



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Engenharia de Computação

SCC0640 - Base de Dados - 2019.2

Projeto - Banco de dados do Inova Arena (Parte 3)

Games XP

Docente: Profª a Drª Elaine Parros M. de Sousa

Leandro Antonio Silva, 9805341

Maria Luisa do Nascimento da Silva, 10310721

Marilene Andrade Garcia, 10276974

São Carlos, 24 de novembro de 2019

1. Descrição do Projeto

A Game XP é o primeiro GamePark do mundo, misturando parques de diversões com o mundo dos games. A primeira edição do evento ocorreu junto ao Rock in Rio em 2017, mas desde 2018 o evento tem data e local próprio. A terceira edição da Game XP, em 2019, aconteceu entre os dias 25 e 28 de julho no Parque Olímpico da cidade do Rio de Janeiro. O evento contou com a presença de desenvolvedores, artistas, estandes de empresas de games, parques de diversões e muito mais.

Uma das atrações da Game XP é o Inova Arena, um espaço de construção e compartilhamento de conhecimento em tecnologia, onde inovação se torna entretenimento. A Arena conta com a presença de empresas de referência em seus setores trazendo atrações criativas e divertidas, além de talks e painéis discutindo temas como Tecnologia na Educação, Tecnologia na Saúde, Bastidores do Mundo dos Games, Revolução nos Negócios, Cidades Inteligentes e Gadgets of the Future (robôs, dispositivos de comando de voz, roupas inteligentes).

O sistema de base de dados proposto tem como objetivo gerenciar vários aspectos de organização da Inova Arena da Game XP. Os dados mantidos pelo banco permitem não apenas manter um registro completo de todos os participantes, oradores e atividades, mas também a possibilidade de uma análise personalizada desses dados para extração de informações relevantes como número de pessoas em cada palestra, faixa etária por palestra, entre outros.

As **atividades** da Arena são divididas em três tipos: **palestras**, **debates** e **atrações**. Cada atividade possui título, tipo, descrição e público alvo, além das informações de assunto e de horário de cada tipo de atividade. Além disso, elas podem ou não estar associadas a uma **instituição** parceira, detentora de um nome, um CNPJ e um site, a qual pode ser uma **empresa**, **universidade** ou **ONG**, que também têm suas próprias características, como as empresas que tem o valor avaliado.

O espaço de mais de dois mil metros quadrados da Arena Inova é separado em dois tipos de locais: **anfiteatros** para o Inova Stage, em cada um comporta uma atividade por vez e uma **área de exposição** dividida em **slots**. Eles apresentam nome, tipo, capacidade máxima, descrição e informações sobre os níveis de acessibilidade. Além disso, os anfiteatros possuem informações sobre sua infraestrutura - se apresentam microfone, projetor e ar-condicionado. Cada slot da área de exposição comporta uma atração (assim como cada atração ocupa apenas um slot) e são identificados por um código único que o referencia dentro da área de exposição.

No Inova Stage acontecem os debates e palestras, que têm um ou mais **palestrantes** e **moderadores** associados que são as pessoas que ministrarão a atividade. Eles possuem nome, CPF, RG, código de inscrição e currículo. Em debates, é obrigatório a presença de ao menos um moderador para conduzir a discussão. Esses palestrantes e moderadores podem ou não estarem associados a uma instituição parceira do evento.

- As atrações são atividades especiais da Inova Arena e ficam nos slots da área de exposição acontecendo conjuntamente ao Inova Stage. Essas atrações vão desde o

lançamento e teste de **produtos** até quizzes e jogos. Os produtos possuem nome, descrição e uma data de lançamento e obrigatoriamente têm uma empresa associada. Também é possível que uma universidade ou uma ONG apresentem algum **projeto**, que tem nome e descrição.

