

from abc import ABC

class Usuário (ABC):

Atributos:

- nome, idade, __salário, id

Método:@property salário

Método @abstractmethod

```
def __str__:
```

```
pass
```

classes derivadas:

- Engenheiro
 - Atributo extra: especialidade
 - Método –str–

```
return f "Engenheiro: {self.nome}, Idade: {self.idade}, Salário {self.salário}, Id {self.Id}, Especialidade: {self.especialidade}"
```
- Técnico
 - Atributo extra: atuação
 - Método –str–

```
return f "Técnico: {self.nome}, Idade: {self.idade}, Salário {self.salário}, Id {self.Id}, Atuação: {self.atuação}"
```
- Supervisor
 - Método –str–

```
return f" Supervisor: {self.nome}, Idade: {self.idade}, Salário {self.salário}, Id {self.Id}, Especialidade: {self.especialidade}"
```

Usina:

Atributos:

- Capacidade (Float)
- __Desgaste (int de 0 a 100) - começa com 0
- __Eficiência (Float de 0 a 1) - começa com 0.2
- __Resíduos (List) - começa vazia
- Estado (Bool) - começa false
- Produção (Float) - Capacidade * eficiência

def __init__(self, capacidade)

Métodos:

- Gerar_energia
 - apenas o supervisor pode ativar esse método
 - faz com que o desgaste aumenta e gera resíduos
 - a usina só pode operar caso a lista de resíduos possua, no máximo, 10 itens
 - Ao atingir 10 itens na lista de resíduos, ela é desligada automaticamente
 - A cada uso da função, 1 resíduo é gerado e desgaste é acrescido de 10
 - Caso atinja 100 de desgaste, para de operar
 - A eficiência diminui em 5% para cada 10 de desgaste.
- Manutenção
 - feita pelo técnico
 - diminui o desgaste
 - só pode ser feito com a usina desligada
- Otimização
 - feita pelo supervisor
 - escolher metodologia (*Lean Manufacturing, Six Sigma, Kaizen, 5S, Total Productive Maintenance (TPM)*)
 - definir quanto cada sistema vai incrementar no nível de eficiência
 - dá bônus em eficiência máx e eficiencia
- Gestão de resíduos
 - feita por engenheiros
 - ela limpa a lista de resíduos, possibilitando a usina operar novamente
- Automação Industrial
 - feita por engenheiro ou técnico
 - atributo:
 - sistema_automacao: especifica o tipo de sistema de automação (ex.:PLC, SCADA, DCS, HMI)
 - definir quanto cada sistema vai incrementar no nível de eficiência
 - nivel_automacao: nível de automação do projeto (Baixo, Médio, Alto)
 - definir quanto cada nível vai incrementar (ex: se for baixo, vai ser 1,1x do que o sistema fornece, se for médio, vai ser 1,4x...)
 - método:

- `calcular_eficiencia_automacao()`: Calcula a eficiência do sistema de automação com base na eficiência atual, no `sistema_automacao` e no nível de automação.
- property para os 3 privados
- exibir ficha