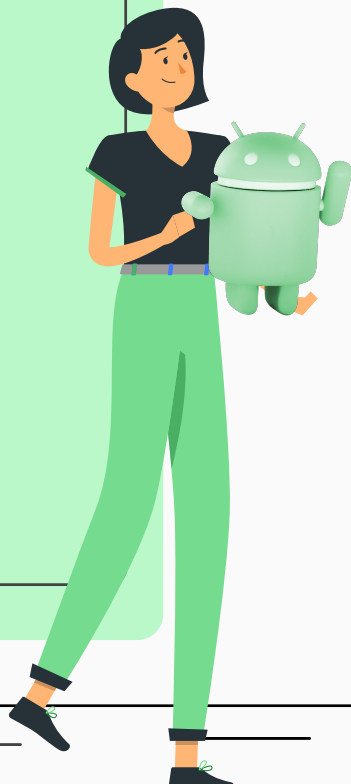


# AULA 9

SOLID, Clean Architecture





## 01 POO

Fundamentos POO

## 02 ARQUITETURA

Praticar o que aprendemos  
de kotlin

## 03 SOLID

Boas práticas de  
programação

## SUPER TRUNFO

Super trunfo é um jogo de cartas. Veja o código do aplicativo Super trunfo para entender como o jogo funciona.

<https://github.com/marcellalcs/supertrunfo>



## CARD CLASS

```
val cardOneMaxVelocity : Int =
    when (currentVehiclePlayerOne["type"]) {
        "car" ->
            if (currentVehiclePlayerOne["style"] == "sedã") {
                currentVehiclePlayerOne["maxAcceleration"]?.toInt() ?: 0
            } else {
                currentVehiclePlayerOne["maxAcceleration"]?.toInt() ?: 0 + 10
            }

        "motorcycle" -> (1 / (currentVehiclePlayerOne["weight"]?.toInt()
            ?: 1)) * (currentVehiclePlayerOne["maxAcceleration"]?.toInt() ?: 0)

        else -> (currentVehiclePlayerOne["maxAcceleration"]?.toInt()
            ?: 1 * (currentDriverPlayerOne["boldness"]?.toInt() ?: 1))
    }

val accelerationTimeCardOne : Int = (currentVehiclePlayerOne["accelerationTime"]?.toInt()
    ?: 1) * (1 / (currentDriverPlayerOne["accelerationTime"]?.toInt() ?: 1))

val passengersCardOne : Int = (currentVehiclePlayerOne["passengers"]?.toInt()
    ?: 0) * (1 + (currentDriverPlayerOne["defensiveDriving"]?.toInt() ?: 0))

val xPCardOne : Int = when (currentVehiclePlayerOne["type"]) {
    "car" -> currentDriverPlayerOne["carXP"]?.toInt() ?: 0
    "motorcycle" -> currentDriverPlayerOne["motorcycleXP"]?.toInt() ?: 0
    else -> currentDriverPlayerOne["bikeXP"]?.toInt() ?: 0
}
```

## Como uma carta é construída?

A partir do veículo e do condutor do jogador

## Quais propriedades da carta importam para definir um ganhador?

- maxVelocity
- accelerationTime
- Passengers
- xP

Como modelar card?

## MODELAR OUTRAS CLASSES

