

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS TECNOLOGIA E SAÚDE
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO
BANCO DE DADOS - DEC-7588

Sistema Gerenciador de Centros Acadêmicos

Professor: Alexandre Leopoldo Gonçalves
Autores: Joel João Medeiros Filho - joel.joao.medeiros@grad.ufsc.br, Nayara Mara
Honorato da Silva - nayaramarah@gmail.com.
Araranguá, 17 de Julho de 2022

Sumário

Sumário	2
Descrição do Objetivo Geral do Sistema	3
Descrição Detalhada do Sistema	3
Modelagem Conceitual	4
Modelagem Lógica	5
Script DDL	6
Consultas	7
Consulta 1	7
Consulta 2	7
Lista de Anexos	9

Descrição do Objetivo Geral do Sistema

Elaborar uma aplicação para gerenciamento das atividades cotidianas de um centro acadêmico, registrando seus alunos, seus artefatos, cooperadores e suas informações de forma geral, como por um exemplo a que centro e instituição ele pertence. Servindo de suporte comunicacional e estratégico para as entidades representativas dos estudantes, centralizando e cooperando diferentes possibilidades de interações.

Descrição Detalhada do Sistema

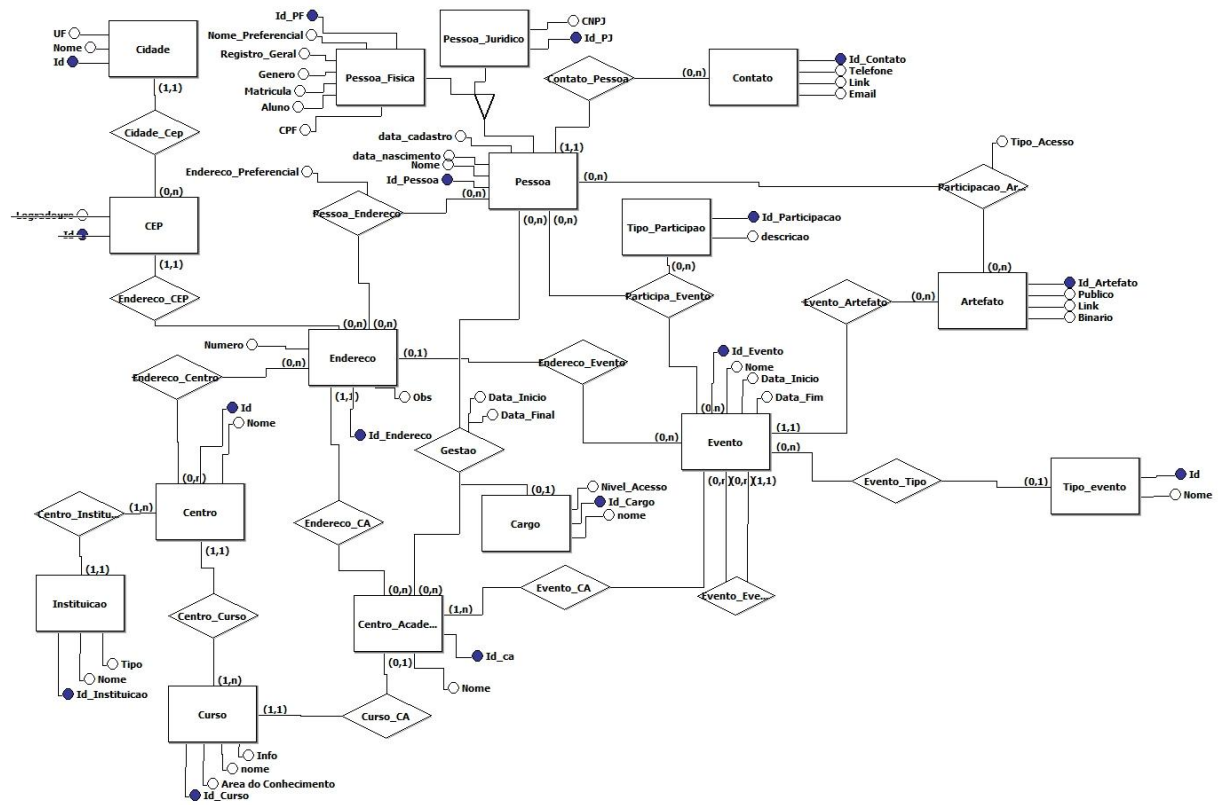
Para alcançar tais objetivos o sistema deve possuir: o registro de todos os estudantes cadastrados nos centros acadêmicos, bem como seus cargos, endereços, formas de comunicação e registros de eventos e que artefatos (documentos, vídeos, ATA's, apresentações) estão vinculados. Já esses centros acadêmicos devem ter o registro da onde estão localizados bem como a que instituições e estruturas (Centros) estão vinculados, além disso deve ser possível realizar eventos entre 1 ou mais centros acadêmicos (conselhos, semanas acadêmicas, reuniões, assembleias). Levando em conta a construção histórica desses mesmos centros acadêmicos todo artefato deve ter um evento associado. Os Eventos podem ter sub-eventos, para melhor controle das atividades como por exemplo as semanas acadêmicas, deve ser possível retornar uma lista de pessoas que participaram de uma oficina numa semana acadêmica.

