

**Disciplina: POO**

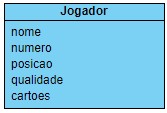
**Professor: Genivan Silva**

**Discente: Marcos Antonio**

# Avaliação

**OBS: É obrigatório o uso das técnicas de encapsulamento para os atributos e também o tratamento de exceções quando houver entrada de dados do usuário.**

1. Crie uma pasta chamada Teste para guardar todos os elementos usados nesta avaliação.
2. Dentro dessa pasta crie um pacote chamado Dominio e um módulo chamado Time. No módulo Time crie uma classe chamada Jogador com os atributos listados na figura abaixo.



1. Crie os métodos abaixo na classe Jogador. Considere que a pontuação mínima de qualidade do jogador é zero e a máxima é 100.

1

* 1. Crie o método **aplicar\_cartoes**. O método deve receber como parâmetros o jogador e a quantidade de cartões recebidos. O método deve modificar o atributo cartões do jogador indicado, aumentando a quantidade de cartões registrados.
  2. Crie o método **sofrer\_lesao** que gera uma lesão no jogador. O método deve gerar um número aleatório entre 0,05 e 0,4 (ou 5% a 40%). Esse valor deve diminuir a pontuação de qualidade do jogador. O jogador nunca deve ficar com pontuação negativa. Caso a qualidade do jogador diminua até um valor menor que zero, a qualidade do jogador deve ser zero.

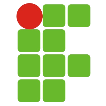
EXEMPLO:

Qualidade do jogador: 70

Método sofrer\_lesao(): número gerado = 0,2 (vinte porcento)

Qualidade do jogador após sofrer lesão: 56

* 1. Crie o método **treinamento**. Do mesmo modo que o método anterior, o treinamento deve aumentar aleatoriamente entre 5 e 40 porcento a pontuação de qualidade do jogador.
  2. Crie o método status\_jogador. O método apenas exibe o nome do jogador, a pontuação de qualidade e a quantidade de cartões que ele já recebeu.



1. Fora do pacote Dominio crie um arquivo python chamado **principal**. Nesse arquivo instancie pelo menos 5 jogadores, execute os métodos sofrer\_lesao, aplicar\_cartão e treinamento em dois deles. Exiba o status desses dois jogadores antes e após a execução dos métodos.

2

import random  
  
separador = 30\*'#'  
class Jogador:  
 def \_\_init\_\_(self, nome, numero, posicao, qualidade=0, cartoes=0):  
 self.\_nome = nome  
 self.\_numero = numero  
 self.\_posicao = posicao  
 self.\_qualidade = qualidade  
 self.\_cartoes = cartoes  
  
 @property  
 def nome(self):  
 return self.\_nome  
  
 @nome.setter  
 def nome(self, nome):  
 self.\_nome = nome  
  
 @property  
 def numero(self):  
 return self.\_numero  
  
 @numero.setter  
 def numero(self, numero):  
 self.\_numero = numero  
  
 @property  
 def posicao(self):  
 return self.\_nome  
  
 @posicao.setter  
 def posicao(self, posicao):  
 self.\_posicao = posicao  
  
 @property  
 def qualidade(self):  
 return self.\_qualidade  
  
 @qualidade.setter  
 def qualidade(self, qualidade):  
 try:  
 if qualidade < 100:  
 if(qualidade > 0):  
 self.\_qualidade = qualidade  
 else:  
 self.\_qualidade = 0  
 else:  
 self.\_qualidade = 100  
 except (ValueError, BaseException):  
 print('APENAS NUMEROS SÃO VALIDOS')  
  
 @property  
 def cartoes(self):  
 return self.\_cartoes  
  
 @cartoes.setter  
 def cartoes(self, cartoes):  
 try:  
 self.\_cartoes = cartoes  
 except (ValueError, BaseException):  
 print('APENAS NUMEROS SÃO VALIDOS')  
  
 def sofre\_lesao(self):  
 lesionado = self.qualidade \* (1 - (random.randint(5, 40) / 100))  
 self.qualidade = lesionado  
  
 def treinamento(self):  
 lesionado = self.qualidade \* (1 + (random.randint(5, 40) / 100))  
 self.qualidade = lesionado  
  
 def status\_jogador(self):  
 print(f'NOME: {self.nome}\nQUALIDADE: {self.qualidade}\nQUANTIDADE DE CARTOES: {self.cartoes}\n{separador}')  
  
  
def aplicar\_cartoes(jogador, qtd):  
 cartoes = jogador.cartoes + qtd  
 jogador.cartoes = cartoes

############################principal.py##################################

from teste.Dominio.time import \*  
  
j1 = Jogador('Messi', 10, 'Segundo Atacante', 94)  
j2 = Jogador('Cristiano Ronaldo', 7, 'Ponta Esquerda', 92)  
j3 = Jogador('Lukako', 9, 'Centroavante', 88)  
j4 = Jogador('Bastoni', 95 , 'Zagueiro', 80)  
j5 = Jogador('pessina', 22, 'Meio-campo', 75)  
  
j1.status\_jogador()  
j1.sofre\_lesao()  
j1.status\_jogador()  
j1.treinamento()  
j1.status\_jogador()  
aplicar\_cartoes(j1, 2)  
j1.status\_jogador()  
  
j4.status\_jogador()  
j4.sofre\_lesao()  
j4.status\_jogador()  
j4.treinamento()  
j4.status\_jogador()  
aplicar\_cartoes(j4, 5)  
j4.status\_jogador()

