

## Lista de Exercícios SOLID

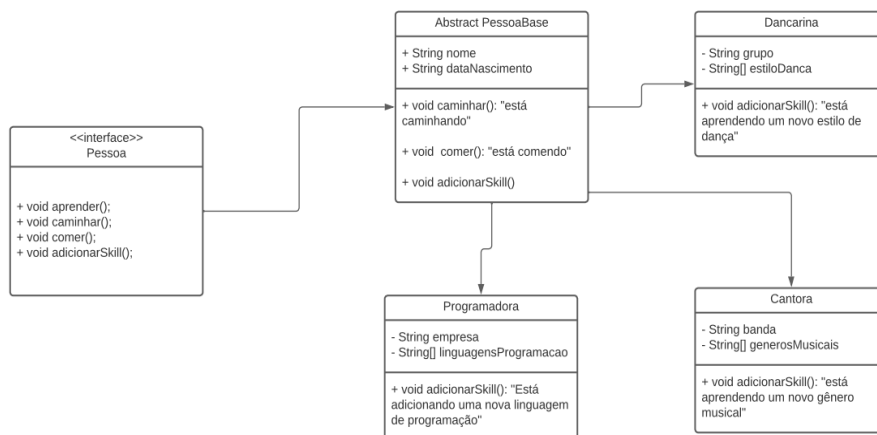
OS CÓDIGOS ESTÃO NESSE REPOSITÓRIO: <https://github.com/Helon21/AtividadeSolid>

Considere o diagrama abaixo:

Programadora	Dancarina	Cantora
<ul style="list-style-type: none"> <li>- nome: String</li> <li>- cpf: String</li> <li>- dataNascimento: String</li> <li>- empresa: String</li> <li>- linguagensProgramacao: String[]</li> </ul>	<ul style="list-style-type: none"> <li>- nome: String</li> <li>- cpf: String</li> <li>- dataNascimento: String</li> <li>- grupo: String</li> <li>- estilosDeDanca: String[]</li> </ul>	<ul style="list-style-type: none"> <li>- nome: String</li> <li>- cpf: String</li> <li>- dataNascimento: String</li> <li>- banda: String</li> <li>- generosMusicais: String[]</li> </ul>
<ul style="list-style-type: none"> <li>+ aprender(): void</li> <li>+ caminhar(): void</li> <li>+ comer(): void</li> <li>+ codificar(): void</li> </ul>	<ul style="list-style-type: none"> <li>+ aprender(): void</li> <li>+ caminhar(): void</li> <li>+ comer(): void</li> <li>+ dançar(): void</li> </ul>	<ul style="list-style-type: none"> <li>+ aprender(): void</li> <li>+ caminhar(): void</li> <li>+ comer(): void</li> <li>+ cantar(): void</li> </ul>

Sabendo que Programadora, Dançarina e Cantora são pessoas, siga os passos abaixo:

**Exercício 1** - Remodele o projeto pensando no conceito de herança.



(infelizmente tem que dar bastante zoom pra ver sorry)

**Exercício 2** – Considere o código a seguir, que calcula a área dada uma forma geométrica:

```

1 public class FormaGeometrica {
2     private double base;
3     private double altura;
4     private double raio;
5
6     public FormaGeometrica(double base, double altura, double raio) {
7         this.base = base;
8         this.altura = altura;
9         this.raio = raio;

```

```

10 }
11
12 public double area(String formaGeometrica) {
13     double area = 0;
14     if(formaGeometrica.equals("Quadrado")) {
15         area = this.base * this.altura;
16     }else if(formaGeometrica.equals("Triângulo")) {
17         area = (this.base * this.altura) / 2;
18     }else if(formaGeometrica.equals("Círculo")) {
19         area = Math.PI * Math.pow(this.raio, 2);
20     }else {
21         System.out.println("Forma geométrica não conhecida");
22     }
23     return area;
24 }
25 }

```

```

1 public class Teste {
2     public static void main(String[] args) {
3         FormaGeometrica quadrado = new FormaGeometrica(2, 2, 0);
4         FormaGeometrica triangulo = new FormaGeometrica(2, 2, 0);
5         FormaGeometrica circulo = new FormaGeometrica(0, 0, 2);
6         System.out.println(quadrado.area("Quadrado"));
7         System.out.println(triangulo.area("Triângulo"));
8         System.out.println(circulo.area("Círculo"));
9     }
10 }

```

O código apresentado acima é possível ser reescrito utilizando herança e/ou polimorfismos de sobrescrita e/ou sobrecarga (**sem perder o sentido**)? Se não, justifique sua resposta. Se sim, implemente uma solução para o problema usando herança e/ou polimorfismo.

**Exercício 3** – Considere o código a seguir, que escreve o nome do dia da semana dado um número inteiro e/ou o idioma, caso nenhum idioma seja passado ele imprime em português:

```

1 public class Semana {
2     public void imprimeDiaSemana(int dia, String idioma) {
3         if(idioma.equals("pt-br") || idioma.equals("")) {
4             switch (dia) {
5                 case 1: System.out.println("Domingo"); break;
6                 case 2: System.out.println("Segunda-feira"); break;
7                 case 3: System.out.println("Terça-feira"); break;
8                 case 4: System.out.println("Quarta-feira"); break;
9                 case 5: System.out.println("Quinta-feira"); break;
10                case 6: System.out.println("Sexta-feira"); break;
11                case 7: System.out.println("Sábado"); break;
12            }
13        } else if(idioma.equals("fr")) {
14            switch (dia) {
15                case 1: System.out.println("Sunday"); break;
16                case 2: System.out.println("Monday"); break;
17                case 3: System.out.println("Tuesday"); break;
18                case 4: System.out.println("Wednesday"); break;
19                case 5: System.out.println("Thursday"); break;
20                case 6: System.out.println("Friday"); break;

```

```
21 case 7: System.out.println("Saturday"); break;
22 }
23 }
24 }
25 }
```

\* Note que o parâmetro idioma é opcional, já que na linha 3 existe uma verificação caso ele seja "" (String vazia).

O código apresentado acima é possível ser reescrito utilizando herança e/ou polimorfismos de sobrescrita e/ou sobrecarga (**sem perder o sentido**)? Se não, justifique sua resposta. Se sim, implemente uma solução para o problema usando herança e/ou polimorfismo.

**Exercício 4** – Considere o código a seguir, que retorna o quanto uma pessoa deve pagar de imposto mensal no ano de 2022:

```
1 public class IRPF {
2     public static double calculaImposto(double salario) {
3         if (salario <= 1903.98)
4             return 0;
5         else if (salario > 1903.98 && salario <= 2826.65)
6             return (salario * 0.075) - 142.80;
7         else if (salario > 2826.65 && salario <= 3751.05)
8             return (salario * 0.15) - 354.80;
9         else if (salario > 3751.05 && salario <= 4664.68)
10            return (salario * 0.225) - 636.13;
11        else
12            return (salario * 0.275) - 869.36;
13    }
14 }
```

O código apresentado acima é possível ser reescrito utilizando herança e/ou polimorfismos de sobrescrita e/ou sobrecarga (**sem perder o sentido**)? Se não, justifique sua resposta. Se sim, implemente uma solução para o problema usando herança e/ou polimorfismo.

**Exercício 5** – Analise os exercícios 1, 2, 3 e 4 quanto aos princípios SOLID. Verifique quais deles foram feridos (tanto no enunciado quanto na solução) e como eles poderiam ser resolvidos em relação aos exercícios 1, 2, 3 e 4