

UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC

LUCAS FRANCO TUMELEIRO

LUIZ FERNANDO COPPINI DE LIMA

GUSTAVO VINCENZI

WILLIAN MARCHESAN

OLDTECH: Um sistema de controle de museu

SÃO MIGUEL DO OESTE

2019

LUCAS FRANCO TUMELEIRO

LUIZ FERNANDO COPPINI DE LIMA

GUSTAVO VINCENZI

WILLIAN MARCHESAN

OLDTECH: Um sistema de controle de museu

Projeto final apresentado às disciplinas de Banco de dados I, Engenharia de software I e Programação III do curso de Ciência da Computação da Universidade do Oeste de Santa Catarina.

SÃO MIGUEL DO OESTE
2019

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO..... | 3 |
| 2 | DESENVOLVIMENTO..... | 4 |
| 2.1 | ENGENHARIA DE SOFTWARE I..... | 4 |
| 2.1.1 | Casos de uso..... | 4 |
| 2.1.2 | Diagrama de sequência..... | 8 |
| 2.1.3 | Diagrama de Atividade | 10 |
| 2.1.4 | Diagrama de Estado | 11 |
| 2.1.5 | Diagrama de Classe | 13 |
| 2.2 | BANCO DE DADOS I..... | 13 |
| 2.2.1 | Diagrama Relacional | 14 |
| 2.2.2 | Dicionário de Dados | 14 |
| 2.2.3 | Scripts para criação | 14 |
| 2.2.4 | Script para deletar tabelas e restrições | 21 |
| 2.3 | PROGRAMAÇÃO iii..... | 24 |
| 3 | Conclusão..... | 27 |
| | Referências | 28 |
| | APENDICE A – Dicionário de Dados | 29 |

1 INTRODUÇÃO

No presente artigo será apresentado os passos utilizados para a criação do sistema de controle de Museu denominado “OldTech”, iniciado pelos modelos e fluxos de caso de uso, diagramas de sequência, atividade, estado e classe. Em seguida demonstrando o diagrama relacional, dicionário de dado, *scripts* e SQLs. Por último, será apresentado as telas do sistema.

2 DESENVOLVIMENTO

2.1 ENGENHARIA DE SOFTWARE I

Antes de programar um *software*, torna-se necessário estudar e elaborar a necessidade real do requisitante, para que o processo de criação de um sistema torne-se metódica, evitando desta forma erros por parte do programador ou pelo mal entendimento de ambas as partes. Seguindo está premissa, abaixo será demonstrado os modelos e fluxos de caso de uso, diagramas de sequência, atividade, estado e classe utilizados na criação do projeto Final.

2.1.1 Casos de uso

Caso de uso 1 - Login

Caso de Uso - Fazer login

Objetivo: Neste caso de uso possibilita que um usuário faça login no sistema.

Atores: Usuário

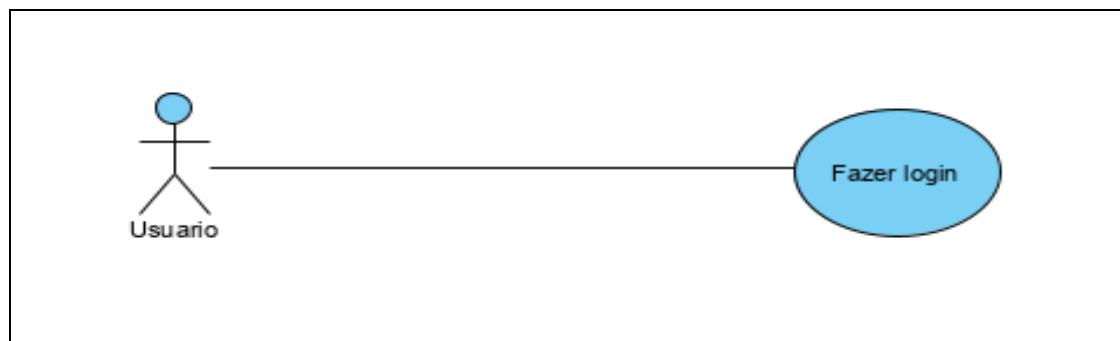
Pré-Condição: O sistema estar funcionando e o usuário já cadastrado.

Fluxo Principal

1. O sistema exibe uma tela para informar login e senha.
2. O usuário digita o login e senha.
3. O sistema verifica se os dados informados são válidos.
4. O sistema abre a tela principal
5. Caso de uso se encerra.

Fluxos Alternativos

2. O usuário não informa login ou senha.
 - 2.1. O sistema exibe mensagem pedindo para informar login e senha
 - 2.2. O sistema foca no primeiro campo que falta ser informado
3. Dados inválidos
 - 2.1. O sistema exibe uma mensagem de dados inválido.
 - 2.2. O sistema limpa a senha e foca no campo de login.



Fonte: Autores

Caso de uso 2 - Menu

Caso de Uso - Abrir um menu da tela principal

Objetivo: Este caso de uso possibilita ao usuário abrir um menu para exibir mais informações ou realizar outras ações como cadastro de dados.

Atores: Usuário

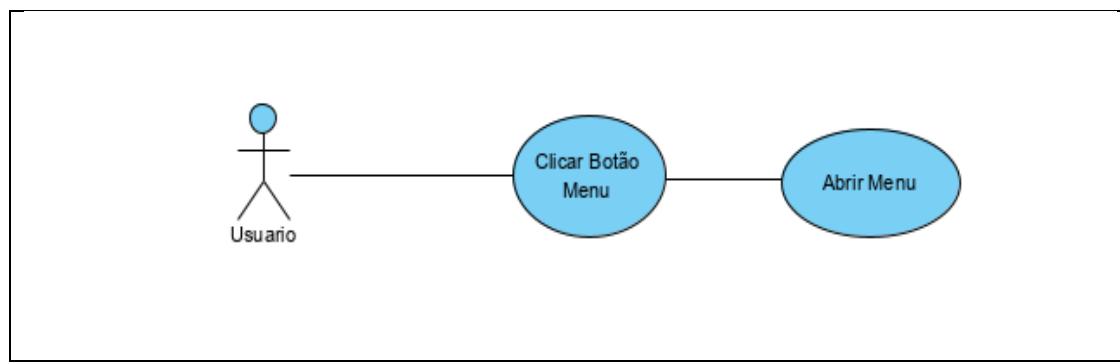
Pré-Condição: O sistema estar funcionando e exibindo os menus, o usuário ter feito login com sucesso.

Fluxo Principal

1. O sistema exibe uma tela principal listando os menus disponíveis.
2. O usuário clica em um menu.
3. O sistema abre uma nova tela referente ao menu selecionado.
4. Caso de uso se encerra.

Fluxos Alternativos

2. O usuário não selecionar nenhum menu.
- 2.1. Caso de uso se encerra.



Fonte: Autores

Caso de uso 3 – Cadastro em Formulário

Caso de Uso - Cadastrar um novo dado no sistema

Objetivo: Neste caso de uso possibilita que um usuário cadastre novos dados no sistema.

Atores: Usuário

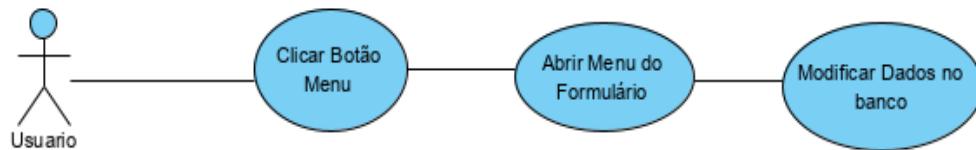
Pré-Condição: O usuário ter selecionado um menu de cadastro.

Fluxo Principal

1. O sistema exibe uma tela de cadastro para informar os dados.
2. O usuário preenche todos os campos necessários.
3. O sistema valida se os dados foram informados corretamente.
4. O sistema salva os dados no banco.
5. Caso de uso se encerra.

