**DCC060 – Trabalho Prático**

1. **INTRODUÇÃO**   
    O sistema tem como objetivo a integração com o SIGA (Sistema Integrado de Gerenciamento Acadêmico) para realizar a reserva e gerenciamento de salas de departamentos da UFJF. As salas poderão ser reservadas pelos professores para ministrar disciplinas ou para projetos de extensão. Somente os professores poderão reservar as salas, mas eles poderão inscrever alunos nos Projetos de Extensão e nas Disciplinas, podendo os alunos verem os horários e locais das suas atividades.

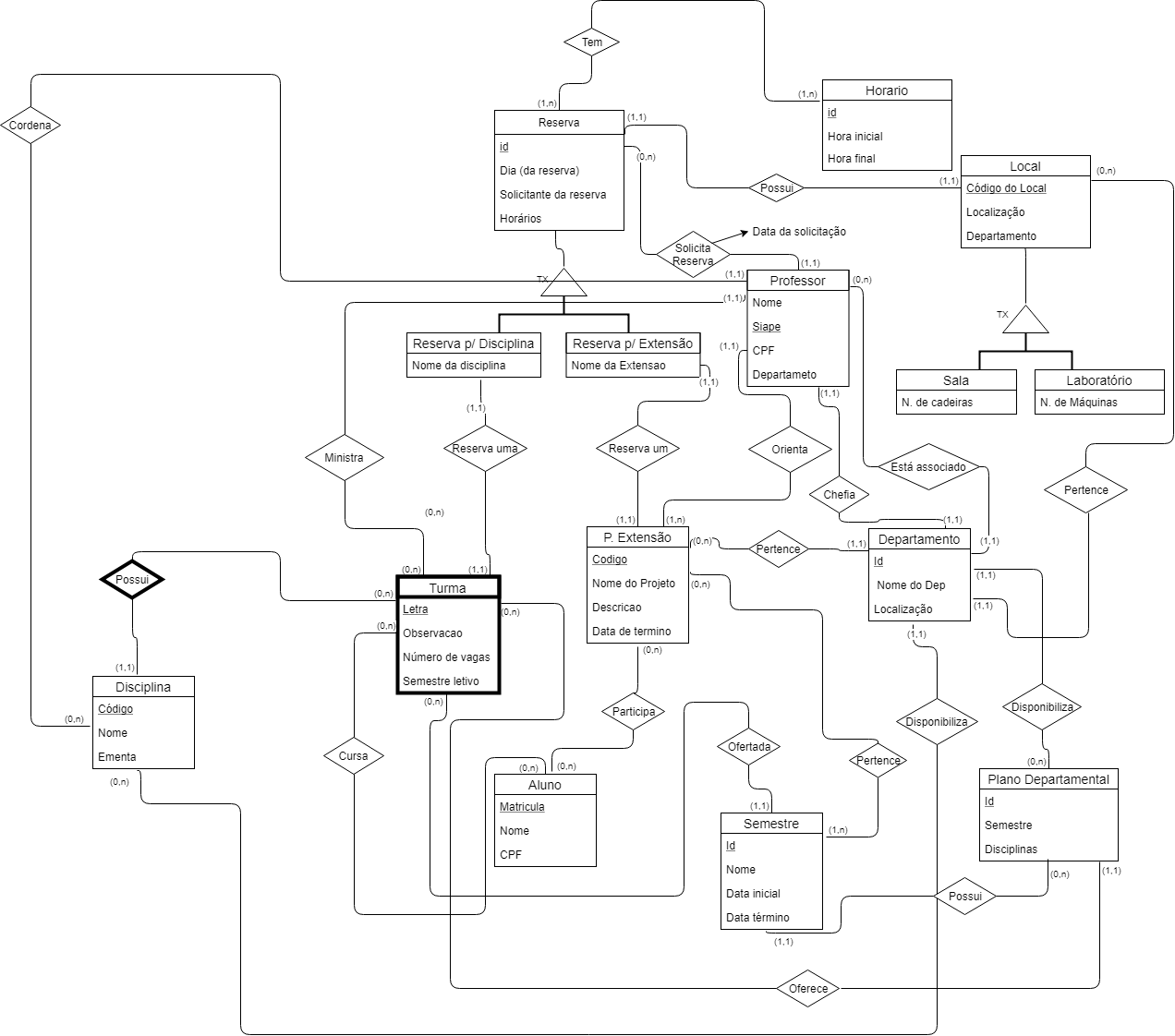
O projeto irá gerenciar uma base de dados que será responsável por armazenar departamentos, disciplinas, projetos de extensão, alunos e professores, semestres, locais onde poderão ser feitas as reservas e planos departamentais e será responsável por garantir que não haja duas reservas da mesma sala no mesmo tempo.  
  
**2. REQUISITOS DO PROJETO**   
Esse projeto irá facilitar a reserva de salas da UFJF, para isso os seguintes requisitos devem ser atendidos.

