

SISTEMA DE GERENCIAMENTO DE ALUNOS (SGA) - DCE

JARDIM, Gabriel Pereira¹

gabriel_pjardim@hotmail.com

MAGALHÃES, Felipe Boeira¹

f.maagalhaes@outlook.com

ROMANI, Samuel Etchichury¹

samueel_romani@hotmail.com

STORNIOLO, Thiago Vaz¹

vaz_vs@hotmail.com

BASTOS, Rafael²

rafaelrodriguesbastos@gmail.com

CAMARGO, Salvador²

salvador@sigasistemas.com.br

FERNANDES, Acauan²

acauanfernandes@gmail.com

PEREIRA, Marcus Vinicius³

marcus_v_sp@hotmail.com

¹ Discentes do curso de Análise e Desenvolvimento de Sistemas, 2016/1 – Faculdade IDEAU – Bagé/RS.

² Docentes do curso de Análise e Desenvolvimento de Sistemas, 2016/1 – Faculdade IDEAU – Bagé/RS.

³ Coordenador do curso de Análise e Desenvolvimento de Sistemas – Faculdade IDEAU – Bagé/RS.

RESUMO: Este artigo aborda sobre o projeto do sistema de gerenciamento de alunos que foi desenvolvido para suprir as necessidades que a faculdade IDEAU e o DCE tem de uma ferramenta onde possa haver benefícios para os alunos e realizar cadastramento da semana acadêmica. Com o auxílio de várias ferramentas foi possível desenvolver um planejamento e uma estrutura onde telas do projeto de software foram desenvolvidos, e um banco de dados que foi criado e testado utilizando o PostgreSQL. Com o desenvolvimento desse projeto, o usuário poderá observar como o software será desenvolvido, podendo assim realizar mudanças onde há necessidade.

Palavras-chave: Cadastramento; Planejamento; Banco de Dados.

ABSTRACT: This article approaches about the project of students management system that was developed to meet the needs of the college IDEAU and have of a tool where may be benefits to the students and conduct registration of the academic week. With the help of various tools was possible to develop a plan and a structure where software project screens were developed, and a database that was created and tested using PostgreSQL. With the development of this project, the user can see how the software will be developed, being able to make changes where there is need.

Keywords: Registration; Plan; Database.

1 INTRODUÇÃO

A necessidade de ter um sistema informatizado para o gerenciamento de seus dados é essencial e indispensável para qualquer empresa, seja ela de pequeno ou grande porte. As grandes organizações estão sempre mantendo maneiras de manter seus dados atualizados, seguros e de uma maneira fácil para ser acessado pelos seus funcionários. Esses programas possuem diversas funcionalidades variando de acordo com a empresa. Muitas delas utilizam programas para realizar essa tarefa. Visando tal importância de manter dados e registro, será realizado um projeto de sistema de gerenciamento e controle dos alunos da Faculdade Instituto de Desenvolvimento Educacional do alto Uruguai-IDEAU, campus Bagé.

O sistema tem como uma de suas funcionalidades manter o registro de todos os alunos que participarem de eventos realizados pelo Diretorio Central de Estudantes (DCE) da Faculdade IDEAU, onde será possível realizar relatórios de cada curso, turma etc. O sistema tem como objetivo principal não só manter registro dos alunos da faculdade, mas como dispor de uma ferramenta onde o DCE, diretório central dos estudantes poderá manter um controle de alunos que optarem por se associar ao DCE. Os alunos que optarem por se associar terão de pagar uma taxa mensal, onde obterão uma carteira de sócio e aluno da Faculdade IDEAU que gerará descontos em empresas que parceiras do DCE. Devido a essa falta de um sistema informatizado, tanto a faculdade como o DCE necessita de um controle para realizar um atendimento melhor aos alunos, e vantagens para os sócios do DCE. O projeto de sistema deverá conter com as seguintes opções:

- Registro dos alunos da faculdade IDEAU;
- Registro dos alunos sócios do DCE;
- Controle de presença na semana acadêmica;
- Controle de mensalidade dos sócios;
- Relatórios afins;

Para realizar o desenvolvimento desse projeto de software, temos como base as dificuldades em que a faculdade e o DCE, e na criação de um novo sistema para sócios, onde terá uma grande evolução para a faculdade IDEAU.