Fluxos Alternativos

3. O usuário não informa os dados corretamente.
 - 3.1. O sistema exibe mensagem informando que possui dados inválidos.
 - 3.2. O sistema foca no primeiro campo que está incorreto.
4. Erro ao salvar no banco
 - 4.1. O sistema exibe uma mensagem que não foi possível salvar no banco.



Fonte: Autores

Caso de uso 4 – Gerar relatório

Caso de Uso - Gerar um relatório

Objetivo: Neste caso de uso possibilita ser gerado um relatório com os dados do sistema.

Atores: Usuário

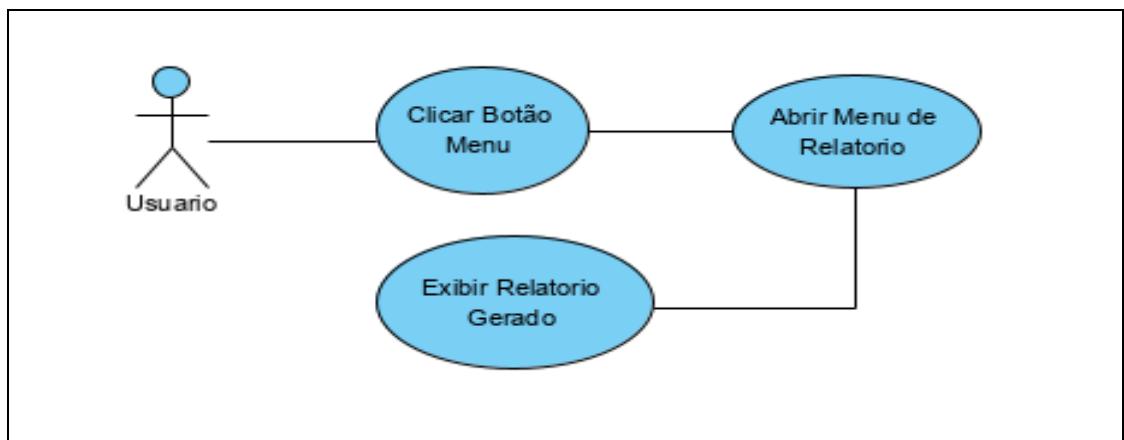
Pré-Condição: O usuário ter selecionado um menu de relatório

Fluxo Principal

1. O sistema exibe uma tela de carregando dados.
2. O sistema busca os dados no banco para o relatório.
3. O sistema exibe um relatório com esses dados.
4. Caso de uso se encerra.

Fluxos Alternativos

2. Não encontrou dados.
 - 2.1. O sistema exibe mensagem informando que não foi encontrado nenhum dado.
 - 2.2. Caso de uso se encerra.
3. Erro ao gerar relatório.
 - 3.1. O sistema exibe uma mensagem que não foi possível gerar o relatório.
 - 3.2. Caso de uso se encerra.



Fonte: Autores

Caso de uso 5 – Tela Sobre

Caso de Uso - Abrir a tela de sobre

Objetivo: Este caso de uso possibilita ao usuário abrir a tela de sobre, contendo informações do sistema e de quem desenvolveu.

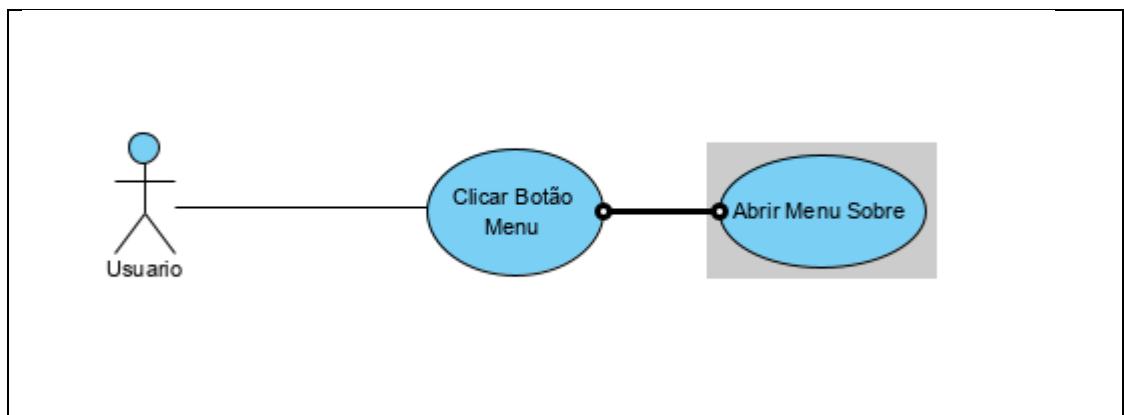
Atores: Usuário

Pré-Condição: O usuário ter clicado no menu de sobre.

Fluxo Principal

1. O sistema exibe a tela de sobre contendo informações do sistema e de quem desenvolveu.

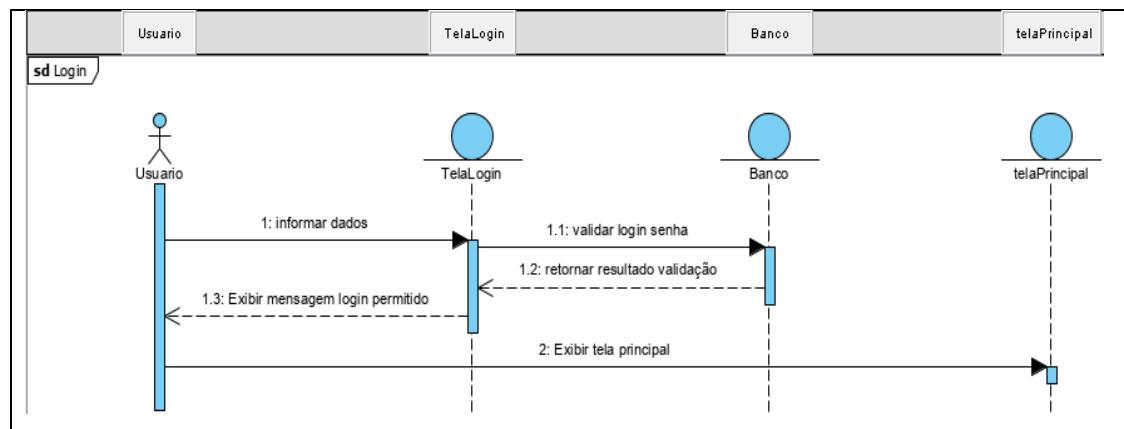
2. Caso de uso se encerra.