|  |
| --- |
| **Regras de Negócio** |
| 1. Um professor pode reservar uma ou mais salas. 2. Cada reserva pode ter somente um local. 3. Um local não pode ser reservado por mais de um professor em um mesmo horário. 4. Uma reserva deve ter um professor, local e horários. 5. Os tipos de reserva são “Reservas p/ Disciplina” ou “Reservas p/ Extensão”. 6. O sistema deve ser capaz de exibir todas as salas. 7. O sistema deve ser capaz de exibir todas as salas reservadas por um determinado professor. 8. As reservas que forem de o tipo “Ministrar disciplina” devem conter o código da disciplina cadastrado. 9. As reversas que forem do tipo “Atividades de Extensão” devem conter o código da atividade de extensão. 10. Uma reserva pode ter seus horários alterados. 11. Um aluno não pode realizar reservas. 12. Um aluno pode estar inscrito em Disciplinas e Projetos de Extensão. 13. A sala sempre estará disponível nos horários em que ela não possui reserva. 14. Um departamento possui vários professores, disciplinas, projetos de extensão e planos departamentais. 15. Uma disciplina está alocada em um semestre, é ofertada por um Departamento e ministrada por um professor. 16. Professores podem cadastrar disciplinas e projetos de extensão. 17. Um aluno pode ver quais reservas ele está alocado (Disciplinas e projetos de extensão com reservas de sala). 18. Os locais disponíveis para reserva são do tipo Sala e Laboratório. 19. Um plano departamental é distribuído em um semestre letivo por um Departamento. 20. Um professor pode ministrar uma ou mais disciplinas. 21. O sistema deve ser capaz de cadastrar Departamento, Disciplinas, Plano Departamental, Semestre, Alunos, Professores, Horários e Reservas. 22. Um projeto de extensão pode ser orientado por somente um Professor. 23. Deve ser possível ter acesso a todas as salas reservadas para Disciplinas e Projetos de Extensão de um Semestre. 24. Deve ser possível ter acesso a todas as disciplinas de um plano departamental. 25. Um Projeto de Extensão está alocado em um ou mais semestres. 26. Uma reserva pode durar somente um dia, podendo ter vários horários nesse dia. 27. O professor deve ter acesso a todas as suas reservas. 28. Um professor não pode ter duas ou mais reservas do tipo “Reservas p/ Disciplina” no mesmo horário (mesmo em salas e disciplinas diferentes). 29. Um professor pode coordenar uma Disciplina e/ou Chefiar um Departamento. |

**3. MODELAGEM RELACIONAL** 

**3.1 Diagrama Entidade Relacionamento**

Segue abaixo um modelo Diagrama Entidade-Relacionamento da aplicação de reserva de salas.



**3.2 Modelagem lógica**

Horário(id\_horario, hora\_inicial, hora\_final)

Reserva(id\_reserva, dia, **id\_professor**, **id\_local,** data reserva)

* id\_professor ref. Professor
* id\_local ref. Local

Reserva\_tem\_Horarios(**id\_reserva**, **id\_horario**)

* id\_reserva ref. Reserva
* id\_horario ref. Horário

Reserva p/ Disciplina(**id\_reserva,** **id\_turma**, nome disciplina)

* id\_reserva ref. Reserva
* id\_turma ref. Disciplina

Reserva p/ Extensão (**id\_reserva, id\_projeto**, nome projeto)

* id\_reserva ref. Reserva
* id\_projeto ref. Projeto

Local(id\_local, localizacao, **id\_departamento**)

* id\_departamento ref. Departamento

Sala(**id\_local**, num cadeiras)

* id\_local ref. Local

Laboratório(**id\_local**, num maquinas)

* id\_local ref. Local

Professor(id\_professor, nome professor, siape, cpf, **id\_departamento**)

* id\_departamento ref. Departamento

Departamento(id\_departamento, nome departamento, localizacao, **id\_professor**)

* id\_chefe ref. Professor

Disciplina(id\_disciplina, nome disciplina, ementa, **id\_coordenador, id\_departamento**)

* id\_coordenador ref. Professor
* id\_departamento ref. Departamento

Turma(id\_turma, letra, observação, numero\_de\_vagas, **id\_semestre**, **id\_disciplina, id\_professor**)

* id\_semestre ref. Semestre
* id\_disciplina ref. Disciplina
* id\_professor ref. Professor

Projeto Extensão(id\_projeto, nome projeto, descrição, data\_termino**, id\_departamento, id\_professor**)

* id\_orientador ref. Professor
* id\_departamento ref. Departamento

Aluno(id\_aluno, matricula, nome aluno, CPF, curso)

Semestre(id\_semestre, nome semestre, data\_inicial, data\_final)

Projeto\_tem\_Semestre(**id\_projeto, id\_semestre**)

* id\_projeto ref. Projeto Extensao
* id\_semestre ref. Semestre

Aluno\_Participa\_Projeto(**id\_aluno, id\_projeto**)

* id\_aluno ref. Aluno
* id\_projeto ref. Projeto

Aluno\_Cursa\_Turma(**id\_aluno, id\_turma**)

* id\_aluno ref. Aluno
* id\_turma ref. Turma

Plano Departamental(id\_plano, **id\_semestre**, **id\_Departamento**,nome plano dep)

* id\_semestre ref. Semestre
* id\_Departamento ref. Departamento

PlanoDepartamental\_Tem\_Turmas(**id\_plano, id\_turma**)

* id\_plano ref. Plano Departamental
* id\_disciplina ref. Turma

**4 CONSULTAS EM ÁLGEBRA RELACIONAL**(devem cobrir todas as sintaxes e casos-exemplo vistos em sala de aula; necessariamente incluir junção, diferença e divisão; explicitar a pergunta que define a consulta e a sequência de operações em Álgebra Relacional necessária para conseguir a resposta à pergunta) 

|  |
| --- |
| **Consulta 1: “Listar todas as reservas com sua data e professor”** |
|  |

|  |
| --- |
| **Consulta 2: “Listar todos os professores, mostrando seu nome, SIAPE, CPF e seu departamento”** |
|  |

|  |
| --- |
| **Consulta 3: “Listar todas as disciplinas, mostrando seu nome, coordenador e departamento”** |
|  |

|  |
| --- |
| **Consulta 4: “Listar todas as turmas, mostrando sua Letra, disciplina, observação, numero de vagas, semestre e professor”** |
|  |

|  |
| --- |
| **Consulta 5: “Listar todas as salas, mostrando seu id e numero de cadeiras”** |
|  |

|  |
| --- |
| **Consulta 6: “Listar todos os Laboratórios, mostrando seu id e o numero de maquinas”** |
| FOREIGN |

|  |
| --- |
| **Consulta 7: “Listar todos os projetos de extensão, mostrando nome do projeto, descrição, data de termino, departamento e orientador”** |
|  |

|  |
| --- |
| **Consulta 8: “Listar todas as turmas que um determinado aluno cursa, mostrando sua letra, disciplina, semestre letivo e professor”** |
|  |

|  |
| --- |
| **Consulta 9: “Listar todos os projetos de extensao que um determinado aluno participa, mostrando nome do projeto, descricao, data de termino, orientador e departamento”** |
|  |