O projeto de software tem como base utilizar os registros dos alunos da faculdade para servir como base do programa. Os alunos que quiserem ser sócios do DCE terão que realizar um pagamento mensal, a ser definido pelo DCE, onde o aluno sócio terá uma carteirinha onde poderá utilizar para ter vários benefícios na cidade de Bagé.

2 DESENVOLVIMENTO

Referencial Teórico

De acordo com Koscianski e Soares (2007): “Qualidade de software tem o proposito de estabelecer sistemas que controlem tudo o que ocorre durante o ciclo da vida, garantindo que os objetivos planejados no inicio do projeto sejam cumpridos e que o programa fabricado fará o que se espera dele.”.

Segundo Macedo (2012): “Sistemas de informações podem ser classificados da maneira em que a informação ou dado for processado. Para realizar essa classificação utiliza-se geralmente uma pirâmide empresarial, onde é dividida em vários níveis. Essa pirâmide é dividida em Estratégica, Tatica e operacional.”.

No entendimento de Elmasri e Navathe (2006) a integridade do banco refere-se à exigência que a informação esteja assegurada de modificações impróprias, incluindo a criação, a inclusão, a alteração e a exclusão. Caso ocorra a perda da integridade dos dados e não for corrigida, podem causar imprecisão, acarretando em tomadas de decisões equivocadas.

Na escolha de quais riscos devem ser gerenciados modelos são bastante úteis, porém sua aplicação deve ser muito criteriosa. Nas organizações, o normal, é cada pessoa ou departamento tentar encobrir suas deficiências e camuflar os verdadeiros riscos, o que dificulta o sucesso da aplicação do modelo. Cabe ao time de gerenciamento trabalhar as informações, eliminar as equivocadas e obter os resultados desejados (PRITCHARD, 2001).

Para manter as informações dos usuários no banco de dados segundo Elmasri e Navathe (2006, p. 529): Não há dificuldades em manter informações dos usuários do banco de dados e de suas contas e senhas por meio da criação de uma tabela ou um arquivo cifrado com os dois campos NumeroDeConta e Senha(...) Sempre que uma nova conta for criada, um novo

registro é incluído na tabela. Quando uma conta for cancelada, o registro correspondente deve ser excluído da tabela.

Segundo Pritchard (2001): “Não é necessário gerenciar todo risco identificado, mas deve ser analisada cuidadosamente cada decisão de qual risco gerenciar e como agir.”.

Materiais

DBDesigner4: É uma ferramenta CASE (Computer-Aided Software Engineering) desenvolvida pela empresa Fabulous Force Database Tools. É uma ferramenta livre e é utilizada para modelagem de dados visual. Com ele é possível modelar tabelas de forma gráfica, relacionamentos, dados da tabela e muito mais. Utilizando este poderoso ambiente temos a geração da modelagem, do projeto, da implementação e da manutenção integradas em apenas um ambiente. É compatível com vários bancos de dados como: MySQL, Oracle, MSSQL e PostgreSQL, entre outros. Esta CASE será utilizada para desenvolver o modelo entidade e relacionamento do banco de dados.

DiagramDesigner: É um software de edição de imagens específico para criar diagramas e organogramas. Oferece uma forma simples para criar qualquer tipo de diagramas e gráficos. Possui gráficos pré-desenhados que incluem todos os tipos de quadrados, círculos, e setas, sendo possível personalizar seus diagramas criados. Utilizada para criação do diagrama de entidade e relacionamento assim como os protótipos de interface do sistema.