Fonte: Autores

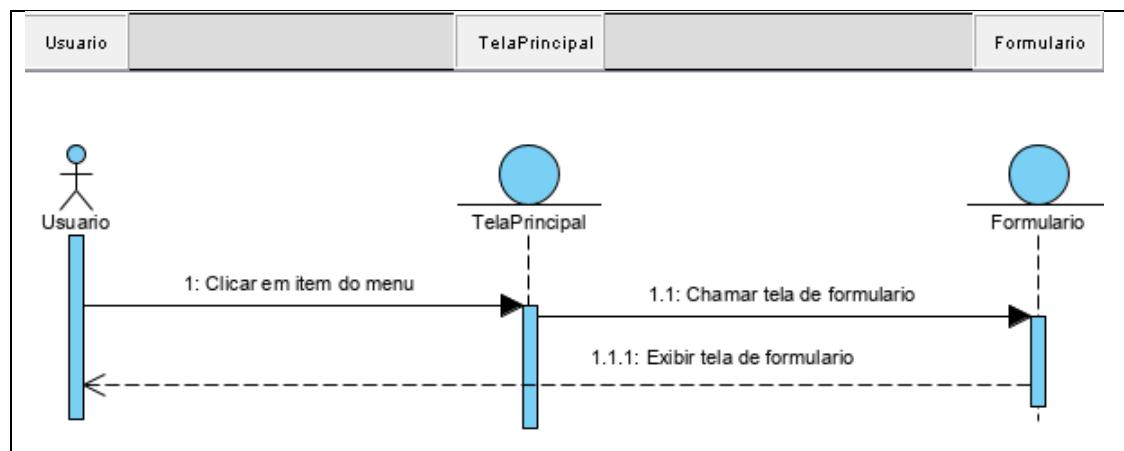
2.1.2 Diagrama de sequência

Diagrama de Sequência 1 - Login



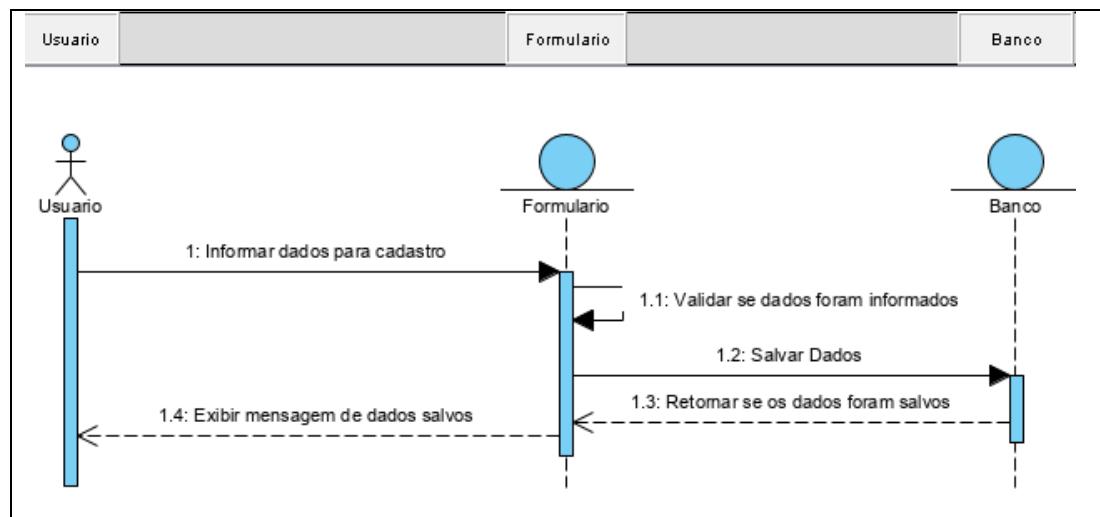
Fonte: Autores

Diagrama de Sequência 2 – Acesso a formulário



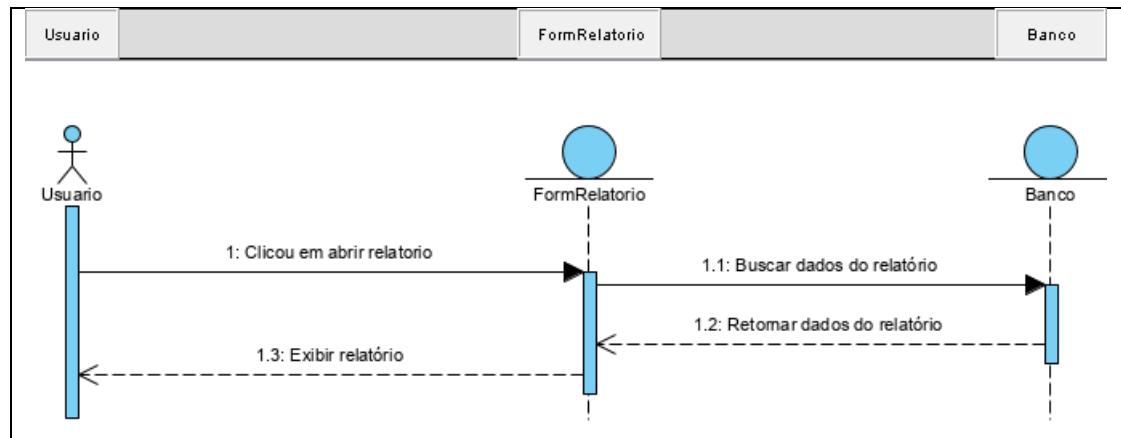
Fonte: Autores

Diagrama de Sequência 3 – Cadastro de Informação



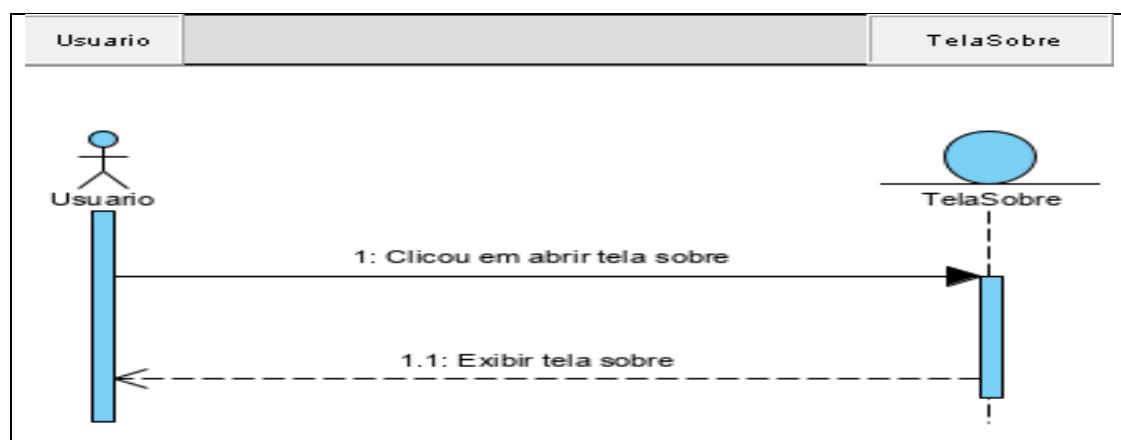
Fonte: Autores

Diagrama de Sequência 4 – Gerar Relatório



Fonte: Autores

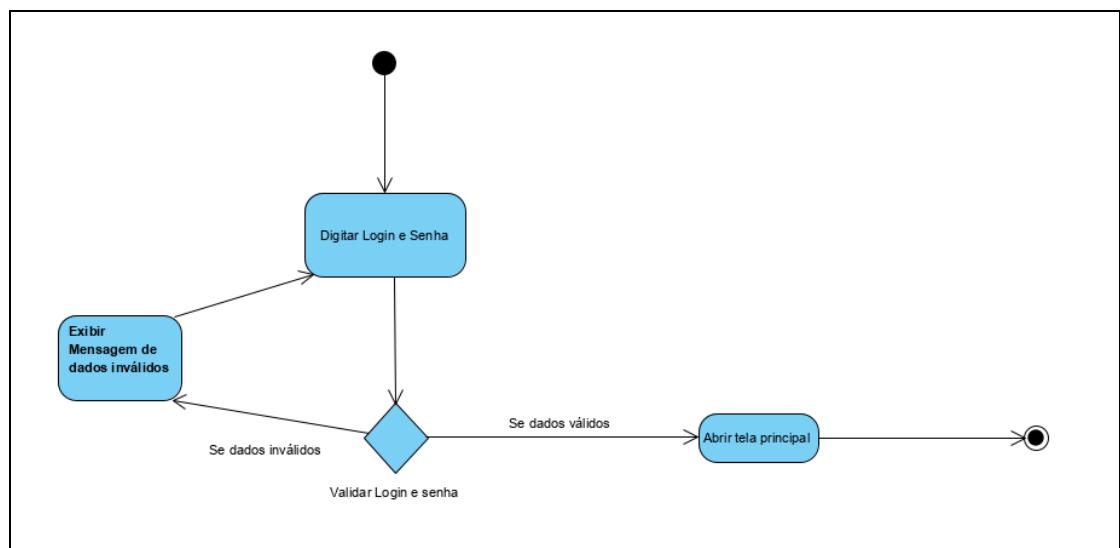
Diagrama de Sequência 5 – Tela sobre



Fonte: Autores

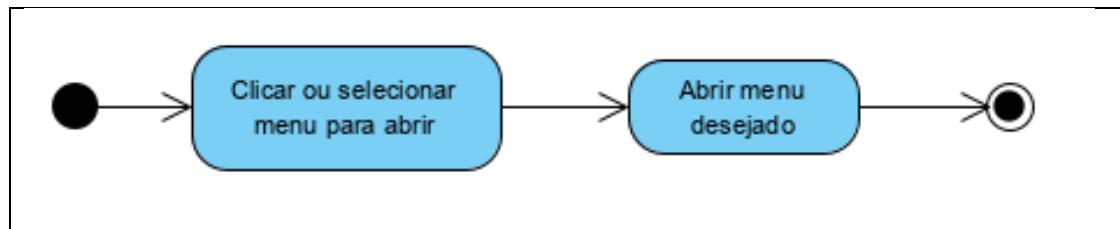
2.1.3 Diagrama de Atividade

Diagrama de Atividade 1 – Login