|  |
| --- |
| **Consulta 10: “Listar todas as turmas que um determinado professor ministra, mostrando sua letra, disciplina, semestre letivo”** |
|  |

|  |
| --- |
| **Consulta 11: “Listar todos os projetos de extensão que um determinado professor orienta, mostrando nome do projeto, descrição, data de termino e departamento”** |
|  |

|  |
| --- |
| **Consulta 12: “Listar todos os horários de uma determinada reserva, mostrando o horário inicial e horário final”** |
|  |

|  |
| --- |
| **Consulta 13: “Listar todos os horários cadastrados, mostrando horário inicial e horário final”** |
|  |

|  |
| --- |
| **Consulta 14: “Listar todos os alunos, mostrando seu nome, matricula, CPF e curso”** |
|  |

|  |
| --- |
| **Consulta 15: “Listar todos os alunos de uma determinada turma, mostrando seu nome, matricula, CPF e curso”** |
|  |

|  |
| --- |
| **Consulta 16: “Quais alunos cursam as disciplinas de determinado professor”** |
|  |

**5 MODELO FÍSICO (Parte 1) E CARGA DE DADOS**

**Todos os itens abaixo devem ser descritos no relatório.**

**5.1 Tabelas**

|  |
| --- |
| **Tabela 1: horario** |
| CREATE TABLE horario(  id\_horario INT PRIMARY KEY,  hora\_inicial TIMESTAMP NOT NULL,  hora\_final TIMESTAMP NOT NULL  ); |

|  |
| --- |
| **Tabela 2: reserva** |
| CREATE TABLE reserva(  id\_reserva INT PRIMARY KEY,  dia DATE NOT NULL,  id\_professor INT NOT NULL REFERENCES professor (id\_profressor),  id\_local INT NOT NULL REFERENCES local (id\_local),  data\_reserva DATE NOT NULL  ); |

|  |
| --- |
| **Tabela 3: reserva\_tem\_horarios** |
| CREATE TABLE reserva\_tem\_horarios(  id\_reserva INT REFERENCES reserva (id\_reserva),  id\_horario INT REFERENCES horario (id\_horario),  PRIMARY KEY(id\_reserva, id\_horario)  ); |

|  |
| --- |
| **Tabela 4: reserva\_p\_disciplina** |
| CREATE TABLE reserva\_p\_disciplina(  id\_reserva INT NOT NULL REFERENCES reserva (id\_reserva),  id\_turma INT NOT NULL REFERENCES turma (id\_turma),  nome\_disciplina VARCHAR(200) NOT NULL,  PRIMARY KEY (id\_reserva, id\_turma)  ); |

|  |
| --- |
| **Tabela 5: reserva\_p\_extensao** |
| CREATE TABLE reserva\_p\_extensao(  id\_reserva INT NOT NULL REFERENCES reserva (id\_reserva),  id\_projeto INT NOT NULL REFERENCES projeto\_extensao (id\_projeto),  nome\_projeto VARCHAR(200) NOT NULL,  PRIMARY KEY (id\_reserva, id\_projeto)  ); |

|  |
| --- |
| **Tabela 6: local** |
| CREATE TABLE local(  id\_local INT PRIMARY KEY,  localizacao VARCHAR(300) NOT NULL,  id\_departamento INT NOT NULL REFERENCES departamento (id\_departamento)  ); |

|  |
| --- |
| **Tabela 7: sala** |
| CREATE TABLE sala(  id\_local INT REFERENCES local (id\_local),  num\_cadeiras INT NOT NULL,  PRIMARY KEY (id\_local)  ); |

|  |
| --- |
| **Tabela 8: laboratorio** |
| CREATE TABLE laboratorio(  id\_local INT REFERENCES local (id\_local),  num\_maquinas INT NOT NULL,  PRIMARY KEY (id\_local)  ); |

|  |
| --- |
| **Tabela 9: professor** |
| CREATE TABLE professor(  id\_professor INT PRIMARY KEY,  nome\_professor VARCHAR(200) NOT NULL,  siape VARCHAR(100) NOT NULL UNIQUE,  cpf VARCHAR(100) NOT NULL UNIQUE,  id\_departamento INT NOT NULL REFERENCES departamanto (id\_departamento)  ); |

|  |
| --- |
| **Tabela 10: departamento** |
| CREATE TABLE departamento(  id\_departamento INT PRIMARY KEY,  nome\_departamento VARCHAR(200) NOT NULL,  localizacao VARCHAR (300) NOT NULL,  id\_chefe INT REFERENCES professor (id\_professor)  ); |

|  |
| --- |
| **Tabela 11: disciplina** |
| CREATE TABLE disciplina(  id\_disciplina INT PRIMARY KEY,  nome\_disciplina VARCHAR(200) NOT NULL,  ementa TEXT NOT NULL,  id\_coornedador INT NOT NULL REFERENCES professor (id\_professor),  id\_departamento INT NOT NULL REFERENCES departamento (id\_departamento)  ); |

|  |
| --- |
| **Tabela 12: turma** |
| CREATE TABLE turma(  id\_turma SERIAL,  letra CHAR NOT NULL,  observacao TEXT,  numero\_de\_vagas INT NOT NULL CHECK(numero\_de\_vagas > 0),  id\_semestre INT NOT NULL REFERENCES semestre (id\_semestre),  id\_disciplina INT NOT NULL REFERENCES disciplina (id\_disciplina),  id\_professor INT NOT NULL REFERENCES professor (id\_professor),  vagas\_ocupadas INT NOT NULL DEFAULT 0 CHECK(vagas\_ocupadas <= numero\_de\_vagas AND vagas\_ocupadas >= 0),  UNIQUE(letra, id\_semestre, id\_disciplina),  PRIMARY KEY(id\_turma)  ); |

|  |
| --- |
| **Tabela 13: projeto\_extensao** |
| CREATE TABLE projeto\_extensao(  id\_projeto INT PRIMARY KEY,  nome\_projeto VARCHAR(300) NOT NULL,  descricao TEXT NOT NULL,  data\_termino DATE NOT NULL,  id\_departamento INT REFERENCES departamento (id\_departamento)  ); |

