## Auto-peças

método fábrica

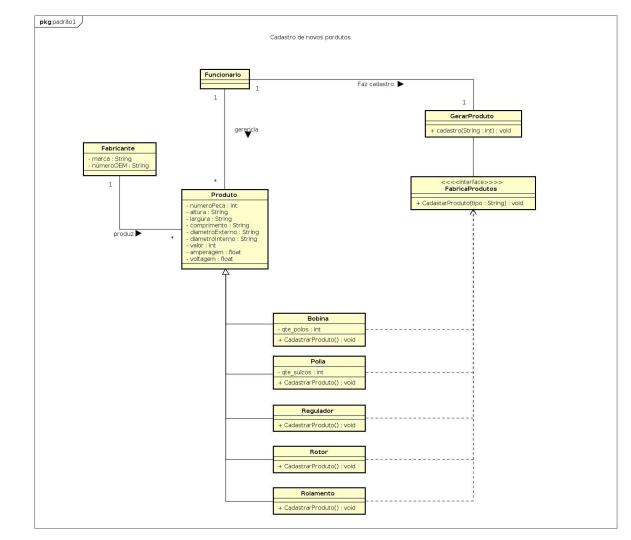
## Motivo pela escolha do método

O funcionário deve cadastrar (n) produtos, entre eles: polia, rotor, regulador, bobina e rolamento.

Bem repetitivo e variante.

Unificar as tarefas e agrupar

## Diagrama



## Código

```
package autopecas.dominio;
public class GerarProduto {
   public static FabricaProdutos cadastro(String tipo){
        FabricaProdutos novo produto= null;
        if(tipo.equalsIgnoreCase("polia")){
           novo produto = new Polia();
            novo produto.CadastrarProduto(tipo);
        else if(tipo.equalsIgnoreCase("rotor")){
            novo produto = new Rotor();
            novo produto.CadastrarProduto(tipo);
        else if (tipo.equalsIgnoreCase("regulador") ){
            novo produto = new Regulador();
            novo produto.CadastrarProduto(tipo);
       else if (tipo.equalsIgnoreCase("rolamento")){
            novo_produto = new Rolamento();
            novo produto.CadastrarProduto(tipo);
        else if (tipo.equalsIgnoreCase("bobina")){
            novo produto = new Bobina();
            novo produto.CadastrarProduto(tipo);
        return novo produto;
```

```
package autopecas.dominio;
public class Rotor extends Produto implements FabricaProdutos{

@Override
public void CadastrarProduto(String tipo) {
    System.out.println("rotor");
    throw new UnsupportedOperationException("Not supported yet.");
}
```

```
package autopecas.dominio;

public class Rolamento extends Produto implements FabricaProdutos{

    @Override
    public void CadastrarProduto(String tipo) {
        System.out.println("rolamento");
        throw new UnsupportedOperationException("Not supported yet.");
    }

}
```

etc...

para mais informações

https://github.com/MarceloMendes94/Autopecas