Pessoas jurídicas como empresas e instituições podem ser cadastradas no sistema e participarem de eventos, como patrocinadores, realizadores, e afins. Nem toda pessoa física necessariamente precisará estar atrelada a um CA, ou instituição. O sistema deve armazenar de pessoas físicas se elas são ou não alunos, a forma preferencial que a pessoa deseja ser chamada e informações cadastrais para registros como CPF e RG (que não serão obrigatórios a não ser que a pessoa realize alguma transação monetária ou material com o CA).

Os endereços devem ser facilmente categorizados por estados da federação sendo assim possível a busca de todos os centros de um determinado estado federativo, uma pessoa pode ter mais de um endereço e deve ser possível assinalar que endereço é o principal da pessoa. Deve se ter alguma forma de armazenar forma de contatar as pessoas..

As gestões do Centro acadêmico devem ser registradas com data de início e fim, para fazer a manipulação de arquivos e configuração o cargo maior (geralmente de presidente) detém acesso total às configurações e acessos aos artefatos. Artefatos podem possuir a característica pública para aparecer para todos numa *homepage*, por exemplo. Deve ser possível descobrir quantos cursos cada instituição tem, quantos integrantes cada centro acadêmico tem.

Modelagem Conceitual



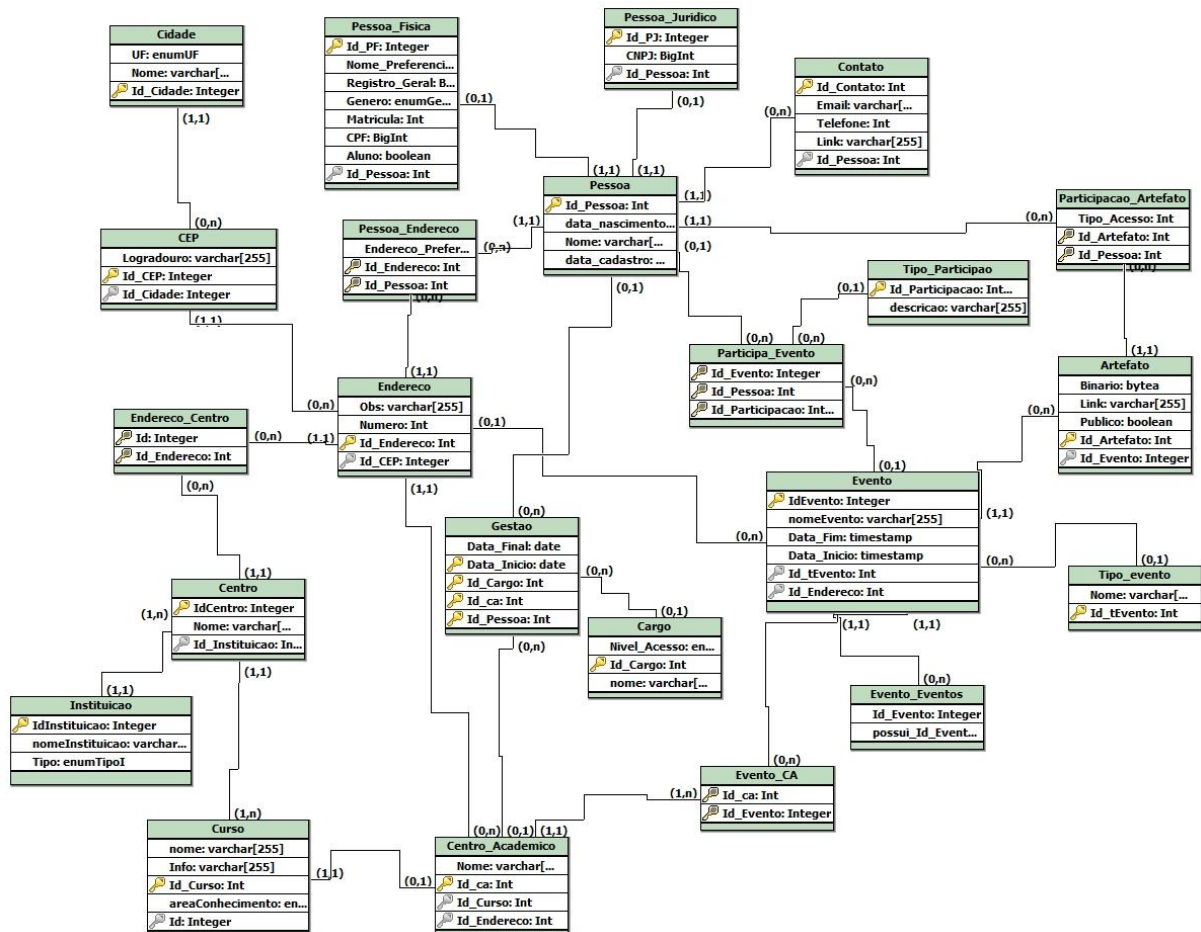
¹Modelo Conceitual

O modelo conceitual foi iniciado pelo Joel no começo do semestre e foi sendo melhorado, o Joel teve algumas reuniões com o professor e Nayara deu algumas dicas finais colaborando para a Descrição Detalhada do Sistema, quais seriam as “colunas” mínimas mas implementamos mais algumas. Apesar de não utilizar todas as funções possíveis com as colunas (dado que não foi criado um aplicativo para o sistema), o modelo se atém a questões de permissões e possibilidades futuras para a composição de um sistema completo, que poderá ser desenvolvido a posteriori. A motivação vem é claro da inconsistência de dados de gestões antigas de centro acadêmico, bem como a normalização para uma comunicação clara, coesa e efetiva. Assim, relacionamos que as seguintes entidades junto dos seus atributos, cumprem as necessidades.