Os **participantes** da Inova Arena são identificados por um código de inscrição único, além de possuir vários outros dados atrelados (nome, CPF, RG, nível de acesso, email, telefone, idade e escolaridade). A cada atividade que comparecer, o participante receberá um certificado com as características da atividade em questão. Não há um limite para as atividades em que cada pessoa irá participar, mas as atividades possuem um número máximo de vagas que precisa ser respeitado.

2. Modelo Entidade Relacionamento - MER

Consultar o anexo 1.

3. Modificações do MER

As seguintes modificações foram realizadas no Modelo Relacional do projeto:

- Foi colocado o critério de especialização *Tipo* nas entidades *Instituição* e *Atividade*;
- Foi adicionada a generalização *Associado*, que possui as especializações *Moderador*, *Palestrante* e *Expositor*, e somente o genérico tem relacionamento com *Instituição*;
- Foi corrigida a representação da entidade *Slot*, adicionando a chave parcial e o retângulo da entidade fraca.
- Foi adicionada a representação de participação total na relação *Comporta* de *Palestra*, assim como *Debate* e na relação *Participa* de *Debate*;

4. Modelo Relacional

Consultar o anexo 2.

5. Modificações do Modelo Relacional

Respostas aos questionamentos feitos na correção da Parte 2 do trabalho a respeito do Modelo Relacional (numeração de itens referente a seção 6 - *Justificativas do Modelo relacional* do relatório da parte 2 do projeto):

- **Item a:** A especialização total é garantida pois o atributo tipo (critério de especialização nas entidades) não pode ser nulo. Ademais, a disjunção é garantida pois o atributo tipo é monovalorado;
- **Item b:** A desvantagem de criar a generalização de *Instituição* é criar mais tabelas, de forma a aumentar a complexidade do modelo e diminuir o desempenho. Uma alternativa seria não criar a generalização e colocar os atributos específicos na própria tabela da entidade abstrata, de forma que eles assumam ou não o valor nulo

dependendo do atributo tipo, mas essa alternativa poderia causar inconsistências no modelo;

- **Item c:** A unicidade do nome do projeto para uma instituição é garantida, mas não para instituições diferentes, nesse caso teria que ser resolvido na implementação;
- **Item d:** foram adicionadas as representações de participação total de anfiteatro e SLOTS no MER;
- **Item h:** Evita por exemplo que diferentes palestras e debates sejam registrados no mesmo horário e mesmo local de atividades do mesmo tipo. De outra forma, não é possível garantir que uma palestra e debate não sejam registrados no mesmo lugar e horário, isso deve ser tratado na implementação;
- **Item i:** O atributo *RG* foi colocado como terciário junto com o *RGUF* no Relacional.

Foram realizadas as mudanças abaixo no modelo relacional entregue na parte anterior do projeto:

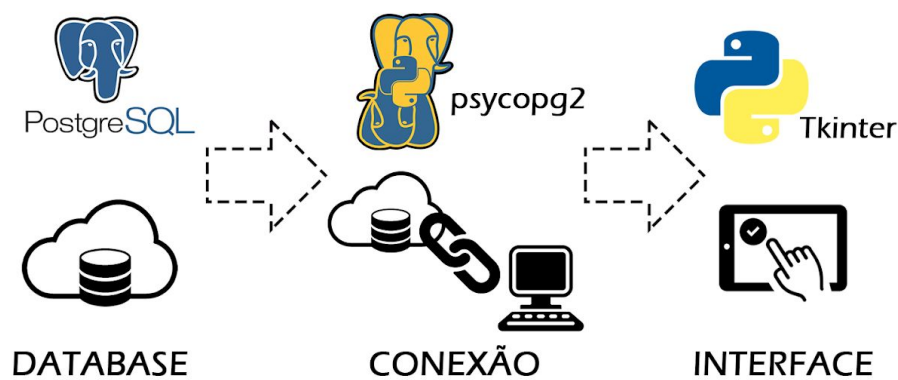
- Adicionada a tabela *Associado* e a tabela *TipoAssociado*, pois um associado pode ter mais de uma especialização para diferentes atividades;
- Foram excluídas as tabelas de *Expositor*, *ExpositorHor*, *DebatesMorad*, *DebatesPalest*, *Moderador* e *Palestrante*, pois *Associado* possui as características de todas;
- A tabela slot foi recolocado, pois foi concluída que a relação é 1:N;
- A tabela *HorarioAtracao* foi substituída pela *InfoAtracao*, que também mantém as informações de local de cada atração.