|  |
| --- |
| **Tabela 14: aluno** |
| CREATE TABLE aluno(  id\_aluno INT PRIMARY KEY,  matricula VARCHAR(20) UNIQUE NOT NULL,  nome\_aluno VARCHAR(300) NOT NULL,  cpf VARCHAR(100) NOT NULL UNIQUE,  curso VARCHAR(100) NOT NULL  ); |

|  |
| --- |
| **Tabela 15: semestre** |
| CREATE TABLE semestre(  id\_semestre INT PRIMARY KEY,  nome\_semestre VARCHAR(200) NOT NULL,  data\_inicial DATE NOT NULL,  data\_final DATE NOT NULL  ); |

|  |
| --- |
| **Tabela 16: projeto\_tem\_semestre** |
| CREATE TABLE projeto\_tem\_semestre(  id\_projeto INT REFERENCES projeto\_extensao (id\_projeto),  id\_semestre INT REFERENCES semestre (id\_semestre),  PRIMARY KEY(id\_projeto, id\_semestre)  ); |

|  |
| --- |
| **Tabela 17: aluno\_participa\_projeto** |
| CREATE TABLE aluno\_participa\_projeto(  id\_aluno INT NOT NULL REFERENCES aluno (id\_aluno),  id\_projeto INT NOT NULL REFERENCES projeto\_extensao (id\_projeto),  PRIMARY KEY(id\_aluno, id\_projeto)  ); |

|  |
| --- |
| **Tabela 18: aluno\_cursa\_turma** |
| CREATE TABLE aluno\_cursa\_turma(  id\_aluno INT NOT NULL REFERENCES aluno (id\_aluno),  id\_turma INT NOT NULL REFERENCES turma (id\_turma),  PRIMARY KEY(id\_aluno, id\_turma)  ); |

|  |
| --- |
| **Tabela 19: plano\_departamental** |
| CREATE TABLE plano\_departamental(  id\_plano INT PRIMARY KEY,  id\_semestre INT NOT NULL REFERENCES semestre (id\_semestre),  id\_departamento INT NOT NULL REFERENCES departamento (id\_departamento),  nome\_plano\_depa VARCHAR(200) NOT NULL  ); |

|  |
| --- |
| **Tabela 18: plano\_departamental\_tem\_turmas** |
| CREATE TABLE plano\_departamental\_tem\_turmas(  id\_plano INT NOT NULL REFERENCES plano\_departamental (id\_plano),  id\_turma INT NOT NULL REFERENCES turma (id\_turma),  PRIMARY KEY (id\_plano, id\_turma)  ); |

**5.2 Verificação**

|  |
| --- |
| **Tentativa 1: Tentativa de inserção de aluno com mesmo CPF** |
| INSERT INTO aluno (id\_aluno, matricula, nome\_aluno, cpf, curso) VALUES  (1,’201635002’,’Vinicius de Castro Sampaio’,’16064483741’,’Ciencia da Computacao’);  INSERT INTO aluno (id\_aluno, matricula, nome\_aluno, cpf, curso) VALUES  (2,’201654087’,’Joao das Aboboras’,’16064483741’,’Jornalismo’);  #A segunda insercao ocorrera error, pois o campo matricula e UNIQUE  Error: Duplicate entry ‘16064483741’ for key ‘cpf’. |

|  |
| --- |
| **Tentativa 2: Tentativa de remover uma disciplina** |
| DELETE FROM disciplina WHERE nome\_disciplina = “Estrutura de Dados I”;  #Se existir uma turma associada a esta disciplina, a consulta não será executada, pois antes de remover a disciplina e preciso remover as turmas associadas a mesma. |

|  |
| --- |
| **Tentativa 3: Tentativa de atualizar chave primaria de disciplina** |
| UPDATE disciplina SET id\_disciplina = 9 WHERE nome\_disciplina = “Estrutura de Dados I”;  #A consulta não ira funcionar pois ao mudar uma PK com referencias nos obtemos error. |