Nesse modelo conceitual temos várias técnicas como por exemplo o auto relacionamento e a especialização. Isso amplia um pouco a complexidade mas supre as necessidades do sistema. Como a ferramenta (BRmodelo) é antiga e relativamente morosa de mexer, os atributos possuem padrões de nomeação diferentes por que foram criados ao longo do tempo, mas que de forma direta não influenciam no resultado do banco de dados.

¹ Imagem obtida através do Modelo Conceitual em anexo 1.

Modelagem Lógica



² Modelo Lógico

A modelagem lógica foi feita de forma "automática" utilizando o BRModelo, indicando a separação de cada entidade especializada, e a criação de indicadores em relações $n \rightarrow 1$ ou semelhantes da entidade com o N menor.

Aqui alguns tipos foram corrigidos de forma manual bem como corrigir as relações que quebraram também, no processo de conversão foi utilizado o correlacionador de tipos para indicar que os Int se transformam em Integer, dado que é mais declarativo.

² Imagem obtida através do Modelo Lógico em anexo 2.

Script DDL

O script DLL³ para a construção do banco, foi criado a partir da ferramenta BRModelo, porém para incluir os tipos *Enum* e as *Functions* foi criado outro script DLL⁴ e esse que carrega o banco de dados final.

Sobre os **tipos ENUM** foram criados: enum_areaConhecimento, enumAcesso, enumTipoI, enumGenero, enumUF. Com o intuito de reduzir a quantidades de tabelas de tipo visto que esses dados até podem mudar, mas são uma forma mais geral e longa de informação. a não ser o enumAcesso que serve diretamente para controlar o acesso (não implementado pois requer diferentes usuarios), haja visto que uma mudança nele também seria uma mudança no código do programa.

As funções criarendereco e criarpessoa servem para resolver um problema de dependência entre algumas tabelas e essas duas endereço e pessoa. Fazendo então de forma “dinamica” a inserção em múltiplas tabelas ao mesmo tempo como é o caso da criarendereco e de resolver da inserção dessincronizada em pessoa e sua especialização.

³ Lista de anexos: ScriptDLL-SGCA.sql

⁴ Lista de anexos: ScriptDLL-SGCA-CORRIGIDO.sql

Consultas

Consulta 1

Objetivo: listar os centros acadêmicos e a quantidade de integrantes que cada um tem.

SQL query: SELECT count(id_pessoa) AS qtdIntegrantes, centro_academico.nome FROM gestao INNER JOIN centro_academico ON centro_academico.id_ca = gestao.id_ca GROUP BY nome ORDER BY qtdIntegrantes DESC;