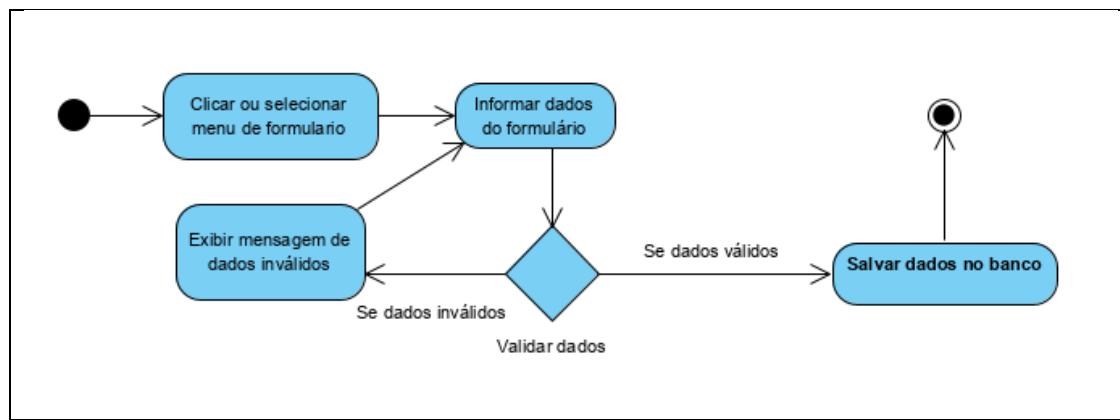
Fonte: Autores

Diagrama de Atividade 2 – Menu



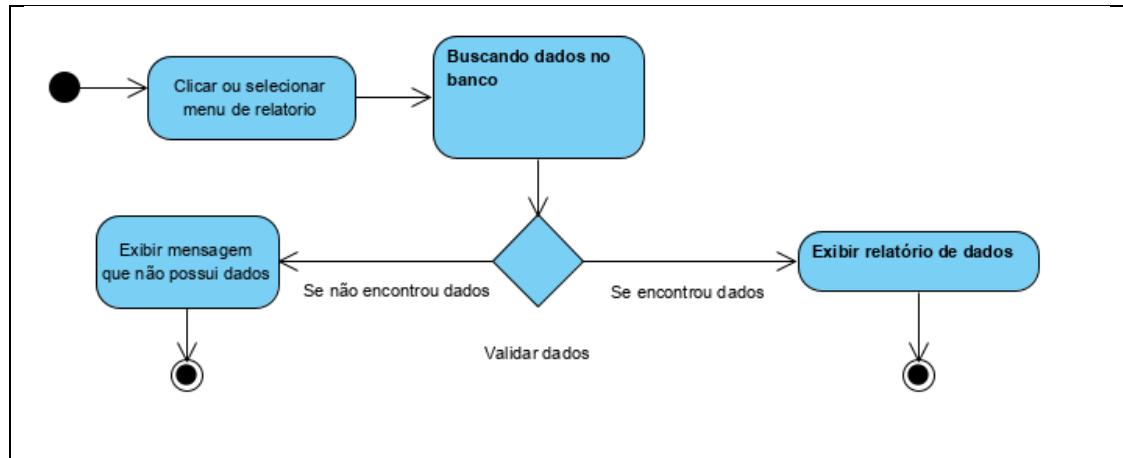
Fonte: Autores

Diagrama de Atividade 3 – Cadastro de informação



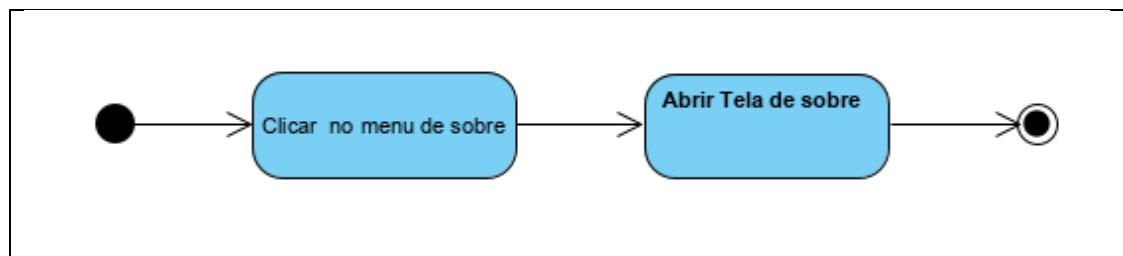
Fonte: Autores

Diagrama de Atividade 4 – Gerar Relatório



Fonte: Autores

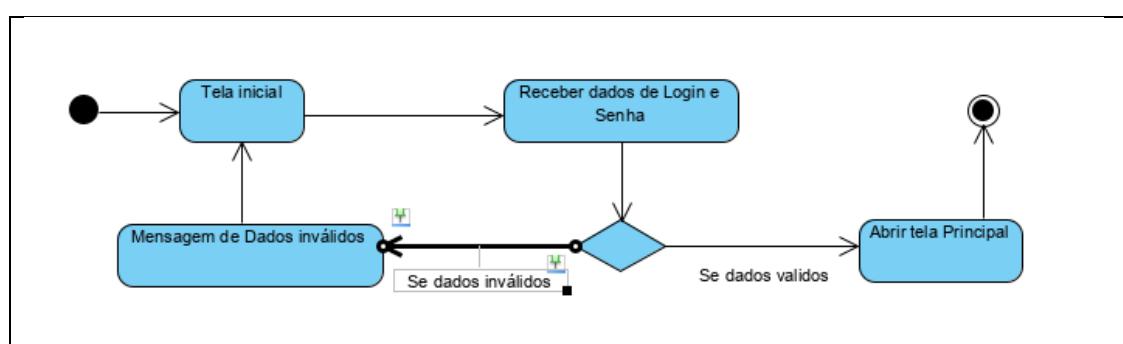
Diagrama de Atividade 1 – Tela sobre



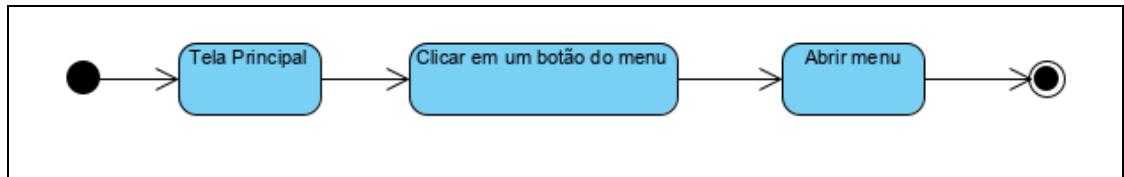
Fonte: Autores

2.1.4 Diagrama de Estado

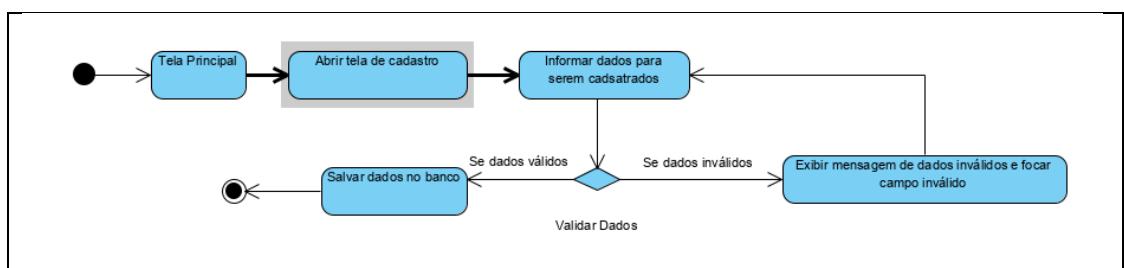
Diagrama de Estado 1 – Login



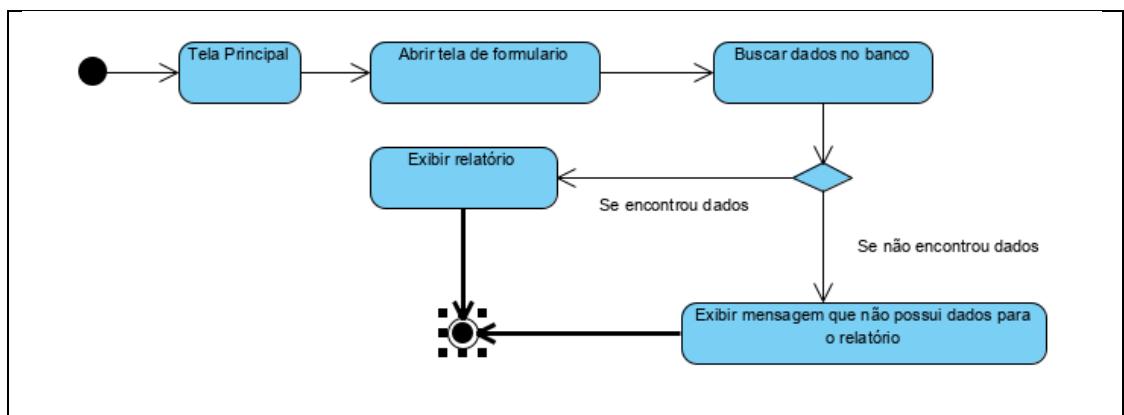
Fonte: Autores

Diagrama de Estado 2 – Menu

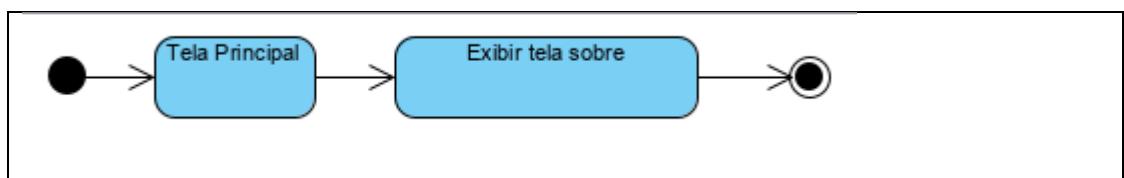
