

1. Carregue para memória a BC definida no ficheiro bicicletas.txt, que contém informação sobre a especificação dos diferentes componentes de uma bicicleta.
2. Escreva o predicado **gerar_elemento** que cria na base de conhecimento factos dinâmicos do tipo elemento/1 que guardam os elementos que existem na árvore de produto. Assim, após executar o predicado **gerar_elemento** a resposta à seguinte questão deverá ser:

```
?- elemento(Elemento).  
Elemento = aperto_rapido ;  
Elemento = aperto_rapido_selim ;  
Elemento = aro ;  
Elemento = avanco_guiador ;  
...
```

3. Escreva o predicado **produto_final(Elemento)** que permite pesquisar elementos que não entram na composição de outros.

```
?- produto_final(X).  
X = bicicleta ;  
false.
```

4. Escreva o predicado **produto_base(Elemento)** que permite pesquisar elementos que apenas entram na composição de outros.

```
?- produto_base(X).  
X = aperto_rapido ;  
X = aperto_rapido_selim ;  
X = aro ;  
X = avanco_guiador  
...
```

5. Escreva o predicado **produto_intermedio(Elemento)** que permite pesquisar elementos que entram na composição de outros e que são formados por outros elementos.

```
?- produto_intermedio(X).  
X = conjunto_selim ;  
X = conjunto_transmissao ;  
X = conjunto_travagem ;  
X = direccao ;  
X = pedaleiro ;  
X = quadro  
...
```

6. Escreva o predicado **nivel(ElementoX,ElementoY,Nivel)** que permite determinar a profundidade que o ElementoY está na árvore de produto relativamente ao ElementoX, considere que a raiz está no nível zero.

7. Escreva o predicado **dois_mais_profundos(ElementoRaiz,(EMax1,Prof1),(EMax2,Prof2))** que permite determinar os dois elementos, e respectivas profundidades, que estão a maior profundidade na árvore de produto ElementoRaiz.

8. Escreva o predicado **reg_custo(Elemento,Custo)** que permite criar um facto (dinâmico) com o custo de um componente, no caso de já existir um custo para esse componente deve ser

atualizado. Assim, a chamada do predicado `reg_custo(pedal, 32)` deve dar origem à criação do facto `custo(pedal, 32)` ou à sua atualização.

9. Escreva o predicado **`calc_custo(Elemento, Valor, LElemSemCusto)`** que permite calcular o custo de um Elemento em função das quantidades e custos dos subprodutos que integram a árvore do referido elemento; os subprodutos que não tenham custo definido devem ser colecionados na lista `LElemSemCusto`.
10. Escreva o predicado **`lista_materiais(Elemento, Qtd)`** que permite listar os produtos base e respectivas quantidades e custos envolvidos na produção de uma Qtd de Elemento por ordem decrescente de custo. A estrutura da listagem deverá ser:

```
Elemento : Qtd
-----
produto_base1 , qtd1, custo1
produto_base2 , qtd2, custo2
...
```

11. Escreva o predicado **`produto_arvore(Elemento, Arvore)`** que permite construir uma estrutura com a árvore de produto a partir de do Elemento com a seguinte estrutura:

```
%(prod1,[(prod11,[(prod111),((prod112,[(prod1121),(prod1122))])),(prod12,...),(prod13,...)])
```

```
?- produto_arvore(conjunto_transmissao,L).
L= [ (conjunto_transmissao,[(pedaleiro,[pedal,braco_pedal,rolamento,prato]),
corrente,desviador_traseiro,desviador_dianteiro,cassete,
(mudancas_dianteira,[manete_dianteira,bicha,cabo]),
(mudancas_traseira,[manete_traseira,bicha,cabo])])]
```

12. Escreva o predicado **`listar_arvore_identada(Elemento, Op_pi, Op_pb, Op_qtd)`** que permite listar de forma indentada (tipo estrutura de diretórios) a árvore de produto do Elemento; `Op_pi` define se são apresentados os produtos intermédios (on/off); `Op_pb` define se são apresentados os produtos base (on/off); `Op_qtd` define se são apresentados as quantidades (on/off)

```
?- listar_arvore_identada(conjunto_transmissao,on,on,off).
conjunto_transmissao
  pedaleiro
    pedal
    braco_pedal
    rolamento
    prato
  corrente
  desviador_traseiro
  desviador_dianteiro
  cassete
  mudancas_dianteira
    manete_dianteira
    bicha
  ...
```

13. Escreva o predicado **`guardarBaseConhecimento(Nome)`** que permite guardar a base de conhecimento num ficheiro de texto (incluindo os predicados criados dinamicamente).