Integrantes por Centro Academico

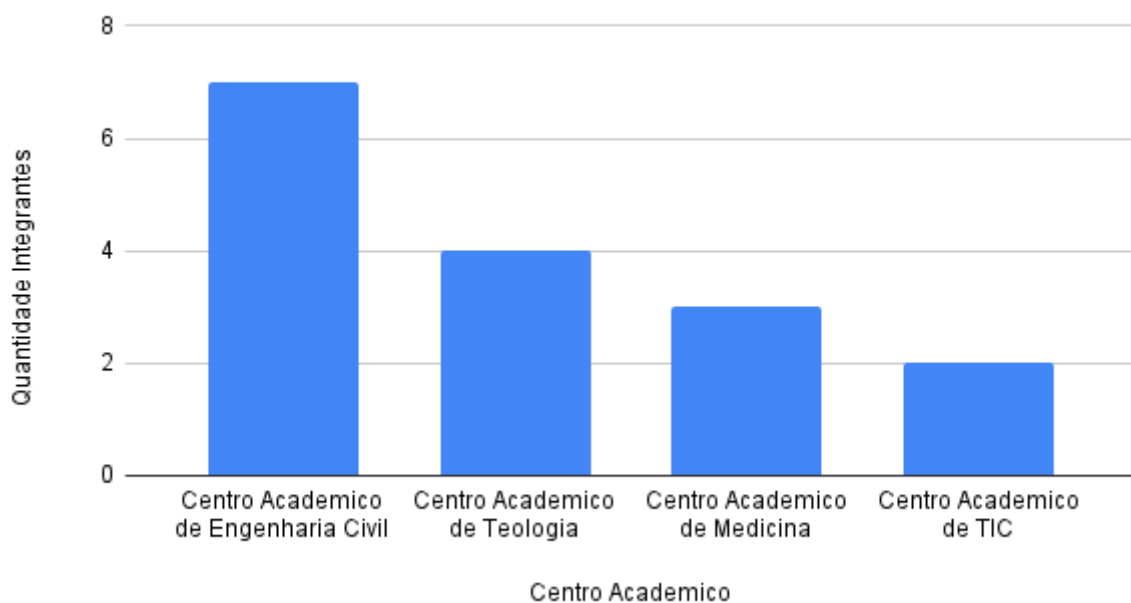


Gráfico resultante consulta 1

Consulta 2

Objetivo: Descobrir a quantidade de cursos em cada instituição.

SQL query: SELECT count(id_instituicao), nomeinstituicao FROM instituicao INNER JOIN (SELECT curso.nome, centro.idcentro, centro.id_instituicao FROM centro INNER JOIN curso ON centro.idcentro = curso.id_centro)AS centro_curso ON idinstituicao = id_instituicao GROUP BY nomeinstituicao;

Quantidade de Cursos por Instituição

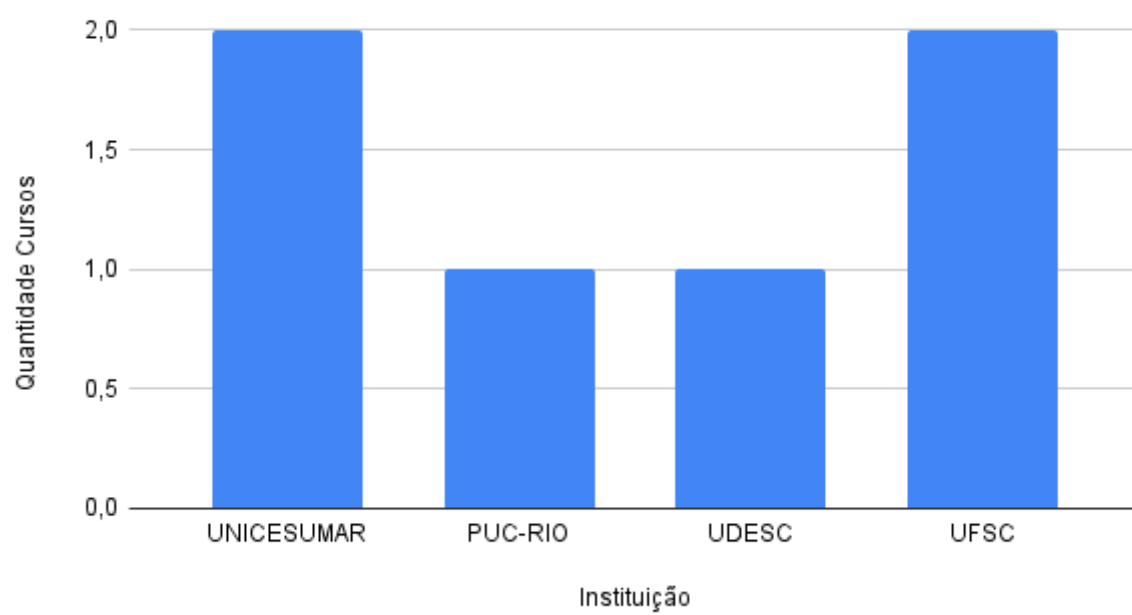


Gráfico resultante da consulta 2

Lista de Anexos

1. ModeloConceitual-SGCA.brM
2. ModeloLogico-SGCA.brM
3. ScriptDLL-SGCA.sql
4. ScriptDLL-SGCA-CORRIGIDO.sql
5. ./SGCA/main.py
6. ./SGCA/strings.py
7. ./SGCA/utility.py
8. ./SGCA/dados.py

Os seguintes anexos estão na pasta raiz que se encontra esse trabalho.