PostgreSQL: É um sistema de gerenciamento de banco de dados do tipo objeto-relacional (ORDBMS) com ênfase em extensibilidade e em padrões de conformidade. Como um servidor de banco de dados, sua principal função é armazenar dados de forma segura, apoiando as melhores práticas, permitindo a recuperação dos dados a pedido de outras aplicações de software. Ele pode lidar com cargas de trabalho que vão desde pequenas aplicações single-machine a aplicações de grande porte voltadas para a Internet, onde será utilizada de forma simultânea por vários usuários. O PostgreSQL foi o SGBD utilizado para a criação do banco de dados e armazenamento dos dados.

Métodos

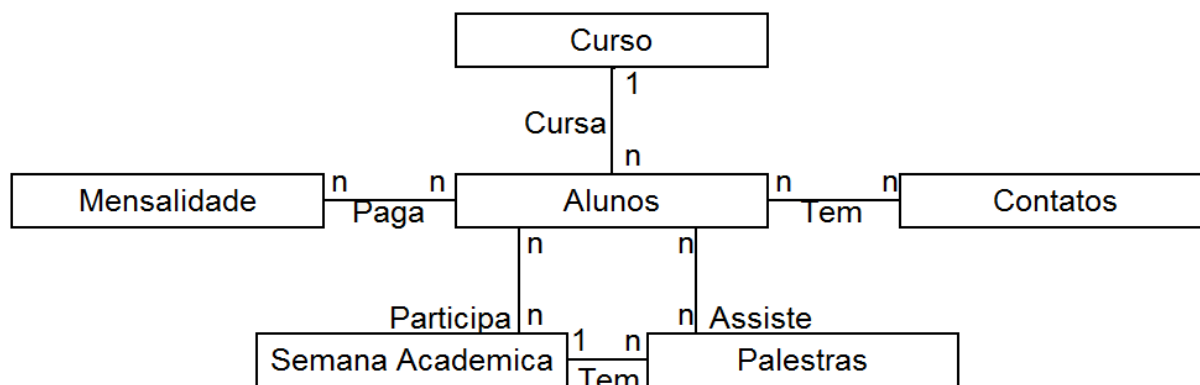
Para o desenvolvimento do banco de dados e realização do projeto do sistema foi iniciado com a elicitação de requisitos que foi elaborada junto a um dos membros do DCE o qual nos informou as necessidades do sistema. Nesta solicitação foi informado que havia a necessidade do cadastro dos alunos que participam da semana acadêmica e dos alunos que se associarem ao DCE, onde necessitaria o curso do aluno, numero para contato, controle da mensalidade e controle da semana acadêmica, com isso foi levantado os seguintes requisitos para o banco de dados:

Requisitos do Sistema (1º reunião)				
Alunos	Contato	Curso	Sem. Acad.	Mensalidade

