

# Programação Orientada a Objetos em R: classes S3

Nelson S. dos Santos

**Núcleo de Ciência de Dados e Computacional em Economia e Finanças**

Faculdade de Ciências Econômicas

Universidade Federal do Rio Grande do Sul

1 de novembro de 2024

# Sumário

- 1 Introdução
- 2 Modelo de Classes S3
- 3 Considerações finais
- 4 Referências

# Introdução

# Problema

Como escrever código orientado a objeto em GNU R?

## Contexto do problema

- O dialeto GNU R da linguagem S suporta prioritariamente o paradigma de programação funcional, sendo fortemente inspirado na sua concepção pelas linguagens [Scheme](#) e como relata [lhaka \(1998\)](#).
- Por isso, o mecanismo de classes tradicionalmente usado em linguagens de programação que suportam orientação a objeto parece antinatural.

# Solução

- 1 Observemos que o GNU R tal como as demais linguagens funcionais derivadas da linguagem Lisp apresenta a capacidade de executar funções genéricas, isto é, funções que, embora tenham o mesmo nome, apresentam implementações diferentes a depender do tipo do objeto.
- 2 Então, vamos usar funções genéricas para produzir o polimorfismo que caracteriza a orientação a objetos.
- 3 Chama-se este modelo de orientação a objetos de orientação a objeto por mecanismo de funções genéricas.

## Observações

- A primeira implementação GNU R foi baseada na versão 3 da linguagem S que foi a primeira versão que suportou orientação a objetos, já usando o mecanismo de funções genéricas no seu pacote Base. Por isso, o primeiro modelo de classes tanto de S quanto de R é denominado modelo de classes S3.
- Atualmente, tanto a linguagem S atual quanto o GNU R contam também com o modelo S4, que é uma evolução do S3, mas não totalmente compatível com este.
- Além do S4, as novas versões do GNU R contam ainda com o modelo de classes de referência (RC) e, por meio de pacotes adicionais ao pacote Base, o modelo R6.
- As referências básicas desta apresentação são [[Matlof \(2011\)](#)], [[Wickham \(2019\)](#)] e [[Rodrigues \(2020\)](#)].

## Modelo de Classes S3



## Classes S3: definição

Uma classe S3 é uma lista ou vetor a que se adiciona o atributo *class* para definir a classe do objeto R.

### Observação

Note que, ao adicionar o atributo *class* ao objeto base do GNU R, seus dados se tornam automaticamente atributos da classe S3 criada.

## Classes S3: exemplo de definição de atributos

Criando um vetor de inteiros e verificando sua classe.

```
> x <- c(1,2,3)
> is.object(x)
[1] FALSE
> class(x)
[1] "numeric"
```

## Classes S3: exemplo de definição de atributos (cont.)

Transformando o objeto base GNU R em um objeto S3

```
> class(x) <- 'carro'
> class(x)
[1] "carro"
> is.object(x)
[1] TRUE
> x
[1] 1 2 3
attr(,"class")
[1] "carro"
```

## Classes S3: melhor prática para definição de atributos

Use a função `structure()` na criação do objeto como demonstrado no exemplo abaixo.

### Exemplo de criação de objeto da classe carro

```
x <- structure(  
  list(modelo = "celta",  
        marca = 'chevrolet',  
        motor = '1.0'),  
  class = "carro"  
)
```

## Classes S3: construtores de classe

Para garantir que todos os objetos de uma classe tenham sempre os mesmos atributos, usa-se uma função que fará o papel de método construtor da classe, validando o tipo de dados de cada atributo definido.

Não abordaremos aqui como realizar este procedimento, mas mostraremos a seguir a lógica subjacente à sintaxe básica da criação de métodos para as classes.

## Classes S3: definição de métodos

Os métodos das classes S3 são criados por meio de funções genéricas, isto é, funções que determinam operações diferentes para cada tipo de objeto.

Para isso, as funções genéricas chamam uma função denominada `UseMethod()`. Por exemplo, se `comprar()` for uma função genérica, ao se chamar esta função sobre um objeto da classe `carro`, ela chamará a função `UseMethod(comprar.carro())`.

Mas, se `comprar()` for chamada sobre um objeto de classe `roupa`, será chamada `UseMethod(comprar.roupa())`.

## Considerações finais

## Considerações finais

A linguagem S e seu dialeto GNU R são essencialmente funcionais e, por isso, a criação de classes e seus objetos utiliza funções genéricas para conseguir o polimorfismo típico de programas orientados a objeto. .

Apresentou-se uma introdução à programação orientada a objeto com classes S3 em R.




O desenvolvimento posterior é agrupar métodos e atributos por meio de métodos construtores e funções ajudantes ao modo costumeiro em JavaScript quando não se usa a sintaxe de classes disponível nas versões superiores a ES6.

Mais isso fica para uma próxima oportunidade...



## Referências

## Referências

-  RODRIGUES, Bruno. *Modern R with the tidyverse*, Chapter 8. LeanPub: 2020. Disponível em [link](#).
-  WICKHAM, Hadley. *Advanced R, 2nd. Edition*. Boca Raton, FL: CRC Press, 2019. Disponível em [link](#).
-  MATLOF, N. *The Art of R Programming*. San Francisco, CA: No Starch, 2011.