**5.3 Carga de Dados**

|  |
| --- |
| **Carma mínima** |
| #CARGA MÍNIMA SQL  #INSERÇÃO DE ALUNOS  INSERT INTO aluno (id\_aluno, matricula, nome\_aluno, cpf, curso) VALUES  (1,'201654087','Joao das Aboboras','47064483741','Jornalismo');  INSERT INTO aluno (id\_aluno, matricula, nome\_aluno, cpf, curso) VALUES  (2,'201635002','Vinicius de Castro Sampaio','16064483741','Ciencia da Computacao');  INSERT INTO aluno (id\_aluno, matricula, nome\_aluno, cpf, curso) VALUES  (3,'201874012','Maria Joaquina Isabela','47532498517','Sistemas de Informação');  #INSERÇÃO DE SEMESTRES  INSERT INTO semestre (id\_semestre, nome\_semestre,data\_inicial,data\_final) VALUES  (1,'2018-1','2018-01-01','2018-07-01');  INSERT INTO semestre (id\_semestre, nome\_semestre,data\_inicial,data\_final) VALUES  (2,'2018-2','2018-07-01','2018-12-01');  #INSERÇÃO DE DEPARTAMENTOS  INSERT INTO departamento (id\_departamento, nome\_departamento, localizacao) VALUES  (1, 'Departamento de Ciência da Computação', 'Campus Universidade Federal de Juiz de Fora, Via Local, 4569, Juiz de Fora - MG');  INSERT INTO departamento (id\_departamento, nome\_departamento, localizacao) VALUES  (2, 'Departamento De Jornalismo', 'Campus Universidade Federal de Juiz de Fora, Via Local, 7452, Juiz de Fora - MG');  #INSERÇÃO DE PROFESSORES  INSERT INTO professor (id\_professor, nome\_professor, siape, cpf, id\_departamento) VALUES  (1,'João das Abóboras','HAS478XDF','14785464789',1);  INSERT INTO professor (id\_professor, nome\_professor, siape, cpf, id\_departamento) VALUES  (2,'Maria Joaquina','H8FX78XDF','78541265478',1);  INSERT INTO professor (id\_professor, nome\_professor, siape, cpf, id\_departamento) VALUES  (3,'Ester Ferreira','H8FX7981F','65423187495',2);  #INSERÇÃO DE LOCAL  INSERT INTO local (id\_local, localizacao, id\_departamento) VALUES  (1, 'Prédio ICE',1);  INSERT INTO local (id\_local, localizacao, id\_departamento) VALUES  (2,'Prédio ICE',1);  INSERT INTO local (id\_local, localizacao, id\_departamento) VALUES  (3,'Prédio ICH',1);  INSERT INTO sala (id\_local, num\_cadeiras) VALUES  (1,95);  INSERT INTO sala (id\_local, num\_cadeiras) VALUES  (3,63);  INSERT INTO laboratorio (id\_local, num\_maquinas) VALUES  (3,45);  #INSERÇÃO DE DISCIPLINAS  INSERT INTO disciplina (id\_disciplina, nome\_disciplina, ementa, id\_coornedador,id\_departamento) VALUES  (1,'Estrutura de Dados I','Listas ligadas: listas simples, duplas, circulares, ortogonais e matrizes.  Alocação dinâmica de memória.  Pilhas e filas.  Árvores: implementação, algoritmos de busca, inserção e remoção.  Árvores binárias de busca, árvores balanceadas: AVL, rubro-negras, B-árvores.  Representação de conjuntos.  Estruturas abstratas de dados, encapsulamento.  Exemplos de aplicações de estruturas de dados.',1,1);  INSERT INTO disciplina (id\_disciplina, nome\_disciplina, ementa, id\_coornedador,id\_departamento) VALUES  (2,'Teoria dos Grafos','Conceitos Básicos  Grafos Eulerianos e Hamiltonianos  Árvores  Grafos Bipartidos  Coloração de Vértices  Emparelhamentos  Coloração de Arestas  Grafos Planares  Digrafos',2,1);  INSERT INTO disciplina (id\_disciplina, nome\_disciplina, ementa, id\_coornedador,id\_departamento) VALUES  (3,'Introdução ao Jornalismo','A função social do jornalismo. A caracterização, a captação e a redação de notícias. A estrutura da notícia nos diversos meios impressos e audiovisuais. A organização e o funcionamento dos jornais e dos departamentos de jornalismo de emissoras de rádio e de televisão. As assessorias de imprensa. As agências de notícias. História do jornalismo no Brasil.',3,2);  #INSERÇÃO DE TURMAS  INSERT INTO turma (letra, observacao, numero\_de\_vagas, id\_semestre, id\_disciplina, id\_professor) VALUES  ('A','Matriculas encerradas!',45,1,1,1);  INSERT INTO turma (letra, observacao, numero\_de\_vagas, id\_semestre, id\_disciplina, id\_professor) VALUES  ('B','Matriculas encerradas!',45,1,1,1);  INSERT INTO turma (letra, observacao, numero\_de\_vagas, id\_semestre, id\_disciplina, id\_professor) VALUES  ('A','Matriculas encerradas!',60,1,2,2);  INSERT INTO turma (letra, observacao, numero\_de\_vagas, id\_semestre, id\_disciplina, id\_professor) VALUES  ('A','Matriculas encerradas!',35,2,3,3);  #INSERÇÃO DE PROJETO DE EXTENSÃO  INSERT INTO projeto\_extensao (id\_projeto, nome\_projeto, descricao, data\_termino, id\_departamento) VALUES  (1,'Farmacia Solidaria','Desenvolvimento de aplicação para coleta de medicamentos.','2018-12-01',1);  #INSERÇÃO DE ALUNO PARTICIPA PROJETO  INSERT INTO aluno\_participa\_projeto (id\_aluno, id\_projeto) VALUES  (1,1);  #INSERÇÃO DE HORARIOS  INSERT INTO horario (id\_horario, hora\_inicial, hora\_final) VALUES  (1,'2008-01-01 08:00:00','2008-01-01 09:00:00');  INSERT INTO horario (id\_horario, hora\_inicial, hora\_final) VALUES  (2,'2008-01-01 09:00:00','2008-01-01 10:00:00');  INSERT INTO horario (id\_horario, hora\_inicial, hora\_final) VALUES  (3,'2008-01-01 10:00:00','2008-01-01 11:00:00');  INSERT INTO horario (id\_horario, hora\_inicial, hora\_final) VALUES  (4,'2008-01-01 11:00:00','2008-01-01 12:00:00');  INSERT INTO horario (id\_horario, hora\_inicial, hora\_final) VALUES  (5,'2008-01-01 12:00:00','2008-01-01 13:00:00');  #INSERÇÃO DE RESERVAS  INSERT INTO reserva (id\_reserva, dia, id\_professor, id\_local, data\_reserva) VALUES  (1,'2018-01-05',1,1,'2018-10-21');  INSERT INTO reserva\_p\_disciplina (id\_reserva, id\_turma, nome\_disciplina) VALUES  (1,1,'Estrutura de Dados I');  INSERT INTO reserva\_tem\_horarios (id\_reserva, id\_horario) VALUES  (1,1);  INSERT INTO reserva\_tem\_horarios (id\_reserva, id\_horario) VALUES  (1,2);  #INSERÇÃO ALUNO CURSA TURMA  INSERT INTO aluno\_cursa\_turma (id\_aluno, id\_turma) VALUES  (1,1);  INSERT INTO aluno\_cursa\_turma (id\_aluno, id\_turma) VALUES  (1,2);  INSERT INTO aluno\_cursa\_turma (id\_aluno, id\_turma) VALUES  (2,3);  INSERT INTO aluno\_cursa\_turma (id\_aluno, id\_turma) VALUES  (3,2);  INSERT INTO aluno\_cursa\_turma (id\_aluno, id\_turma) VALUES  (3,4); |

**(INSERTs para carga dos dados. Inserir uma quantidade razoável de dados para que façam sentido as consultas que serão feitas posteriormente em SQL)**   
  