Fonte: Autores

Diagrama de Estado 3 – Cadastro de informação

Fonte: Autores

Diagrama de Estado 4 – Gerar Relatório

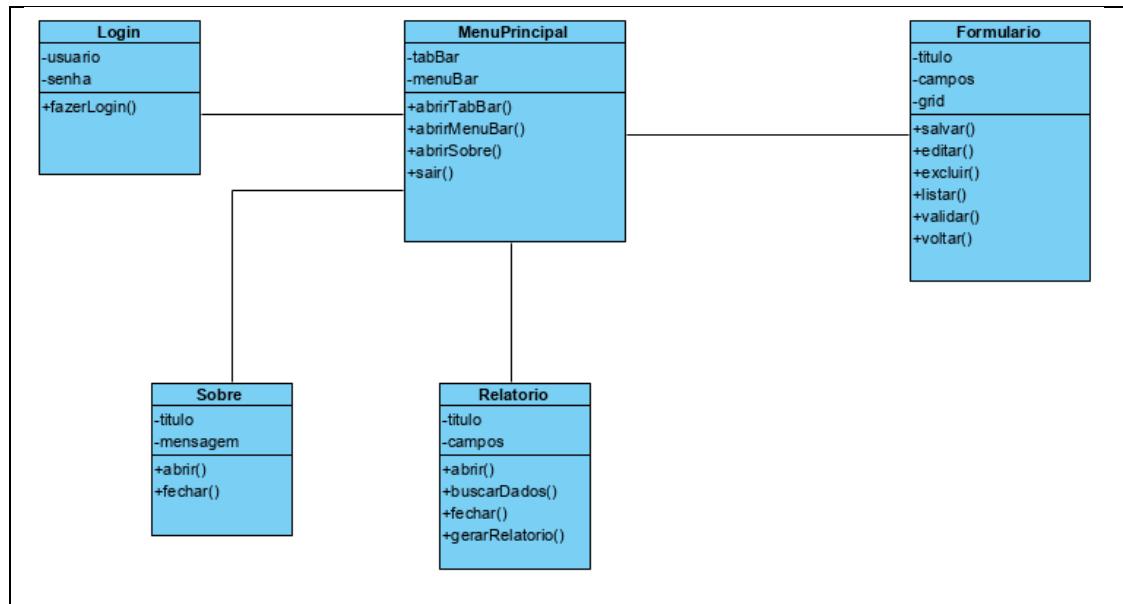
Fonte: Autores

Diagrama de Estado 5 – Tela sobre

Fonte: Autores

2.1.5 Diagrama de Classe

Diagrama de Classe



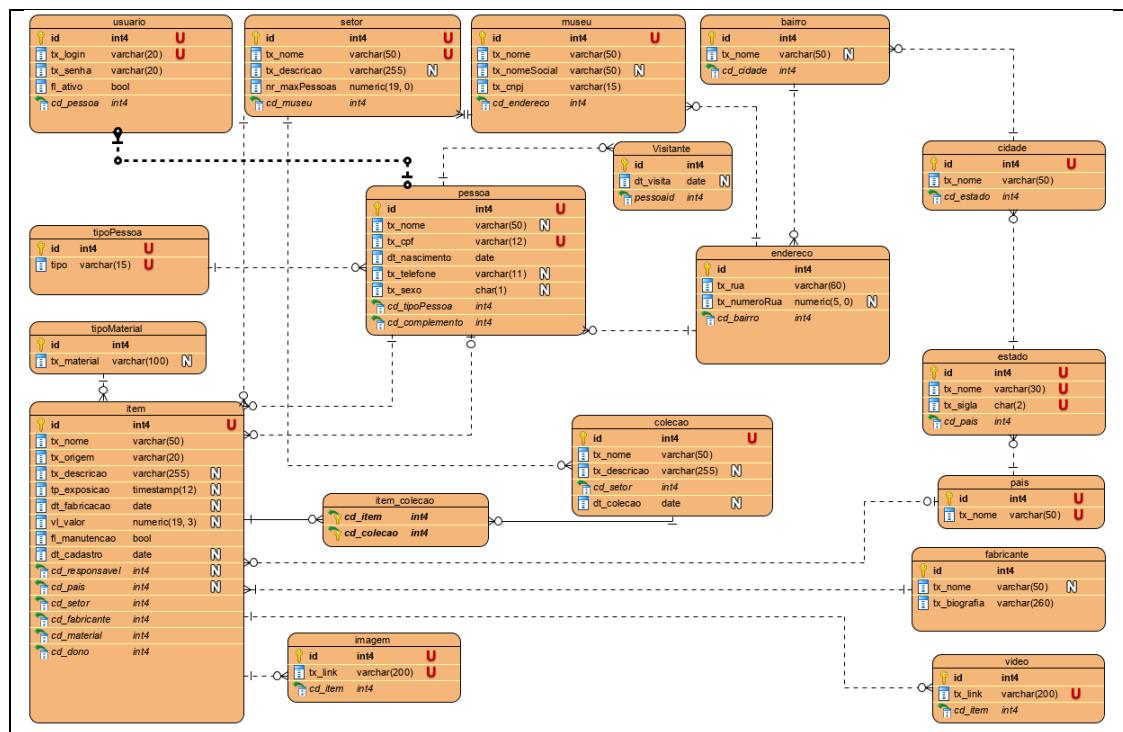
Fonte: Autores

2.2 BANCO DE DADOS I

O banco de dados pode ser definido segundo Connolly e Begg (2005), como uma coleção de dados logicamente relacionados, cada qual com sua respectiva descrição, com o objetivo de atender as necessidades propostas em sua demanda. Sendo o banco de dados relacional utilizado no projeto final.