6. Implementação

a. Introdução

A partir do Modelo Relacional, uma base de dados foi implementada em **SQL** e uma aplicação foi desenvolvida para interface com usuário, permitindo a criação das tabelas, a inserção de tuplas e consultas estratégicas dentro do banco.

A base de dados foi implementada no **PostgreSQL**, um poderoso e gratuito banco de dados, e gerenciada pela ferramenta de nuvem **pgAdmin 4**. A interação com o banco e a interface com o usuário foram feitas em **Python 3**, utilizando as bibliotecas **psycopg2** e **tkinter**, respectivamente. O esquema abaixo sumariza o projeto de implementação.



Os códigos e arquivos desenvolvidos na implementação foram:

- read_me: instruções de instalação dos pacotes necessários e de como executar o projeto.
- main.py: código central em python, faz a conexão com banco, roda os arquivos em SQL e faz a chamada do código da interface com o usuário.
- interface.py: código da interface com o usuário.
- database.ini: seta parâmetros iniciais para conexão com a nuvem onde o banco de dados está armazenado.
- drop.sql: comandos em SQL para remoção de todas as tabelas antes de cada execução do projeto, para resetar o banco.
- create_tables.sql: comandos em SQL para criação as tabelas.

O projeto foi inteiramente desenvolvido em um sistema **Linux Ubuntu 16.04 e 18.04**. Para execução, basta seguir as instalações necessárias descritas no arquivo READ_ME.txt e rodar o código main em Python.

b. Base de dados

i. Criação

A criação das tabelas com seus atributos e restrições foi feita no arquivo create_tables.sql.

Exemplo com chave primária, atributo não nulo e restrição de conteúdo no atributo Tipo (especialização total de Instituição para ONG, Universidade e Empresa)

```

create table Instituicao(
    CNPJ int,
    Tipo varchar(12) not null,
    ESite varchar(50),
    constraint pk_Instituicao primary key (CNPJ),
    constraint ck_CNPJInstituicao check (floor(log(abs(CNPJ))+1) = 9),
    constraint ck_TipoInst check (upper(Tipo) in ('ONG', 'UNIVERSIDADE', 'EMPRESA'))
  
```

```
);
```

Exemplo com chave estrangeira para outra tabela e checagem de restrição da forma de Código de Inscrição (atributo Associado) que é aceito. Além disso, é colocada uma condição on delete nas chaves estrangeiras em caso de deleção da tabela mãe.

```
create table TipoAssociados(  
    Associado int,  
    Atividade varchar(80),  
    Tipo varchar(11) not null,  
    constraint pk_TipoAssociados primary key (Associado, Atividade),  
    constraint ck_CPF_TipoAssociados check (floor(log(abs(Associado))+1) = 8),  
    constraint ck_Tipo_TipoAssociados check (upper(Tipo) in ('PALESTRANTE',  
'MODERADOR', 'EXPOSITOR')),  
    constraint fk1_TipoAssociados foreign key (Associado) references  
Associado(CodigoInscricao) on delete cascade,  
    constraint fk2_TipoAssociados foreign key (Atividade) references Atividade(Titulo)  
on delete cascade  
);
```

ii. Inserções

As inserções foram feitas no arquivo insert.sql.

Exemplo de inserção:

```
insert into Anfiteatro values ('Inova Stage','Aberto' , 5000, 'Cadeira, palco', 'Rampas,  
Corrimões, Iluminação nas escadas', true, false, false);
```

iii. Consultas

As consultas foram organizadas no arquivo select.sql.