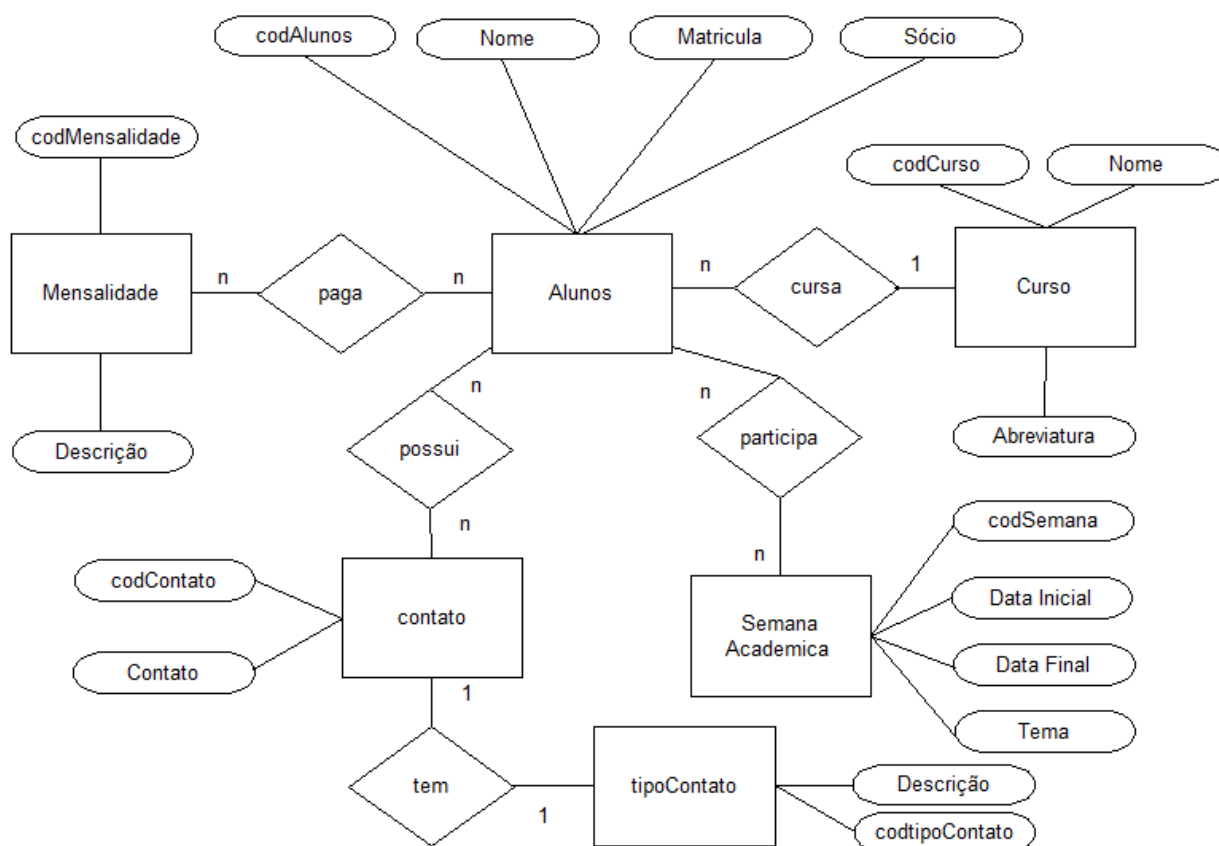
Com outra reunião com o membro do DCE podemos refinar e saber mais da ideia, onde:

Requisitos do Sistema (2º reunião)				
Presença sem. acad.	Matricula	Nome curso	Nome aluno	Telefone aluno
Abreviação do curso	Dia (inicio e fim) sem. acad.	Sem. acad.	Mensalidade para sócios	E-mail aluno
Nome palestra	Nome palestrante	Pag. sem. acadêmica		

DiagramDesigner: Com a elicitação de requisitos pronta iniciamos o modelo conceitual do banco de dados. O modelo conceitual tem um grande nível de abstração, onde podemos envolver o cliente, pois nesta etapa visamos o negocio do cliente e não da tecnologia que será utilizada. No modelo conceitual utilizamos o diagrama de entidade e relacionamento (DER), onde identificamos todas as entidades e os relacionamentos entre elas. Segue abaixo o DER desenvolvimento:

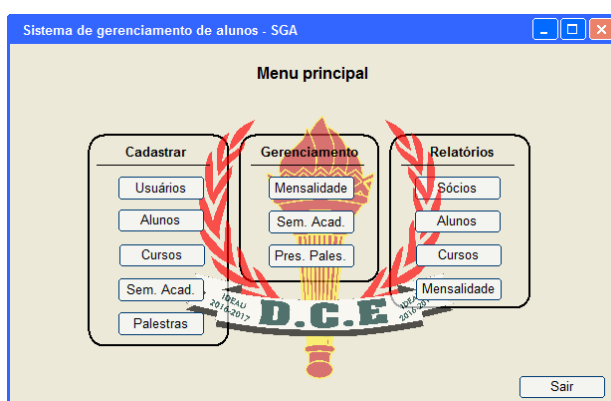


Após o desenvolvimento do DER abstraído desenvolvemos o der com maior numero de informações para auxilio no desenvolvimento do BD.



A ferramenta DiagramDesigner proporciona a prototipação das interfaces.

Tela do Menu Principal



Menu principal do sistema onde temos os módulos divididos na tela por: Cadastrar, Gerenciamento e Relatórios.