**6. CONSULTAS EM SQL**

Segue abaixo as consultas SQL:

|  |
| --- |
| **Consulta 1: “Listar todas as reservas com sua data e professor”** |
| SELECT  p.nome\_professor,  r.data\_reserva  FROM professor p  JOIN  reserva r  ON  p.id\_professor = r.id\_professor; |

|  |
| --- |
| **Consulta 2: “Listar todos os professores, mostrando seu nome, SIAPE, CPF e seu departamento”** |
| SELECT  p.nome\_professor,  p.siape,  p.cpf,  d.nome\_departamento  FROM professor p  JOIN  departamento d  ON  p.id\_departamento = d.id\_departamento; |

|  |
| --- |
| **Consulta 3: “Listar todas as disciplinas, mostrando seu nome, coordenador e departamento”** |
| SELECT  dis.nome\_disciplina,  prof.nome\_professor,  dep.nome\_departamento  FROM departamento dep  JOIN  disciplina dis  ON  dep.id\_departamento = dis.id\_departamento  JOIN  professor prof  ON  prof.id\_professor = dis.id\_coornedador; |

|  |
| --- |
| **Consulta 4: “Listar todas as turmas, mostrando sua Letra, disciplina, observação, número de vagas, semestre e professor”** |
| SELECT  t.letra,  d.nome\_disciplina,  t.observacao,  t.numero\_de\_vagas,  s.nome\_semestre,  p.nome\_professor  FROM  turma t  JOIN  disciplina d  ON  t.id\_disciplina = d.id\_disciplina  JOIN  professor p  ON  p.id\_professor = t.id\_professor  JOIN  semestre s  ON  s.id\_semestre = t.id\_semestre; |

|  |
| --- |
| **Consulta 5: “Listar todas as salas, mostrando seu id e número de cadeiras”** |
| SELECT  l.localizacao,  s.num\_cadeiras  FROM  local l  JOIN  sala s  ON  l.id\_local = s.id\_local; |

|  |
| --- |
| **Consulta 6: “Listar todos os Laboratórios, mostrando seu id e o número de maquinas”** |
| SELECT  l.localizacao,  lab.num\_maquinas  FROM  local l  JOIN  laboratorio lab  ON  l.id\_local = lab.id\_local; |

|  |
| --- |
| **Consulta 7: “Listar todos os projetos de extensão, mostrando nome do projeto, descrição, data de termino, departamento e orientador”** |
| SELECT  proj.nome\_projeto,  proj.descricao,  proj.data\_termino,  dep.nome\_departamento  FROM  projeto\_extensao proj  JOIN  departamento dep  ON  dep.id\_departamento = proj.id\_departamento; |

|  |
| --- |
| **Consulta 8: “Listar todas as turmas que um determinado aluno cursa, mostrando sua letra, disciplina, semestre letivo e professor”** |
| SELECT  t.letra,  d.nome\_disciplina,  s.nome\_semestre,  p.nome\_professor  FROM  aluno a  JOIN  aluno\_cursa\_turma act  ON  a.id\_aluno = act.id\_aluno  JOIN  turma t  ON  t.id\_turma = act.id\_turma  JOIN  professor p  ON  p.id\_professor = t.id\_professor  JOIN  disciplina d  ON  d.id\_disciplina = t.id\_disciplina  JOIN  semestre s  ON  s.id\_semestre = t.id\_semestre  WHERE  a.id\_aluno = ‘X’; |

|  |
| --- |
| **Consulta 9: “Listar todos os projetos de extensao que um determinado aluno participa, mostrando nome do projeto, descricao, data de termino, orientador e departamento”** |
| SELECT  pe.nome\_projeto,  pe.descricao,  pe.data\_termino,  d.nome\_departamento  FROM  aluno a  JOIN  aluno\_participa\_projeto app  ON  a.id\_aluno = app.id\_aluno  JOIN  projeto\_extensao pe  ON  pe.id\_projeto = pe.id\_projeto  JOIN  departamento d  ON pe.id\_departamento = d.id\_departamento  WHERE a.id\_aluno = ‘X’; |

|  |
| --- |
| **Consulta 10: “Listar todas as turmas que um determinado professor ministra, mostrando sua letra, disciplina, semestre letivo”** |
| SELECT  t.letra,  d.nome\_disciplina,  s.nome\_semestre  FROM  turma t  JOIN  semestre s  ON  t.id\_semestre = s.id\_semestre  JOIN  disciplina d  ON  d.id\_disciplina = t.id\_disciplina  JOIN  professor p  ON  t.id\_professor = p.id\_professor  WHERE id\_professor = 'X'; |

|  |
| --- |
| **Consulta 12: “Listar todos os horários de uma determinada reserva, mostrando o horário inicial e horário final”** |
| SELECT  h.hora\_inicial,  h.hora\_final  FROM  reserva r  JOIN  reserva\_tem\_horarios rth  ON  rth.id\_reserva = r.id\_reserva  JOIN  horario h  ON  h.id\_horario = rth.id\_horario  WHERE  r.id\_reserva = 'X'; |

|  |
| --- |
| **Consulta 13: “Listar todos os horários cadastrados, mostrando horário inicial e horário final”** |
| SELECT  h.hora\_inicial,  h.hora\_final  FROM  horario h; |

|  |
| --- |
| **Consulta 14: “Listar todos os alunos, mostrando seu nome, matricula, CPF e curso”** |
| SELECT  matricula,  nome\_aluno,  cpf,  curso  FROM  aluno; |

|  |
| --- |
| **Consulta 15: “Listar todos os alunos de uma determinada turma, mostrando seu nome, matricula, CPF e curso”** |
| SELECT  a.matricula,  a.nome\_aluno,  a.cpf,  a.curso  FROM  turma t  JOIN  aluno\_cursa\_turma act  ON  t.id\_turma = act.id\_turma  JOIN  aluno a  ON  a.id\_aluno = act.id\_aluno  WHERE  t.id\_turma = 'X'; |