Exemplo de busca de quantos projetos ou produtos cada instituição expõe. Utilizando *join* para unir as tabelas *ONG* e *ProjetoONG*, *Universidade* e *ProjetoUniversidade* e *Empresa* e *Produto*, *group by* por nome da instituição, *count* para calcular o números de tuplas resultantes, além do *union* para juntar os resultados de cada tipo de instituição no resultado final.

```
select O.Nome, count(P.*) as "Num Produtos"  
from ONG O join ProjetoONG P on O.Instituicao = P.ONG  
group by O.Nome  
union  
select U.Nome, count(P.*) as "Num Produtos"  
from Universidade U join ProjetoUniversidade P on U.Instituicao = P.Universidade  
group by U.Nome
```

```
union
select E.Nome, count(P.*) as "Num Produtos"
from Empresa E join Produto P on E.Instituicao = P.Empresa
group by E.Nome;
--Quantos projetos ou produtos de cada tipo de instituição é exposto
```

Exemplo que busca algumas informações das atrações a partir das tabelas *Atração*, *InfoAtração* e *Exposição*. Primeiro é realizada a junção interna de *Atração* com *InfoAtração* e em seguida com *Exposição*, recebendo o título, tema, data e horário de início, o local de exposição e o Slot em que acontecerá a atração. O resultado é retornado em ordem alfabética.

```
select C.Atividade, C.Tema, C.Brinde, E.DatHorInicio, E.Duracao, E.AreaExpo, E.NumSlot,
X.Expositor
from Atracao C inner join InfoAtracao E on C.Atividade = E.Atracao inner join Exposicao X
on X.Atracao = C.Atividade
order by C.Atividade;
-- Info de atrações
```

iv. Remoção

As remoções foram feitas no arquivo drop.sql.

Exemplo de remoção:

```
DROP TABLE IF EXISTS Anfiteatro CASCADE;
```

c. Interface com o usuário

i. Tela inicial

Na figura 1.1 é apresentada a tela inicial da interface são apresentadas as funcionalidades possíveis, vale ressaltar que conforme as instruções do projeto, a interface é um protótipo operacional simples, implementado com uma funcionalidade de cadastro de dados (tabela Associado) e uma funcionalidade de consulta (informações de uma atividade).



Figura 1.1. Tela inicial da interface

ii. Tela de da funcionalidade Inserções

Na figura 1.2 é apresentada a tela de possíveis Inserções da interface. Quando o botão no qual está escrito Inserções é pressionado, é aberta uma segunda janela com as inserções possíveis, apenas a inserção de associado foi implementada. É possível reparar que nem todas as tabelas aparecem como possíveis inserções, pois as tabelas que são derivadas de outras deveriam ser inseridas com os dados já digitados pelo usuário para a primeira tabela, ou deveria ser aberta uma nova janela na qual o usuário poderia decidir se gostaria de inserir a tabela derivada da primeira.