- **Cadastrar** tem a possibilidade de cadastrar os usuários, alunos, cursos, semana acadêmica e as palestras.

- No **Gerenciamento** podemos ter controle das mensalidades dos alunos associados, controle do pagamento da semana acadêmica por aluno e o controle de presença dos alunos nas palestras.

- Nos **Relatórios** podemos ter um controle geral dos Alunos associados, alunos, cursos e as mensalidades.

Tela de cadastramento de alunos

Sistema de gerenciamento de alunos - SGA

S. G. A.

Cadastrar Alunos

Nome:

Matricula:

Contato: Tipo contato Contato +

Mensalidade: Mês +

Sócio: ☒

D.C.E.

Na tela de cadastramento de alunos podemos cadastrar o aluno e respectivamente dizer se é sócio e adicionar os meses aos quais já efetuou o pagamento das mensalidades. Nesta tela temos o campo “contato” que via combobox trazemos os tipos salvamos na tabela tipoContato e atribuímos ele ao campo text contato. Quando acionado o botão “+” podemos inserir mais de um contato para o usuário.

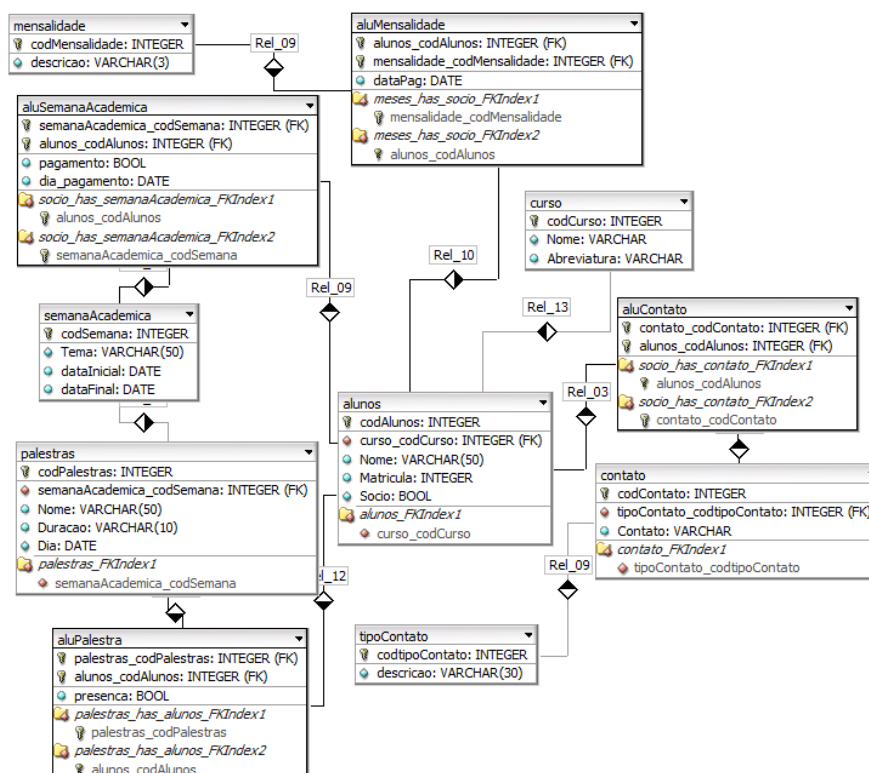
DBDesigner: Com o diagrama entidade e relacionamento (DER) foi iniciado o desenvolvimento do modelo entidade e relacionamento (MER). O modelo entidade e relacionamento é utilizado na engenharia de software, pois ele descreve com detalhamento as entidades e seus atributos, assim como a relação entre elas (entidades).

Entidades: São objetos físicos ou lógicos dependendo da sua existência no mundo real. As entidades físicas são aquelas visíveis no mundo real e as entidades logicas são aquelas que são geradas em decorrência ao relacionamento entre entidades físicas.