2.2.1 Diagrama Relacional

Diagrama Relacional – Base de dados



Fonte: Autores

2.2.2 Dicionário de Dados

O Dicionário de Dados encontra-se no Apêndice A- Dicionário de Dados.

2.2.3 Scripts para criação

Script 1 – Criação do Banco de Dados

```
create database dadosmuse;
```

Fonte: Autores

Script 2 – Criação de tabela

```

create table
    bairro (id serial not null,
    tx_nome varchar(50),
    cd_cidade int4 not null,
    primary key (id));

comment on
column bairro.tx_nome is 'Nome do bairro';

create table
    cidade (id serial not null,
    tx_nome varchar(50) not null,
    cd_estado int4 not null,
    primary key (id));

comment on
column cidade.tx_nome is 'Nome da cidade';

create table
    colecao (id serial not null,
    tx_nome varchar(50) not null,
    tx_descricao varchar(255),
    cd_setor int4 not null,
    dt_coleta date,
    primary key (id));

comment on
column colecao.tx_nome is 'Nome da coleção';

comment on
column colecao.tx_descricao is 'Descrição da coleção';

comment on
column colecao.dt_coleta is 'Data da coleção';

create table
    endereco (id serial not null,
    tx_rua varchar(60) not null,
    tx_numeroRua numeric(5,
    0),
    cd_bairro int4 not null,
    primary key (id));

comment on
column endereco.tx_rua is 'Nome da rua';

comment on
column endereco.tx_numeroRua is 'Número da rua';

create table
    estado (id serial not null,
    tx_nome varchar(30) not null unique,
    tx_sigla char(2) not null unique,
    cd_pais int4 not null,
    primary key (id));

```

```
comment on
column estado.tx_nome is 'Nome do estado';

comment on
column estado.tx_sigla is 'Sigla do estado';

create table
    fabricante (id serial not null,
    tx_nome varchar(50),
    tx_biografia varchar(260) not null,
    primary key (id));

create table
    imagem (id serial not null,
    tx_link varchar(200) not null unique,
    cd_item int4 not null,
    primary key (id));

comment on
column imagem.tx_link is 'Link, url da imagem';

create table
    item (id serial not null,
    tx_nome varchar(50) not null,
    tx_origem varchar(20) not null,
    tx_descricao varchar(255),
    tp_exposicao timestamp(12),
    dt_fabricacao date,
    vl_valor numeric(19,
    3),
    fl_manutencao bool default 'false' not null,
    dt_cadastro date,
    cd_responsavel int4,
    cd_pais int4,
    cd_setor int4 not null,
    cd_fabricante int4 not null,
    cd_material int4 not null,
    cd_dono int4 not null,
    primary key (id));

comment on
column item.tx_nome is 'Nome do item';

comment on
column item.tx_origem is 'Origem do item';

comment on
column item.tx_descricao is 'Descrição do item';

comment on
column item.tp_exposicao is 'Tempo em que o item pode ficar em manutenção';

comment on
column item.dt_fabricacao is 'Data em que foi fabricado';

comment on
column item.vl_valor is 'Valor do item';

comment on
```

```
column item.fl_manutencao is 'Flag para validar se o item está em exibição';
comment on
column item.dtCadastro is 'Data de cadastro do item';

create table
    item_colecao (cd_item int4 not null,
    cd_colecao int4 not null,
    primary key (cd_item,
    cd_colecao));

create table
    museu (id serial not null,
    tx_nome varchar(50) not null,
    tx_nomeSocial varchar(50),
    tx_cnpj varchar(15) not null,
    cd_endereco int4 not null,
    primary key (id));

comment on
column museu.tx_nome is 'Nome do museu';

comment on
column museu.tx_nomeSocial is 'Nome social do museu';

comment on
column museu.tx_cnpj is 'CNPJ do museu';

create table
    pais (id serial not null,
    tx_nome varchar(50) not null unique,
    primary key (id));

comment on
column pais.tx_nome is 'Nome da pessoa';

create table
    pessoa (id serial not null,
    tx_nome varchar(50),
    tx_cpf varchar(12) not null unique,
    dt_nascimento date not null,
    tx_telefone varchar(11),
    tx_sexo char(1),
    cd_tipoPessoa int4 not null,
    cd_complemento int4 not null,
    primary key (id));

comment on
column pessoa.tx_nome is 'Nome da pessoa';

comment on
column pessoa.tx_cpf is 'CPF da pessoa';

comment on
column pessoa.dt_nascimento is 'Data de nascimento';

comment on
column pessoa.tx_telefone is 'Telefone da pessoa';
```

```

comment on
column pessoa.tx_sexo is 'Flag para o sexo da pessoa F ou M';

create table
    setor (id serial not null,
    tx_nome varchar(50) not null unique,
    tx_descricao varchar(255),
    nr_maxPessoas numeric(19,
    0) not null,
    cd_museu int4 not null,
    primary key (id));

comment on
column setor.tx_nome is 'Nome do setor';

comment on
column setor.tx_descricao is 'Descricao do setor';

comment on
column setor.nr_maxPessoas is 'Numero máximo de pessoas que podem entrar no setor';

create table
    tipoMaterial (id serial not null,
    tx_material varchar(100),
    primary key (id));

comment on
column tipoMaterial.tx_material is 'Tipo do material do item';

create table
    tipoPessoa (id serial not null,
    tipo varchar(15) not null unique,
    primary key (id));

comment on
column tipoPessoa.tipo is 'Tipo de pessoa.
Para validar se é funcionario, visitante, etc.';

create table
    usuario (id serial not null,
    tx_login varchar(20) not null unique,
    tx_senha varchar(20) not null,
    fl_ativo bool default 'true' not null,
    cd_pessoa int4 not null,
    primary key (id));

comment on
column usuario.tx_login is 'Login do usuario';

comment on
column usuario.tx_senha is 'Senha do usuario';

comment on
column usuario.fl_ativo is 'Flag para validar se o usuario está ativo';

create table
    video (id serial not null,
    tx_link varchar(200) not null unique,
    cd_item int4 not null,
    primary key (id));

```

```
primary key (id));

comment on
column video.tx_link is 'Link do video';

create table
Visitante (id serial not null,
dt_visita date,
pessoaid int4 not null,
primary key (id));

comment on
column Visitante.dt_visita is 'Data de visita ao museu';

alter table
endereco add constraint FKendereco538192 foreign key (cd_bairro) references
bairro (id);

alter table
Visitante add constraint FKVisitante776277 foreign key (pessoaid) references
pessoa (id);

alter table
video add constraint FKvideo893307 foreign key (cd_item) references item (id);

alter table
item add constraint FKitem150895 foreign key (cd_dono) references pessoa (id);

alter table
item_colecao add constraint FKItem_colec189670 foreign key (cd_colecao)
references colecao (id);

alter table
item_colecao add constraint FKItem_colec969352 foreign key (cd_item) references
item (id);

alter table
item add constraint FKItem762432 foreign key (cd_material) references tipoMaterial
(id);

alter table
museu add constraint FKmuseu165157 foreign key (cd_endereco) references
endereco (id);

alter table
item add constraint FKItem235442 foreign key (cd_fabricante) references fabricante
(id);

alter table
pessoa add constraint FKpessoa916393 foreign key (cd_complemento) references
endereco (id);

alter table
imagem add constraint FKimagem561566 foreign key (cd_item) references item
(id);

alter table
colecao add constraint FKcolecao717444 foreign key (cd_setor) references setor
(id);
```

```

alter table
    item add constraint FKitem709999 foreign key (cd_setor) references setor (id);

alter table
    setor add constraint FKsetor755313 foreign key (cd_museu) references museu (id);

alter table
    item add constraint FKitem313911 foreign key (cd_pais) references pais (id);

alter table
    estado add constraint FKestado596102 foreign key (cd_pais) references pais (id);

alter table
    item add constraint FKitem223997 foreign key (cd_responsavel) references pessoa
(id);

alter table
    pessoa add constraint FKpessoa206129 foreign key (cd_tipopessoa) references
tipopessoa (id);

alter table
    usuario add constraint FKusuario265371 foreign key (cd_pessoa) references
pessoa (id);

alter table
    bairro add constraint FKbairro278341 foreign key (cd_cidade) references cidade
(id);

alter table
    cidade add constraint FKcidade486232 foreign key (cd_estado) references estado
(id);

