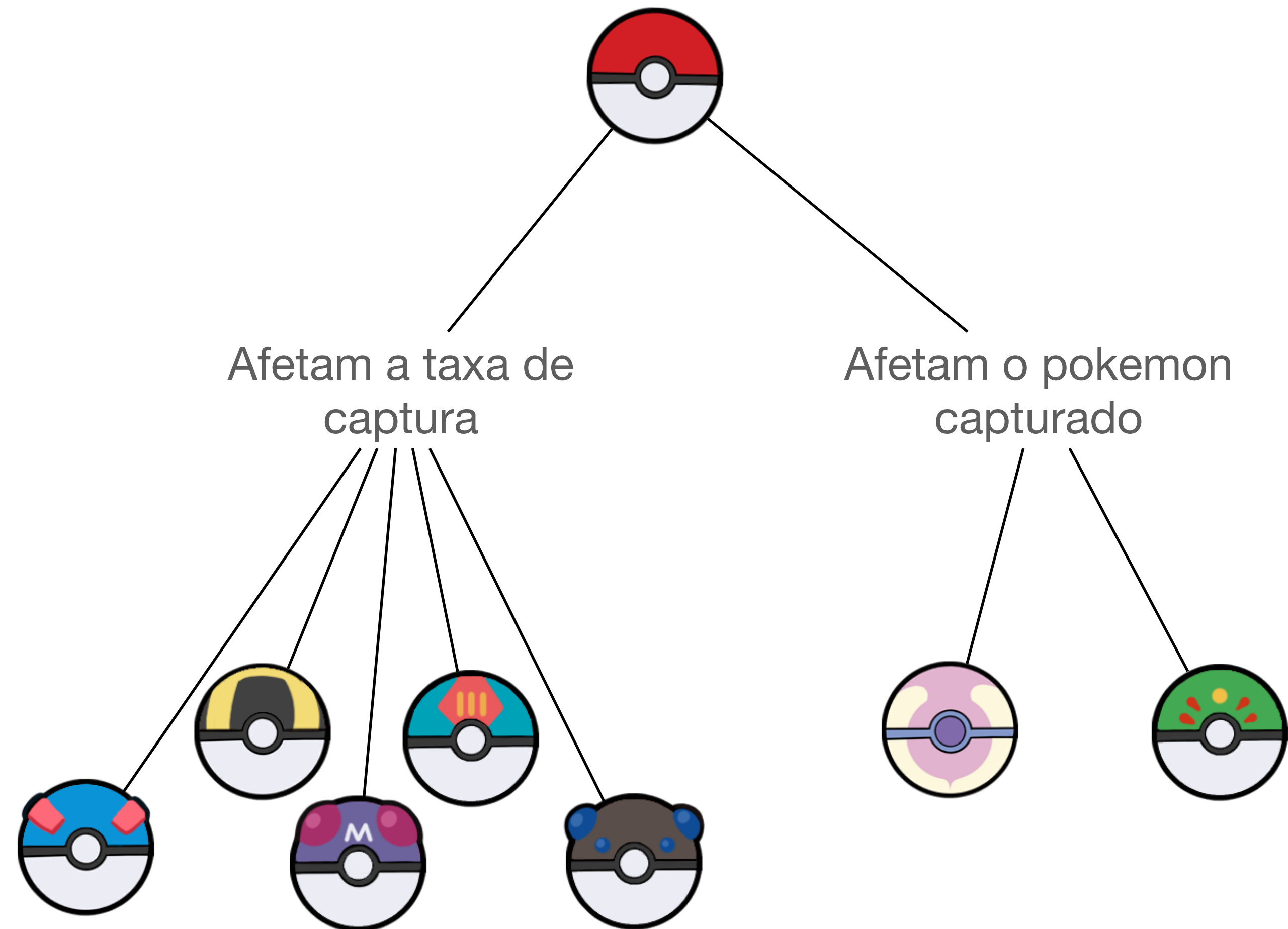


Restaura o HP, o PP e qualquer condição de status do pokemon capturado



# Capturando um pokemon

## Reuso com subtipos



# Por hoje é só