Relacionamentos: Após identificar as entidades foram definidos seus relacionamentos, onde foram obtidos relacionamentos N:N, 1:N (N:1) e 1:1. Os relacionamentos N:N são muitos registros de uma entidade que relacionam ou muitos requisitos de outra entidade, sendo assim gerando uma terceira entidade com as respectivas chaves primarias das outras duas entidades. Os relacionamentos 1:N (N:1) ocorre quando um registro de uma entidade

relaciona com muitos registro de outra entidade assim a chave primaria da entidade que 1 registro liga em muitos será referenciada como uma chave estrangeira na entidade que recebe o N registros. Os relacionamentos 1:1 são quando 1 registro de uma entidade relaciona com 1 registro de outra entidade.

Atributos: São as características que descrevem cada entidade como, por exemplo, a entidade “alunos” tem suas características/atributos que são nome e matricula.



Tendo conhecimento dos padrões da IEEE para o desenvolvimento de um software com qualidade desenvolvemos o dicionário de dados.

Tabela Alunos: Utilizada para o cadastramento de todos os alunos que participarem de atividades elaboradas pelo diretório central de estudantes (DCE).

Alunos			
Nome	Tipo	Tamanho	Descrição
codAlunos	integer(PK)		Identificador único do aluno
curso_codCurso	integer(FK)		Identificador do curso que o aluno esta matriculado
Nome	varchar	50	Nome do aluno cadastrado
Matricula	integer		Matricula que identifica o aluno dentro

			da faculdade
Socio	bool		Campo que identifica se o aluno é sócio da DCE ou não

Tabela curso: Utilizada para o cadastramento de todos os cursos oferecidos pela faculdade IDEAU – Campus Bagé.

curso			
Nome	Tipo	Tamanho	Descrição
codCurso	Integer(PK)		Identificador único do curso
Nome	varchar	50	Nome do curso cadastrado
Abreviatura	varchar	4	Abreviatura do curso cadastrado

Tabela mensalidade: Utilizada para o cadastramento dos 12 meses do ano.

mensalidade			
Nome	Tipo	Tamanho	Descrição
codMensalidade	integer(PK)		Identificador único do mês
descricao	varchar	3	Nome do mês com 3 caracteres: - Jan - Fev - Mar - Abr - Jul - Jun - Ago - Set - Out - Nov - Dez

Tabela semanaAcademica: Utilizada para o cadastramento das semanas acadêmica da faculdade IDEAU – Campus Bagé

semanaAcademica			
Nome	Tipo	Tamanho	Descrição

codSemana	integer(PK)		Identificador único da semana acadêmica.
Tema	varchar	50	Nome do tema central da semana acadêmica.
dataInicial	date		Data ao qual começa a semana acadêmica.
dataFinal	date		Data de termino da semana acadêmica.

Tabela palestras: Utilizada para o cadastramento das palestras.

palestras			
Nome	Tipo	Tamanho	Descrição
codPalestras	integer(PK)		Identificador único da palestra
semanaAcademica_codSemana	Integer(FK)		Identificador da semana acadêmica.
Nome	varchar	50	Nome da palestra cadastrada
Duracao	varchar	10	Tempo de duração da palestra.
Dia	date		Dia da palestra cadastrada

Tabela contato: Utilizada para cadastrar o contato dos alunos.

contato			
Nome	Tipo	Tamanho	Descrição
codContato	integer(PK)		Identificador único do contato de um referente aluno.
tipoContato_codtipoContato	integer(FK)	50	Nome do tema central da semana acadêmica.
Contato	varchar	30	O contato cadastrado.

Tabela tipoContato: Utilizada para o cadastramento dos tipo de contatos.

tipoContato			
Nome	Tipo	Tamanho	Descrição
codtipoContato	integer(PK)		Identificador único do tipo de contato.
descricao	varchar	10	O tipo de contato.

