

Número USP	RunCodes (30%)	Legibilidade (30%)	Modularização (20%)	Lógica (20%)	Total (100%)	Observações
14558311	10,00	10,00	10,00	10,00	10,00	
14783601	10,00	10,00	10,00	10,00	10,00	
						Ao jogar a lógica dos danos para a classe Pokémon você faz ela ser responsável por todos os novos tipos que vierem a ser criados, e tira a praticidade que o polimorfismo traz, em que cada classe filha cuida do seu próprio dano. O cálculo de cada dano ficava nos filhos exatamente para exemplificar como o polimorfismo permite tirar a complexidade de uma classe só e facilita a extensão de código para novos tipos apenas criando novos filhos. Do jeito que fez, se eu quiser adicionar um novo tipo vou ter que alterar a classe Pokémon e colocar mais ifs, até ficar uma lista gigantesca, difícil de ler e modificar. Lembre-se disso e que polimorfismo está aí para ajudar :) Os objetos não são instanciados. No exemplo já existia o instanciamento pronto e exemplificado. Além disso, ao não usar uma função para criar um novo Pokémon, teve que repetir o código de verificar o nome e instanciar o Pokémon.
14562528	0,00	5,00	5,00	0,00	3,00	
12450102	0,00				0,00	
11930735	0,00				0,00	
10831740	0,00				0,00	
14675819	10,00	10,00	10,00	10,00	10,00	
10351727	0,00	10,00	10,00	0,00	6,00	Não lê os nomes e não faz a lógica de instanciação dos dados. Lógica de dano do tipo elétrico está errada.
12725411	10,00	10,00	10,00	10,00	10,00	
14746377	0,00				0,00	
14571981	10,00	10,00	10,00	10,00	10,00	
10748177	10,00	10,00	10,00	10,00	10,00	
14573207	0,00				0,00	
13661979	0,00				0,00	
13875369	0,00				0,00	

14777754	0,00	5,00	5,00	0,00	3,00	Removeu as outras classes. Ao jogar a lógica dos danos para a classe Pokémon você faz ela ser responsável por todos os novos tipos que vierem a ser criados, e tira a praticidade que o polimorfismo traz, em que cada classe filha cuida do seu próprio dano. O cálculo de cada dano ficava nos filhos exatamente para exemplificar como o polimorfismo permite tirar a complexidade de uma classe só e facilita a extensão de código para novos tipos apenas criando novos filhos. Do jeito que fez, se eu quiser adicionar um novo tipo vou ter que alterar a classe Pokémon e colocar mais ifs, até ficar uma lista gigantesca, difícil de ler e modificar. Lembre-se disso e que polimorfismo está aí para ajudar :)
12692645	10,00	10,00	10,00	10,00	10,00	
14577917	10,00				2,00	
13833960	6,00	8,00	8,00	5,00	7,00	Ao jogar a lógica dos danos para a classe Pokémon você faz ela ser responsável por todos os novos tipos que vierem a ser criados, e tira a praticidade que o polimorfismo traz, em que cada classe filha cuida do seu próprio dano. O cálculo de cada dano ficava nos filhos exatamente para exemplificar como o polimorfismo permite tirar a complexidade de uma classe só e facilita a extensão de código para novos tipos apenas criando novos filhos. Do jeito que fez, se eu quiser adicionar um novo tipo vou ter que alterar a classe Pokémon e colocar mais ifs, até ficar uma lista gigantesca, difícil de ler e modificar. Lembre-se disso e que polimorfismo está aí para ajudar :) Além disso, ao não usar uma função para criar um novo Pokémon, teve que repetir o código de verificar o nome e instanciar o Pokémon.
14654166	10,00	10,00	10,00	10,00	10,00	
14614626	0,00				0,00	
13827570	10,00	10,00	10,00	10,00	10,00	
14581075	0,00				0,00	
13732366	0,00				0,00	
14713471	0,00				0,00	
12690813	0,00				0,00	
14585955	0,00				0,00	
11299940	8,00	10,00	10,00	9,00	9,40	Erro na lógica de dano em elétrico
14785493	10,00	10,00	10,00	10,00	10,00	

10310641	0,00				0,00	
14589789	0,00				0,00	
14748104	0,00				0,00	
12610708	0,00				0,00	
15111714	10,00	10,00	10,00	10,00	10,00	
14572613	10,00	10,00	10,00	10,00	10,00	
13874841	0,00				0,00	
11912849	0,00				0,00	
13783183	0,00				0,00	
14801416	0,00				0,00	
4202166	10,00	10,00	10,00	10,00	10,00	Esqueceu o rstrip na leitura do nome
14670752	10,00	10,00	10,00	10,00	10,00	
13686793	0,00				0,00	
14778776	0,00				0,00	
11234756	0,00				0,00	
14606961	0,00				0,00	
12558512	0,00				0,00	
5340621	10,00	10,00	10,00	10,00	10,00	
10748708	0,00	10,00	10,00	6,00	7,20	Esqueceu de tirar o '\r' da entrada e a lógica de dano está errada.
14611644	10,00	9,00	9,00	10,00	9,40	Ao não usar uma função para criar um novo Pokémon, teve que repetir o código de verificar o nome e instanciar o Pokémon. O que é uma variável maravilha?
13725587	0,00				0,00	
14745682	10,00	10,00	10,00	8,00	9,60	Criou um objeto de Pokémon para cada classe para depois saber qual a classe certa.
14577917	10,00	10,00	10,00	10,00	10,00	Esqueceu o rstrip na leitura do nome
13688030	0,00				0,00	
13749844	0,00				0,00	
12541802	8,00	10,00	10,00	9,00	9,40	Esqueceu de tirar o '\r' da entrada e a lógica de dano de água em grama está errada.
9328096	0,00				0,00	
10689651	0,00				0,00	
12610497	0,00				0,00	