**7 MODELO FÍSICO (Parte 2)**

**7.1 Gatilhos e Funções**

|  |
| --- |
| **Gatilho 1: “Atualizar coluna vagas\_ocupadas quando inserir na tabela aluno\_cursa\_turma** |
| CREATE OR REPLACE FUNCTION inserirAlunoNaTurma() RETURNS trigger AS  '  Begin  UPDATE turma SET vagas\_ocupadas = vagas\_ocupadas+1 WHERE id\_turma = new.id\_turma;  return new;  end;  '  LANGUAGE plpgsql;  CREATE TRIGGER inserirAlunoNaTurma  AFTER INSERT ON aluno\_cursa\_turma  FOR EACH ROW  EXECUTE PROCEDURE inserirAlunoNaTurma();  -- Para testar o Gatilho  INSERT INTO (id\_aluno, id\_turma) VALUES aluno\_cursa\_turma(1,1);  INSERT INTO (id\_aluno, id\_turma) VALUES aluno\_cursa\_turma(1,2); |

|  |
| --- |
| **Gatilho 2: “Atualizar coluna vagas\_ocupadas quando remover na tabela aluno\_cursa\_turma** |
| CREATE OR REPLACE FUNCTION removerAlunoNaTurma() RETURNS trigger AS  '  Begin  UPDATE turma SET vagas\_ocupadas = vagas\_ocupadas-1 WHERE id\_turma = old.id\_turma;  return old;  end;  '  LANGUAGE plpgsql;  CREATE TRIGGER removerAlunoNaTurma  BEFORE DELETE ON aluno\_cursa\_turma  FOR EACH ROW  EXECUTE PROCEDURE removerAlunoNaTurma();  --Para testar o Gatilho:  DELETE FROM aluno\_cursa\_turma WHERE id\_aluno = 1; |

**7.3 Visões**

|  |
| --- |
| **Visão 1: “Professor e Aluno/Visualizar todas as turmas com o nome da disciplina, semestre e professor”** |
| Objetivo: Essa visão tem como objetivo listar todas as turmas, que será usada no plano departamental. É uma das consultas mais “pesadas” que a aplicação tem.  Visão:  CREATE OR REPLACE VIEW listarTurmas AS (SELECT  t.letra,  d.nome\_disciplina,  t.observacao,  t.numero\_de\_vagas,  s.nome\_semestre,  p.nome\_professor  FROM  turma t  JOIN  disciplina d  ON  t.id\_disciplina = d.id\_disciplina  JOIN  professor p  ON  p.id\_professor = t.id\_professor  JOIN  semestre s  ON  s.id\_semestre = t.id\_semestre); |

|  |
| --- |
| **Visão 2: “Professor e Aluno/Visualizar todas as turmas com o nome da disciplina, semestre e professor das reservas com os alunos inscritos nela”** |
| Objetivo: Essa visão tem como objetivo listar todas as turmas com os alunos que será usada para saber quais alunos está em quais reservas. Também é uma consulta “pesada”.  Visão:  CREATE OR REPLACE VIEW reserva\_com\_todos\_dados AS (SELECT  t.id\_turma,  t.letra,  d.nome\_disciplina,  t.observacao,  t.numero\_de\_vagas,  s.nome\_semestre,  p.nome\_professor,  r.id\_reserva,  a.id\_aluno,  a.nome\_aluno,  a.matricula  FROM  turma t  JOIN  disciplina d  ON  t.id\_disciplina = d.id\_disciplina  JOIN  professor p  ON  p.id\_professor = t.id\_professor  JOIN  semestre s  ON  s.id\_semestre = t.id\_semestre  JOIN  aluno\_cursa\_turma act  ON  act.id\_turma = t.id\_turma  JOIN  aluno a  ON  a.id\_aluno = act.id\_aluno  JOIN  reserva\_p\_disciplina rpd  ON  rpd.id\_turma = t.id\_turma  JOIN  reserva r  ON  r.id\_reserva = rpd.id\_reserva); |

**8 OTIMIZAÇÃO DO BANCO**

**8.1 Índices**

|  |
| --- |
| **Índice 1: Hash na coluna matricula da tabela aluno** |
| **Criando índice:** CREATE INDEX aluno\_matricula ON aluno USING hash (matricula);  **Motivo da criação:** Pelo fato do número de matrícula ser comumente usados como parâmetro para busca, usar uma hash auxiliaria na busca. |

