

- Com base na **modelagem**, em Prolog, feita inicialmente (trabalho 1, partes A e B) de um “Curriculum Vitae” (CV), contendo os seguintes dados para, ao menos, 10 pessoas:
  - Nome completo/pessoa
  - Data de nascimento
    - \* dia
    - \* mês
    - \* ano
  - Cidade
  - Telefone (apenas números, sendo os dois primeiros para o código DDD)
  - Formação (podem ser declarados vários cursos)
    - \* Nome do curso
    - \* Nome da instituição
  - \* Nome completo do orientador(a) (se houver)
  - \* Nome completo de colega(s) de curso ou professor(es) como referência(s)
  - \* Ano de ingresso
  - \* Ano de término
  - Experiência (podem ser declarados vários empregos)
    - \* Nome da empresa
    - \* Nome do cargo
    - \* Nome completo de colega(s) de trabalho como referência(s)
    - \* Ano de ingresso
    - \* Ano de término
- Implemente um banco de dados (BD) em Prolog, considerando a criação e atualização de um ou mais arquivos em memória secundária (disco). As seguintes funções devem ser possíveis:
  - Inclusão de dados pelo nome (ou alguma chave primária)
  - Exclusão de dados pelo nome (ou alguma chave primária)
  - Alteração de dados (pode ser apenas da cidade e do telefone) pelo nome (ou alguma chave primária)
  - Consultas de formas variadas para obtenção de um ou mais CVs:
    - \* Por nome
    - \* Por cidade
    - \* Pelo nome da instituição de ensino
    - \* Pelo nome da empresa
    - \* Crie uma consulta qualquer com ao menos dois campos combinados (exemplo: pessoas de Florianópolis com mais de 60 anos)

Um arquivo deve ser lido e carregado para a memória Prolog e, assim que as manipulações forem finalizadas, os dados devem ser salvos no mesmo arquivo. Comandos a pesquisar: `assert`, `retract`, `open`, `close`, `see`, `seen`, `findall`, `append`, `select`, ...

- Dicas:
  - Utilização de `dynamic` para o `assert` funcionar, e consulta ao banco de dados (`c:/local/banco.pl`):

```
:- dynamic progenitor/2.  
progenitor(maria,joao).  
progenitor(jose,joao).  
progenitor(maria,ana).  
progenitor(jose,ana).
```

Consulta ao banco de dados e acréscimo de uma cláusula:

```
?- consult('c:/local/banco.pl')  
?- assert(progenitor(fulano,ciclano)).  
?- tell('c:/local/banco.pl'), listing(progenitor), told.
```

---

– Leitura e gravação de arquivo:

```
?- open('c:/local/base.txt', write, X), write(X,'abc.\nxyz.\nmnop.'), close(X).  
X = <stream>(01FD9900).
```

```
?- open('c:/local/base.txt', read, X), read(X,Y1), read(X,Y2), read(X,Y3), close(X),  
    write(Y1), nl, write(Y2), nl, write(Y3).
```

```
abc
```

```
xyz
```

```
mnop
```

```
X = <stream>(01FD8B58),
```

```
Y1 = abc,
```

```
Y2 = xyz,
```

```
Y3 = mnop.
```

• Entrega do  $T_2$ –parte A:

- **Prazo:** dia 21mai2015 até 23h55
- **Forma:** Individual ou em grupo de até três alunos
- **Entrega pelo Moodle:**
  1. **Códigos** fontes (Prolog)
  2. **PDF** com explicações e exemplos de aplicação de cada regra e as respostas obtidas