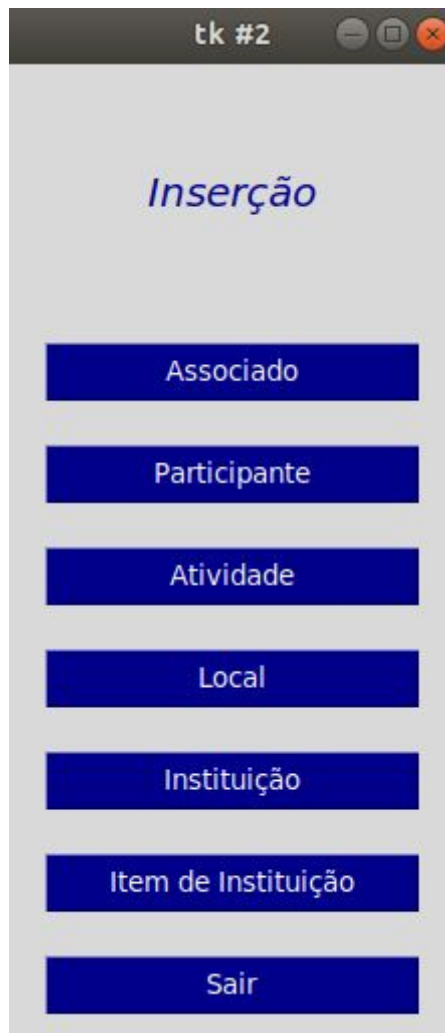


Figura 1.2. Tela das possíveis inserções

iii. Tela da Inserção dos dados de Associado

Na figura 1.3 é apresentada a tela de inserção para a Tabela associado, os campos devem ser preenchidos e ao final o botão de confirmar apertado para que sejam abertas ou a tela com a mensagem que a inserção foi realizada com sucesso, ou a tela com a mensagem de que os dados são inválidos e o erro que ocorreu.

tk #3

Associado

Código Inscrição

Código CPF

Digitos RG

UF RG

Nome

Instituição

Curriculo (opcional)

Confirmar Sair

Figura 1.3. Tela para a inserção dos dados da tabela Associado

iv. Tela de inserção realizada com sucesso

Caso a inserção ocorreu com sucesso no banco de dados é aberta a janela da figura 1.4.

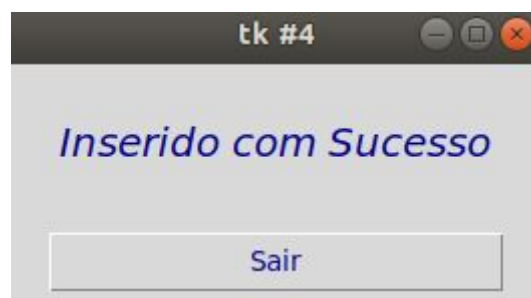


Figura 1.4. Tela de inserção realizada com sucesso

v. Tela de erro na inserção

Caso algum dado seja inválido e os dados não seja inseridos no banco, é aberta a janela com a mensagem de dados inválidos e o erro que ocorreu, como apresentado na figura 1.5.

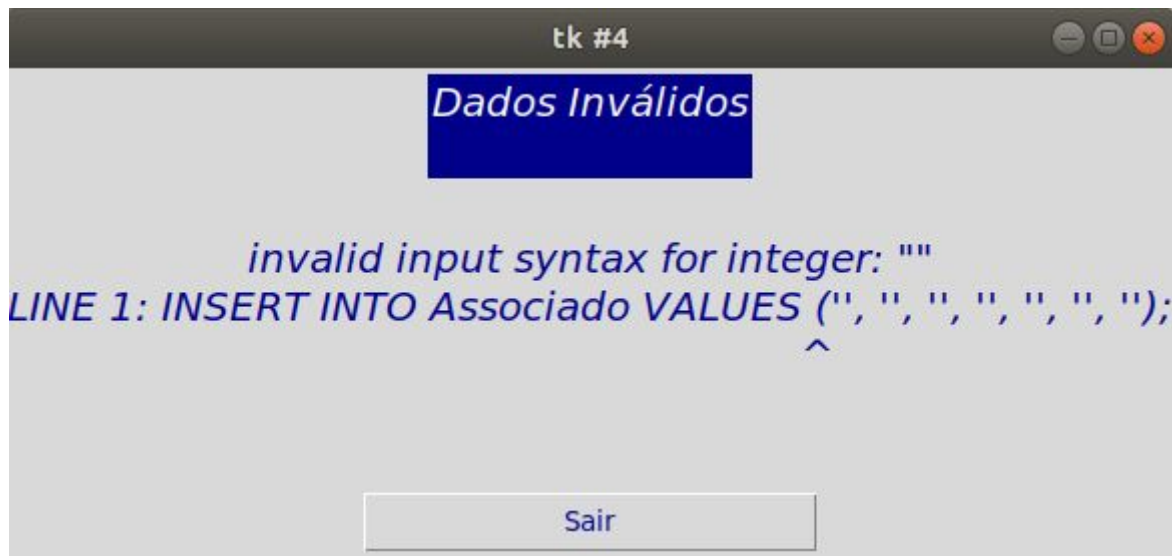


Figura 1.5. Tela de inserção realizada com dados inválidos

vi. Tela de da funcionalidade Consultas

Na figura 1.6 é apresentada a tela de possíveis Consultas da interface. Quando o botão no qual está escrito Consultas é pressionado, é aberta uma segunda janela com as consultas possíveis, apenas a consulta de informações de uma atividade foi implementada.