```

Fonte: Autores

Script 3 – Índices

```

CREATE INDEX pessoa_tipo_index ON pessoa (cd_tipopessoa);

CREATE INDEX item_material_index ON item (cd_material);

CREATE INDEX item_fabricante_index ON item (cd_fabricante);

CREATE INDEX item_pais_index ON item (cd_pais);

CREATE INDEX item_colecao_index ON item_colecao (cd_colecao);

```

Fonte: Autores

Script 4 – Comandos de Consulta previstos no modelo de negócio

```

SELECT i.tx_nome, i.dt_fabricacao,f.tx_nome FROM item i
INNER JOIN fabricante f ON i.cd_fabricante = f.id
WHERE i.dtCadastro >= '2018-01-01' AND i.fl_manutencao
ORDER BY i.tx_nome DESC

```

```

SELECT p.id, p.tx_nome FROM visitante v
INNER JOIN pessoa p ON v.pessoaid = p.id
INNER JOIN endereco ON endereco.id = p.cd_complemento
INNER JOIN bairro ON bairro.id = endereco.cd_bairro
INNER JOIN cidade ON cidade.id = bairro.cd_cidade

```

```

INNER JOIN estado ON estado.id = cidade.cd_estado
WHERE v.dt_visita BETWEEN '2019-05-01' AND '2019-06-30' AND estado.id != 24
ORDER BY p.tx_nome ASC

SELECT t.id, t.tx_material, COUNT(t.id) AS qtde FROM tipomaterial t
INNER JOIN item i ON t.id = i.cd_material
GROUP BY t.id, t.tx_material
HAVING COUNT(t.id) % 5 = 0
ORDER BY qtde

SELECT c.tx_nome as nome, c.dt_colecao as fabricado, COUNT(i.vl_valor) FROM colecao
c
INNER JOIN item_colecao ic ON c.id = ic.cd_colecao
RIGHT JOIN item i ON i.id IN(ic.cd_item)
WHERE c.tx_nome NOTNULL
GROUP BY nome, fabricado

```

Fonte: Autores

2.2.4 Script para deletar tabelas e restrições

```

alter table
    Visitante drop constraint FKVisitante776277;

alter table
    video drop constraint FKvideo893307;

alter table
    item drop constraint FKitem150895;

alter table
    item_colecao drop constraint FKItem_colec189670;

alter table
    item_colecao drop constraint FKItem_colec969352;

alter table
    item drop constraint FKItem762432;

alter table
    museu drop constraint FKmuseu165157;

alter table
    item drop constraint FKItem235442;

alter table
    endereco drop constraint FKendereco343795;

alter table
    pessoa drop constraint FKpessoa916393;

alter table
    imagem drop constraint FKimagem561566;

```

```
alter table
    colecao drop constraint FKcolecao717444;

alter table
    item drop constraint FKItem709999;

alter table
    setor drop constraint FKsetor755313;

alter table
    item drop constraint FKItem313911;

alter table
    estado drop constraint FKestado596102;

alter table
    item drop constraint FKItem223997;

alter table
    pessoa drop constraint FKpessoa206129;

alter table
    usuario drop constraint FKusuario265371;

alter table
    bairro drop constraint FKbairro278341;

alter table
    cidade drop constraint FKcidade486232;

drop table
    if exists bairro cascade;

drop table
    if exists cidade cascade;

drop table
    if exists colecao cascade;

drop table
    if exists endereco cascade;

drop table
    if exists estado cascade;

drop table
    if exists fabricante cascade;

drop table
    if exists imagem cascade;

drop table
    if exists item cascade;

drop table
    if exists item_colecao cascade;

drop table
    if exists museu cascade;
```

```
drop table
    if exists pais cascade;

drop table
    if exists pessoa cascade;

drop table
    if exists setor cascade;

drop table
    if exists tipoMaterial cascade;

drop table
    if exists tipoPessoa cascade;

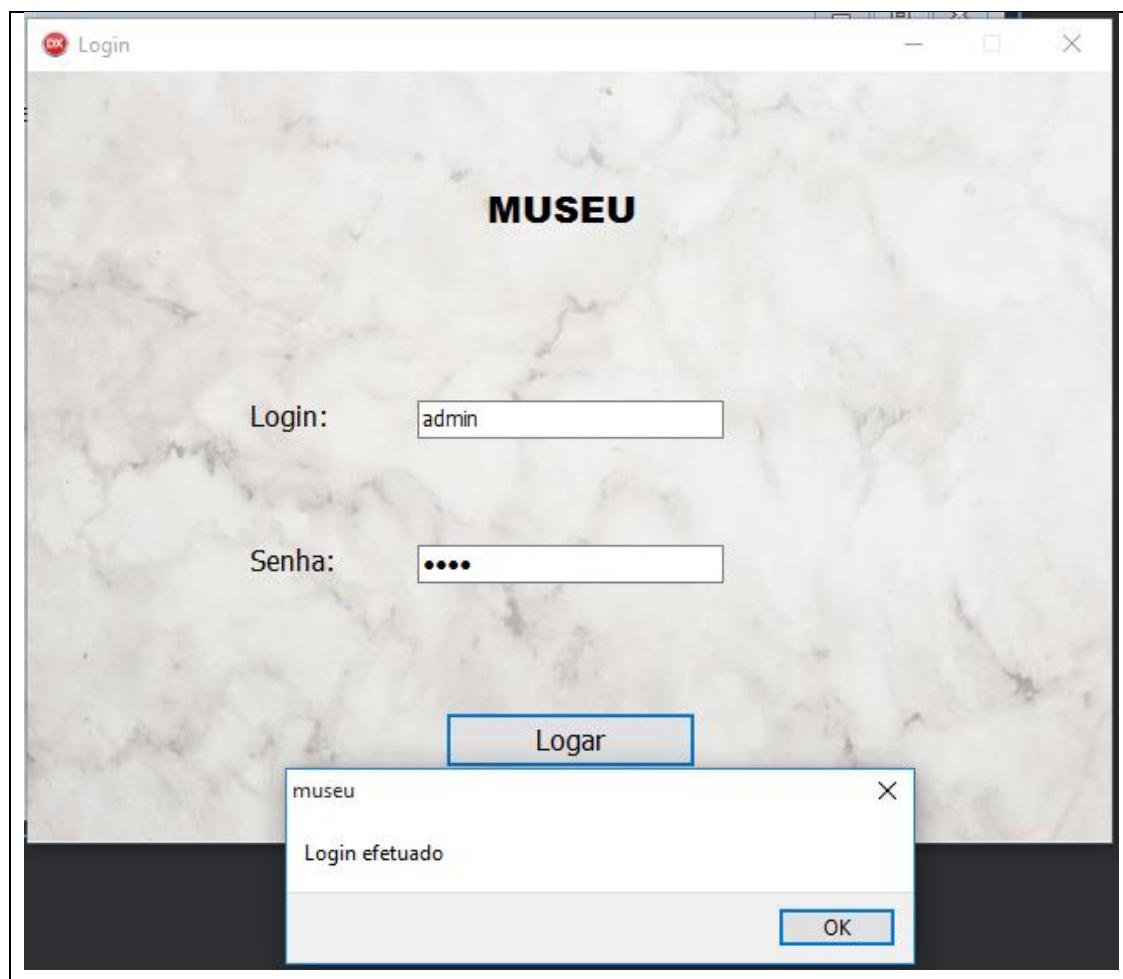
drop table
    if exists usuario cascade;

drop table
    if exists video cascade;

drop table
    if exists Visitante cascade;
```

2.3 PROGRAMAÇÃO III

Tela 1 – Login



Fonte: Autores

Tela 2 – Tela Principal



Fonte: Autores

Tela 3 – Tela Cadastro

Pessoa:

| Código | Nome | CPF | Data Nasc. | Fone | Cód | Tipo Pessoa | Ende |
|--------|-----------|--------------|------------|-------------|-----|-------------|------|
| 1 | Lucas | 12345678912 | 01/01/1995 | 4936641234 | | 1 | |
| 6 | wsfsadsa' | 111-111-111- | 14/06/1995 | (11)1 1111- | | 1 | |

Buttons:

- Novo
- Salvar
- Editar
- Excluir

Fonte: Autores

Tela 3 – Tela Cadastro

Fonte: Autores

3 CONCLUSÃO

Na criação deste projeto final, tivemos uma experiência controlada dos passos a serem seguidos na criação de um software, desde a análise através de metodologias de Engenharia de Software, a criação do *Backend/Banco de Dados* ao Frontend/Programação, assim como cada campo interage para com os demais, desta forma, podemos concluir a importância de cada disciplina na edificação de um software.