			- Email - Celular - Residencial
--	--	--	---------------------------------------

Temos a tabelas criadas pela cardinalidade de N:N como as:

Tabela aluMensalidade: Utilizada para referenciar que a Mensalidade com os alunos.

aluMensalidade			
Nome	Tipo	Tamanho	Descrição
alunos_codAlunos	integer(FK)		Identificador do aluno com a mensalidade
mensalidade_codMensalidade	Integer(FK)		Identificador da mensalidade com o aluno
dataPag	date		Campo para cadastro do dia do pagamento

Tabela aluContato: Utilizada para referenciar o aluno com seus contatos.

aluContato			
Nome	Tipo	Tamanho	Descrição
contato_codContato	integer(FK)		Identificador do contato com o aluno
alunos_codAlunos	Integer(FK)		Identificador do aluno com contato

Tabela aluSemanaAcademica: Utilizada para referenciar o aluno na semana acadêmica.

aluSemanaAcademica			
Nome	Tipo	Tamanho	Descrição
semanaAcademica_codSemana	integer(FK)		Identificador da semana acadêmica com o aluno
alunos_codAlunos	Integer(FK)		Identificador do aluno com semana acadêmica.
pagamento	bool		Identificador de pagamento
dia_pagamento	date		Dia ao qual foi efetuado o pagamento

Tabela aluPalestra: Utilizada para referenciar as presenças do aluno nas palestras.

aluPalestras			
Nome	Tipo	Tamanho	Descrição
palestras_codPalestras	integer(FK)		Identificador da palestra com o aluno
alunos_codAlunos	Integer(FK)		Identificador do aluno com a palestra.
presenca	bool		Identificador de presença do aluno.

A ferramenta case **DiagramDesigner** também foi utilizada para o desenvolvimento do protótipo de interface.

PostgreSQL: Após a modelagem do banco concluída geramos o script sql pelo DBDesigner o qual nos gerou o código:

```
CREATE TABLE curso (
  codCurso INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Nome VARCHAR NULL,
  Abreviatura VARCHAR NULL,
  PRIMARY KEY(codCurso));

CREATE TABLE mensalidade (
  codMensalidade INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descricao VARCHAR(3) NULL,
  PRIMARY KEY(codMensalidade));

CREATE TABLE tipoContato (
  codtipoContato INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descricao VARCHAR(30) NULL,
  PRIMARY KEY(codtipoContato));

CREATE TABLE semanaAcademica (
  codSemana INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Tema VARCHAR(50) NULL,
  dataInicial DATE NULL,
  dataFinal DATE NULL,
  PRIMARY KEY(codSemana));

CREATE TABLE palestras (
  codPalestras INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  semanaAcademica_codSemana INTEGER UNSIGNED NOT NULL,
  Nome VARCHAR(50) NULL,
  Duracao VARCHAR(10) NULL,
  Dia DATE NULL,
  PRIMARY KEY(codPalestras),
  INDEX palestras_FKIndex1(semanaAcademica_codSemana),
  FOREIGN KEY(semanaAcademica_codSemana)
  REFERENCES semanaAcademica(codSemana)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

CREATE TABLE contato (
```

```
codContato INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
tipoContato_codtipoContato INTEGER UNSIGNED NOT NULL,  
Contato VARCHAR NULL,  
PRIMARY KEY(codContato),  
INDEX contato_FKIndex1(tipoContato_codtipoContato),  
FOREIGN KEY(tipoContato_codtipoContato)  
REFERENCES tipoContato(codtipoContato)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

```
CREATE TABLE alunos (  
codAlunos INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
curso_codCurso INTEGER UNSIGNED NOT NULL,  
Nome VARCHAR(50) NULL,  
Matricula INTEGER UNSIGNED NULL,  
Socio BOOL NULL,  
PRIMARY KEY(codAlunos),  
INDEX alunos_FKIndex1(curso_codCurso),  
FOREIGN KEY(curso_codCurso)  
REFERENCES curso(codCurso)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

```
CREATE TABLE aluMensalidade (  
alunos_codAlunos INTEGER UNSIGNED NOT NULL,  
mensalidade_codMensalidade INTEGER UNSIGNED NOT NULL,  
dataPag DATE NULL,  
PRIMARY KEY(alunos_codAlunos, mensalidade_codMensalidade),  
INDEX meses_has_socio_FKIndex1(mensalidade_codMensalidade),  
INDEX meses_has_socio_FKIndex2(alunos_codAlunos),  
FOREIGN KEY(mensalidade_codMensalidade)  
REFERENCES mensalidade(codMensalidade)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
FOREIGN KEY(alunos_codAlunos)  
REFERENCES alunos(codAlunos)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