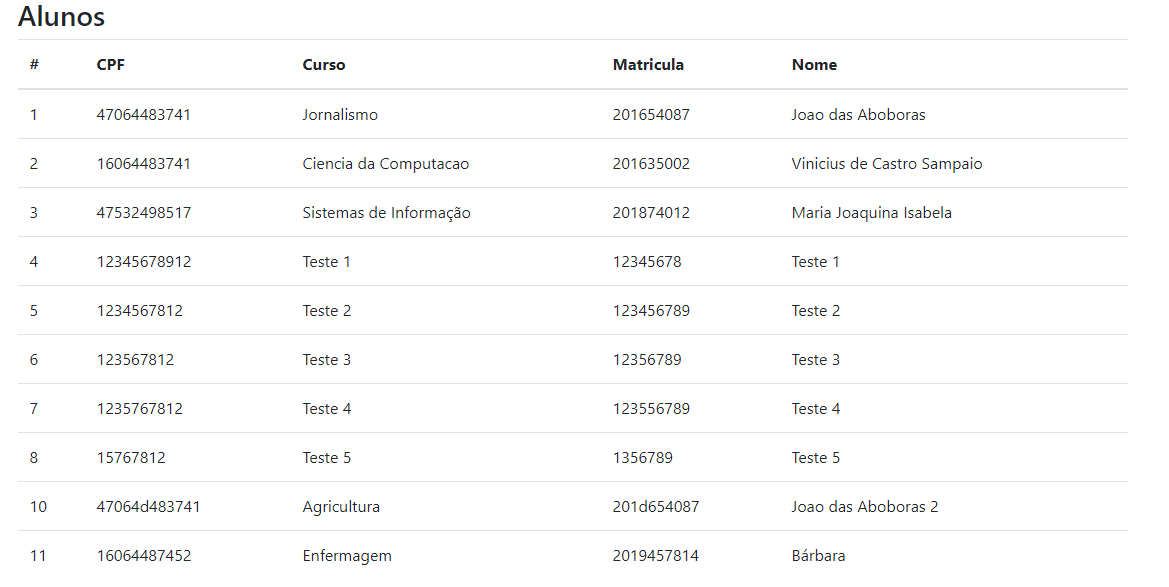
**8.2 Verificação dos índices**

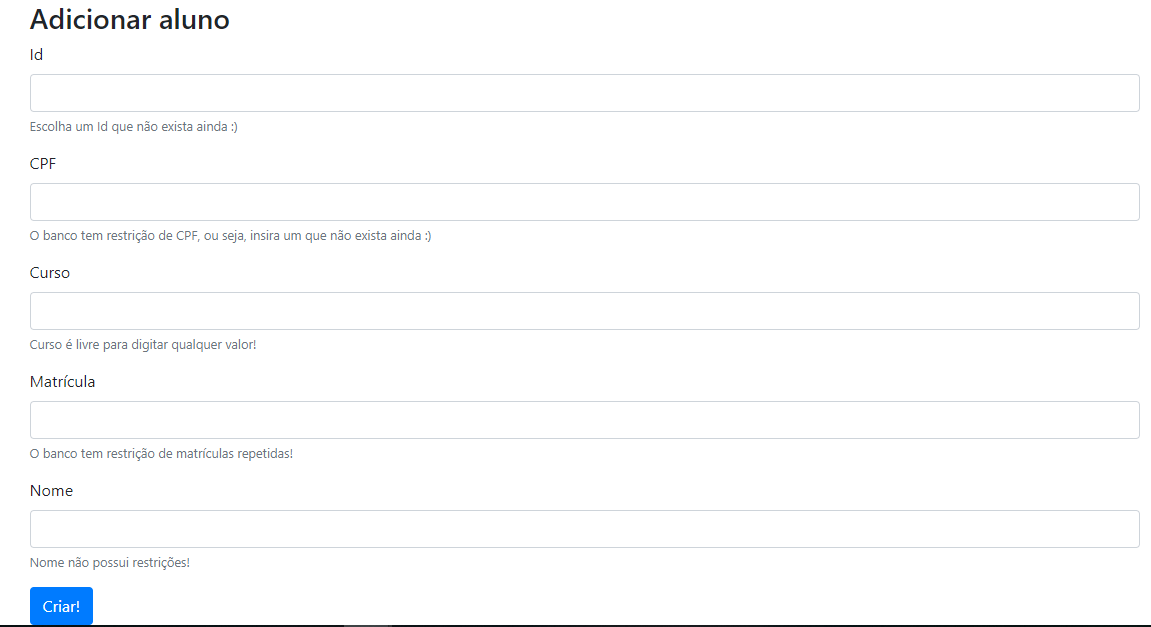
|  |
| --- |
| **Teste 1: Selecionando todas as colunas da tabela aluno buscando por um número de matrícula.** |
| **Select sem Index com WHERE passando matrícula:** 1.761  **Select com Index com WHERE passando matrícula:** 1.247  select \* from aluno where matricula='123556789';  Obs: Pelo fato do banco possuir poucos dados, a diferença não pode ser notada. |

|  |
| --- |
| **Teste 2: Buscando todas as turmas que um aluno está matrículado.** |
| **Select sem Index com WHERE passando matrícula:** 3.166  **Select com Index com WHERE passando matrícula:** 2.870  select \* from aluno a  join aluno\_cursa\_turma act  on act.id\_aluno = a.id\_aluno  join turma t  on t.id\_turma = act.id\_turma  where a.matricula='201635002';  Obs: Pelo fato do banco possuir poucos dados, a diferença não pode ser notada. |

**9. TELAS DA APLICAÇÃO**

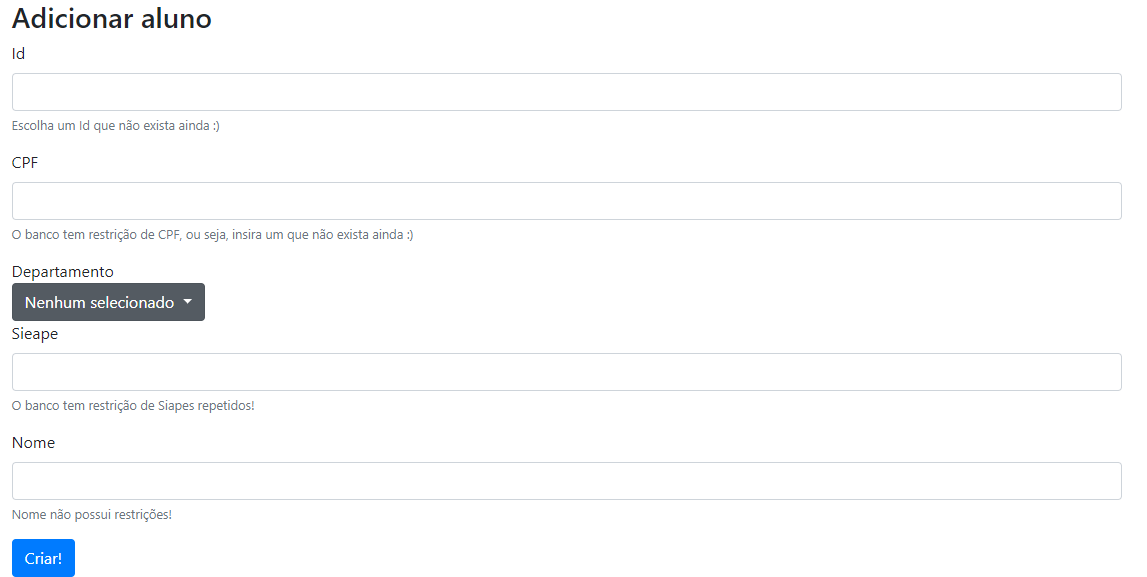
**Tela 1: Visualizar e inserir alunos**





**Tela 2: Visualizar e inserir professores**





**Tela 3: Visualizar disciplinas e suas turmas:**

