

# Trabalho de Linguagem de Programação

---

## Sistema de Gestão Acadêmico em Prolog

*Alunos: Lucas Piazzzi de Castro - 201635003 e Cristiano Nascimento - 201635029*

### Especificações

O sistema desenvolvido possui as seguintes funcionalidades:

#### Predicados

- Consulta de histórico escolar de um estudante; ✓
- Consulta da matriz curricular de um curso; ✓
- Consulta da relação de disciplinas que faltam ser cursadas para um dado estudante ✓
- Consulta da relação de cursos que contém uma dada disciplina. ✓
- Consulta da relação de estudantes de um dado curso, podendo incluir critério de seleção por nota em disciplina ou IRA; ✓
- Consulta da relação de estudantes que já cursaram uma dada disciplina, podendo ou não incluir um critério de seleção por nota; ✓

#### CRUD

- É possível cadastrar, remover ou editar novos cursos, disciplinas, estudantes e relacionamentos entre eles. ✓

#### Persistência

- As modificações realizadas serão salvas no programa para serem carregadas nos próximos usos. ✓

# Estratégias Utilizadas

## Metodologia

Antes de começar a parte prática do trabalho, buscamos nos fundamentar nas bases teóricas de disciplinas cursadas previamente por nós, principalmente em banco de dados e modelagem de sistemas. Então, construímos um pequeno diagrama de classes e banco de dados para nos ajudar a visualizar a estrutura do programa que seria construído. Consideramos as tabelas e classes como se fossemos implementar um sistema web para as especificações citadas, contendo chaves primária e estrangeira, além dos tipos de atributos em MySQL. Porém, esse tipo de especificação não é necessária no Prolog, decidimos colocar para o trabalho ficar mais completo e poder reforçar nossos conhecimentos aprendidos anteriormente em outras disciplinas.

Todo código foi versionado utilizando Git, e foi depositado no seguinte repositório do Github: [https://github.com/RagnarCris/Gestao\\_Academica\\_Prolog](https://github.com/RagnarCris/Gestao_Academica_Prolog). Utilizamos um arquivo .txt (*manipulacao\_banco.txt*) como um histórico de manipulação do banco de dados inicial. Não optamos por desenvolver uma versão gráfica do sistema por questão de tempo.

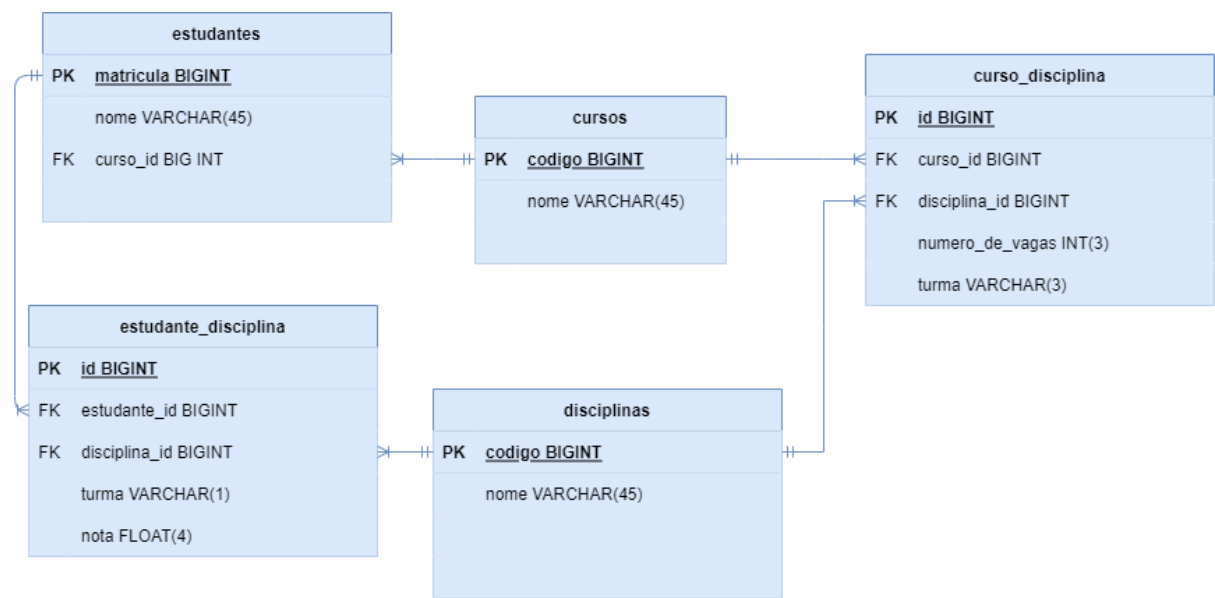
## Código

Optamos por separar o projeto em dois arquivos, `database.pl` e `gestao_academica.pl`, pois logo no início do trabalho ao tentarmos métodos salvar os dados inseridos, o código principal estava sendo reescrito com diversos erros. Portanto, foi necessário a separação dos dados do programa principal. Fazer isso também ajuda na organização do projeto. Outro fator importante para a organização é a escolha do padrão `snake_case` para nomear os predicados, e o padrão `PascalCase` para os parâmetros das funções. Com o aumento do tamanho do código, a utilização desses dois padrões nos ajudou a identificar mais facilmente as partes do programa.

Para tratar a persistência utilizamos o `listing` para cada predicado, de forma que tivemos que criar um arquivo de dados para cada “tabela” do banco. E esses arquivos não possuem comentários pois ao salvar o arquivo, os comentários são perdidos. Por isso deixamos o arquivo chamado `database_inicial.pl` ainda na pasta, ele é o primeiro arquivo de dados feito pela dupla.

# Diagramas

## Diagrama de Banco de Dados



## Diagrama de Classes