REFERÊNCIAS

CONNOLLY, Thomas; BEGG, Carolyn. **Database Systems**: a practical approach to design, implementation, and management. 4 ed. England: Pearson Education Limited, 2005.

A Essencialidade da Engenharia de Software. Disponível em:
[<https://www.devmedia.com.br/a-essencialidade-da-engenharia-de-software/24833>](https://www.devmedia.com.br/a-essencialidade-da-engenharia-de-software/24833).
Acesso em: 25 nov 2019.

APENDICE A – Dicionário de Dados

| Entity Name | Entity Description | Column Name | Column Description | Data Type | Length | Primary Key | Nullable | Unique |
|-------------|--------------------|--------------|----------------------|-----------|--------|-------------|----------|--------|
| bairro | | cd_bairro | | int4 | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | false |
| | | tx_nome | Nome do bairro | varchar | 50 | false | true | false |
| cidade | | cd_cidade | | int4 | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | true |
| | | tx_nome | Nome da cidade | varchar | 50 | false | false | false |
| colecao | | cd_col | | int4 | 0 | false | false | false |
| | | dt_col | Data da coleção | date | 0 | false | true | false |
| | | id | | int4 | 0 | true | false | true |
| | | tx_descricao | Descrição da coleção | varchar | 255 | false | true | false |
| endereco | | cd_endereco | | int4 | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | false |
| | | tx_numeroRua | Número da rua | numero | 5 | false | true | false |
| | | tx_rua | Nome da rua | varchar | 60 | false | false | false |
| estado | | cd_pais | | int4 | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | true |
| | | tx_nome | Nome do estado | varchar | 30 | false | false | true |
| | | tx_sigla | Sigla do estado | char | 2 | false | false | true |

| Entity Name | Entity Description | Column Name | Column Description | Data Type | Length | Primary Key | Nullable | Unique |
|-------------|--------------------|------------------|--|-----------|--------|-------------|----------|--------|
| fabricante | | Id | | int4 | 0 | true | false | false |
| | | tx_biotecnologia | | varchar | 250 | false | false | false |
| | | tx_nome | | varchar | 50 | false | true | false |
| imagem | | cod_item | | int4 | 0 | false | false | false |
| | | Id | | int4 | 0 | true | false | true |
| | | tx_link | Link, url da imagem | varchar | 200 | false | false | true |
| item | | cod_descricao | | int4 | 0 | false | false | false |
| | | cod_fabricante | | int4 | 0 | false | false | false |
| | | cod_material | | int4 | 0 | false | false | false |
| | | cod_pais | | int4 | 0 | false | true | false |
| | | cod_responsavel | | int4 | 0 | false | true | false |
| | | cod_setor | | int4 | 0 | false | false | false |
| | | dt_cadastro | Data de cadastro do item | date | 0 | false | true | false |
| | | dt_fabricacao | Data em que foi fabricado | date | 0 | false | true | false |
| | | fl_manutencao | Flag para validar se o item está em exibição | bool | 0 | false | false | false |
| | | Id | | int4 | 0 | true | false | true |
| | | tp_exclusao | Tempo em que o item pode ficar em manutenção | timestamp | 12 | false | true | false |

| Entity Name | Entity Description | Column Name | Column Description | Data Type | Length | Primary Key | Nullable | Unique |
|--------------|--------------------|----------------|----------------------|-----------|--------|-------------|----------|--------|
| | | cd_itens | Int4 | 0 | false | false | false | |
| | | tx_descricao | Descrição do item | varchar | 255 | false | true | false |
| | | tx_nome | Nome do item | varchar | 50 | false | false | false |
| | | tx_origem | Origem do item | varchar | 20 | false | false | false |
| | | vl_valor | Valor do item | numeric | 10,3 | false | true | false |
| item_colecao | | cd_collecao | Int4 | 0 | true | false | false | |
| item_colecao | | cd_item | Int4 | 0 | true | false | false | |
| museu | | cd_endereco | Int4 | 0 | false | false | false | |
| museu | | id | Int4 | 0 | true | false | true | |
| museu | | tx_cnpj | CNPJ do museu | varchar | 15 | false | false | false |
| museu | | tx_nome | Nome do museu | varchar | 50 | false | false | false |
| museu | | tx_nomesocial | Nome social do museu | varchar | 50 | false | true | false |
| pais | | id | Int4 | 0 | true | false | true | |
| pais | | tx_nome | Nome da pessoa | varchar | 50 | false | false | true |
| pessoa | | cd_complemento | Int4 | 0 | false | false | false | |
| pessoa | | cd_tipoPessoa | Int4 | 0 | false | false | false | |
| pessoa | | dt_nascimento | Data de nascimento | date | 0 | false | false | false |
| pessoa | | id | Int4 | 0 | true | false | true | |
| pessoa | | tx_cpf | CPF da pessoa | varchar | 12 | false | false | true |

| Entity Name | Entity Description | Column Name | Column Description | Data Type | Length | Primary Key | Nullable | Unique |
|--------------|--------------------|-------------------|--|-----------|--------|-------------|----------|--------|
| | | tx_cd_complemento | | int4 | 0 | false | false | false |
| | | tx_nome | Nome da pessoa | varchar | 50 | false | true | false |
| | | tx_sexo | Flag para o sexo da pessoa F ou M | char | 1 | false | true | false |
| | | tx_telfone | Telefone da pessoa | varchar | 11 | false | true | false |
| setor | | tx_cd_museu | | int4 | 0 | false | false | false |
| setor | | id | | int4 | 0 | true | false | true |
| setor | | nr_max_Pessoas | Numero máximo de pessoas que podem entrar no setor | numeric | 19 | false | false | false |
| setor | | tx_descricao | Descrição do setor | varchar | 255 | false | true | false |
| setor | | tx_nome | Nome do setor | varchar | 50 | false | false | true |
| tipoMaterial | | id | | int4 | 0 | true | false | false |
| tipoMaterial | | tx_material | Tipo do material do item | varchar | 100 | false | true | false |
| tipoPessoa | | id | | int4 | 0 | true | false | true |
| tipoPessoa | | tipo | Tipo de pessoa. Para validar se é funcionário, visitante, etc. | varchar | 15 | false | false | true |

| Entity Name | Entity Description | Column Name | Column Description | Data Type | Length | Primary Key | Nullable | Unique |
|-------------|--------------------|-------------|---|-----------|--------|-------------|----------|--------|
| usuario | | cd_pessoa | | int4 | 0 | false | false | false |
| | | f_ativo | Flag para validar se o usuário está ativo | bool | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | true |
| | | tx_login | Login do usuário | varchar | 20 | false | false | true |
| | | tx_senha | Senha do usuário | varchar | 20 | false | false | false |
| video | | cd_item | | int4 | 0 | false | false | false |
| | | id | | int4 | 0 | true | false | false |
| | | tx_link | Link do video | varchar | 200 | false | false | true |
| Visitante | | dt_visita | Data de visita ao museu | date | 0 | false | true | false |
| | | id | | int4 | 0 | true | false | false |
| | | tpessoa_id | | int4 | 0 | false | false | false |