

Trabalho 1

Alunos: Arthur de Freitas Dornelas - 201735004

Vinicius da Cruz Soranço - 201735003

1. Organização de pastas

Fizemos a separação do código em dois arquivos:

main.pl - Responsável pelo menu e chamada das funcionalidades.

database.pl - Responsável por guardar os fatos.

2. Estrutura dos fatos

- **curso(NomeDoCurso)** : representando o cada curso
- **disciplina(NomeDaDisciplina)**: representando cada disciplina existente
- **aluno(NomeDoAluno, Curso)**: representando cada aluno que estuda na instituição
- **disciplinaCurso(Curso, Disciplina, Período)**: representando a relação entre uma disciplina e um curso, tal qual seu período associado.
- **cursou(NomeDoAluno, Disciplina, Nota)**: representando a disciplina que um aluno já cursou, contendo sua nota, e a disciplina.

3. Implementação do menu

Para o menu utilizamos uma operação de repeat que aguarda a seleção da opção pelo usuário. Na opção "0" temos a funcionalidade de "Gerenciar Dados" que é onde será inserido, editado ou removido os fatos. Nas opções seguintes temos as funcionalidades do sistema.

4. Gerenciar Dados

Para mostrar como foi realizado a parte de gerenciamento dos dados, consideremos o gerenciamento do estudante:

Na opção de adicionar estudante, colocamos a opção de o usuário digitar o nome e curso e utilizamos o assert para inserir o fato novo em **aluno(Nome, Curso)**.

Já para opção de editar o estudante, pedimos ao usuário o nome do estudante e do curso e utilizamos o retract para apagar o aluno atual com esse nome e após pedir o novo nome e curso utilizamos o assert para inseri-lo novamente.

Agora para a opção de remover, realizamos apenas o retract de no fato do aluno e curso inserido.

Para "Curso", "Disciplina" e para os relacionamentos "Cursou" a "DisciplinaCurso" elas funcionam da mesma forma, com exceção do seguinte:

- Ao deletar um curso, realizamos também o `retractAll` dos alunos desse curso, sendo todos apagados também.
- Ao deletar uma disciplina, realizamos o `retractAll` do fato ***curso***(_, ***Disciplina***), apagando todos os dados de alunos que cursaram a disciplina.

5. Funcionalidades

- Histórico Escolar

Para essa funcionalidade, pedimos ao usuário que insira o nome do estudante e a partir dele utilizamos na query:

curso(***Estudante***, ***Disciplina***, ***Nota***), ***aluno***(***Estudante***, ***Curso***), ***disciplinaCurso***(***Curso***, ***Disciplina***, ***Periodo***)

Ela funciona de forma que fornece as disciplinas do estudante através do ***curso*** com sua respectiva nota, verificando no fato ***aluno*** o estudante que foi digitado. Também utilizamos o ***disciplinaCurso*** para termos acesso ao período da disciplina.

- Matriz Curricular de um Curso

Foi utilizado para essa funcionalidade a query:

disciplinaCurso(***Curso***, ***Disciplina***, ***Periodo***)

Que já nos informa exatamente as disciplinas do curso digitado pelo usuário e o período.

- Relação de estudantes que já cursaram uma dada disciplina, podendo ou não incluir um critério de seleção por nota

O usuário digita a disciplina que ele quer e se ele quer a seleção por nota.

Caso queira nota recebemos em ***Grade*** e fazemos a query:

curso(***Estudante***, ***Disciplina***, ***Grade***)

Caso não queira realizamos a query:

curso(***Estudante***, ***Disciplina***, ***Nota***)

Dessa forma dispomos o nome do estudante e sua nota na disciplina escolhida.

- Relação de disciplinas que faltam ser cursadas para um dado estudante

Nessa funcionalidade utilizamos a funcionalidade de lista através do **findall**.

Primeiro criamos a lista das disciplinas que estão presentes no curso do estudante, através da query:

findall(DisciplinaNaoCursada, disciplinaEstudante(Estudante, DisciplinaNaoCursada), DisciplinasCurso)

Após isso criamos a lista das disciplinas que o estudante já cursou através da query:

findall(DisciplinaCursada, cursou(Estudante, DisciplinaCursada, _), DisciplinasCursadas)

Possuindo essas duas listas agora só precisamos fazer a diferença da primeira lista para a segunda lista, utilizando o **subtract**, de forma que a lista resultante são das disciplinas que faltam ser cursadas:

subtract(DisciplinasCurso, DisciplinasCursadas, R)

- Relação de estudantes de um dado curso, podendo incluir critério de seleção por nota em disciplina ou IRA

Primeiramente, perguntando se o usuário deseja usar um critério ou não. Caso queira, o critério por nota, usamos um **findAll** para buscar todos os alunos de um determinado curso, de acordo com a entrada do usuário.

findall(Estudante, func(Curso, Estudante), Notas)

Após isso usamos um **sort()** para remover o itens duplicados, caso existam. Caso o usuário digite 'c', todos os alunos do curso serão buscado a partir do fato de **aluno**.

- Relação de cursos que contém uma dada disciplina

O usuário digita a disciplina que ele deseja saber e através disso realizamos a query:

disciplinaCurso(Curso, Disciplina, _), curso(Curso)

Através delas temos o fato da **disciplinaCurso** que irá selecionar todos os relacionamentos de curso dessa disciplina e o **curso** que irá selecionar o curso da relação para ser impresso.

Dessa forma temos a exposição dos cursos que têm a disciplina.

6. Como executar

Para executar o programa basta abrir o projeto no programa SWI-prolog, ou executar a seguinte linha de comando no terminal:

```
swipl /path/para/main.pl
```

Obs: é preciso ter o [swipl](#) instalado no computador.