```
CREATE TABLE aluContato (  
contato_codContato INTEGER UNSIGNED NOT NULL,  
alunos_codAlunos INTEGER UNSIGNED NOT NULL,  
PRIMARY KEY(contato_codContato, alunos_codAlunos),  
INDEX socio_has_contato_FKIndex1(alunos_codAlunos),  
INDEX socio_has_contato_FKIndex2(contato_codContato),  
FOREIGN KEY(alunos_codAlunos)  
REFERENCES alunos(codAlunos)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
FOREIGN KEY(contato_codContato)  
REFERENCES contato(codContato)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

```
CREATE TABLE aluPalestra (  
palestras_codPalestras INTEGER UNSIGNED NOT NULL,  
alunos_codAlunos INTEGER UNSIGNED NOT NULL,  
presenca BOOL NULL,
```

```
PRIMARY KEY(palestras_codPalestras, alunos_codAlunos),
INDEX palestras_has_alunos_FKIndex1(palestras_codPalestras),
INDEX palestras_has_alunos_FKIndex2(alunos_codAlunos),
FOREIGN KEY(palestras_codPalestras)
REFERENCES palestras(codPalestras)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(alunos_codAlunos)
REFERENCES alunos(codAlunos)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

CREATE TABLE aluSemanaAcademica (
semanaAcademica_codSemana INTEGER UNSIGNED NOT NULL,
alunos_codAlunos INTEGER UNSIGNED NOT NULL,
pagamento BOOL NULL,
dia_pagamento DATE NULL,
PRIMARY KEY(semanaAcademica_codSemana, alunos_codAlunos),
INDEX socio_has_semanaAcademica_FKIndex1(alunos_codAlunos),
INDEX socio_has_semanaAcademica_FKIndex2(semanaAcademica_codSemana),
FOREIGN KEY(alunos_codAlunos)
REFERENCES alunos(codAlunos)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(semanaAcademica_codSemana)
REFERENCES semanaAcademica(codSemana)
ON DELETE NO ACTION
ON UPDATE NO ACTION);
```

Após edições em determinados pontos do código a execução dos comandos SQL geraram o banco de dados na PostgreSQL.

4 CONCLUSÃO

Foi abordado sobre o gerenciamento informatizado do controle dos alunos participantes dos eventos propostos pelo DCE ou os alunos que se associarem ao mesmo, onde foi concluído que é de suma importância à informatização para a eficácia e eficiência do controle do DCE. Todos os objetivos propostos foram cumpridos uma vez que a implementação será futuramente feita. Com este trabalho foi possível aperfeiçoar nossas técnicas e ampliar nosso conhecimento sobre as regras do negocio e como modela-las. Foram realizadas reuniões tanto para o planejamento como para o levantamento dos requisitos necessários para o sistema, onde todo o trabalho, pesquisa e estudo nos permitiu projetar o sistema de gerenciamento de alunos – SGA.

REFERÊNCIAS

ELMASRI, R. e NAVATHE, S. B.: **Sistemas de banco de dados**, 4 ed, São Paulo, Pearson Addison Wesley, 2006.

KOSCIANSKI, A. e SOARES, M. S.: **Qualidade de Software**, Novatec, 2007.

MACÊDO, D.: **Sistemas de Informação Operacional, Tático e Estratégico**, 2012.
Disponível em: <<http://www.diegomacedo.com.br/sistemas-de-informacao-operacional-tatico-e-estrategico/>>. Acesso em: 02 Jun, 2016

PEITCHARD, C. L.: **Risk Management – Concepts and Guidance**, ESI International, 2001.