Figura 1.6. Tela das possíveis consultas

vii. Tela da Inserção dos dados da consulta Informação Atividades

Na figura 1.7 é apresentada a tela de inserção a consulta que retorna as informações de todas as atividades do banco.

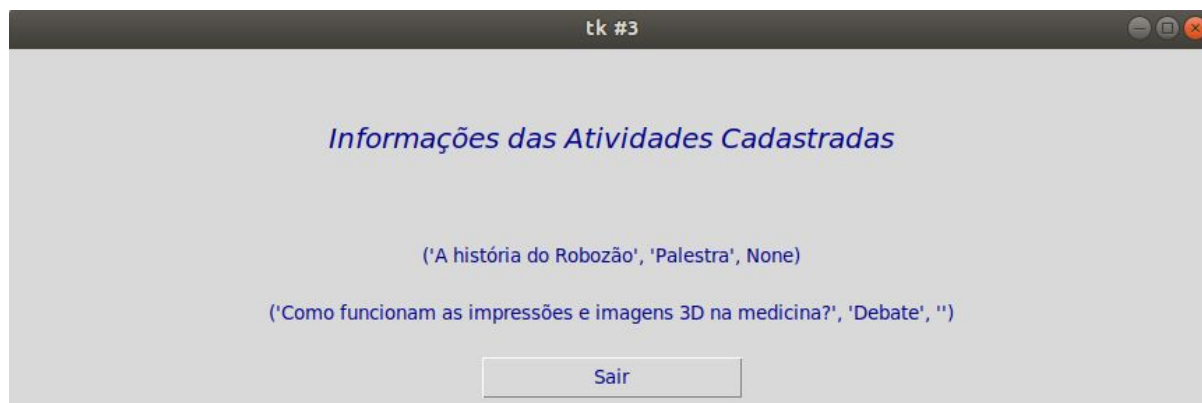


Figura 1.6. Tela das informações das atividades cadastradas

7. Conclusão

O grupo aprendeu muito com o trabalho, conseguindo colocar em prática os conceitos aprendidos em sala de aula. Assim como a divisão em três partes, fragmentando o conteúdo em cada entrega de maneira que coincidissem com o conteúdo ministrado em sala foi muito interessante, e não ficou um único trabalho muito massante no final do semestre.

A maior dificuldade em todas as partes do trabalho foi no início, onde não estava ainda claro quais eram os primeiros passos a serem seguidos. Assim que tinha-se um norte de por onde começar, ficava mais claro o que tinha que ser feito.

A dinâmica de corrigir as partes anteriores na parte seguinte do trabalho ajuda na fixação da matéria e aprendizado com os erros, mas ter que modificar muitas coisas além de justificar cada mudança acaba pesando ainda mais as partes 2 e 3 do trabalho.

Contudo, principalmente a Parte 3 ficou muito pesada, por conta do desenvolvimento em SQL além da implementação do banco utilizando um software que nenhum dos integrantes do grupo tinha nenhuma familiaridade com. Uma sugestão seria disponibilizar alguns tutoriais de como utilizar as ferramentas que a professora recomendar, e talvez dar uma aula extra explicando como usar uma das ferramentas.

Resumindo, o trabalho em si é uma ótima forma de colocar em prática os conhecimentos aprendidos em sala, sendo uma boa maneira de estudar, embora acabe sendo um pouco cansativo e pesado e que muitas vezes os maiores problemas estão na utilização de ferramentas de implementação do que na parte referente ao conteúdo da matéria em si.