



UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC CAMPUS DE SÃO MIGUEL
DO OESTE

Anderson Luiz Trampusch

Luis Justino

Thiago

SPOTED

SÃO MIGUEL DO OESTE – SC

2019

Anderson Luiz Trampusch

Luis Henrique Justino

Thiago Favretto

SPOTED

Trabalho da disciplina de Banco de Dados II e Engenharia de Software I apresentado no curso de Ciência da Computação da Universidade do Oeste de Santa Catarina – Campus São Miguel do Oeste – SC como requisito para a média final.

Professor: Roberson Junior Fernandes Alves e Franciele Petry.

SÃO MIGUEL DO OESTE – SC

2019

ILUSTRAÇÕES

Figura 1: modelo Entidade-Relacionamento

6

SUMÁRIO

INTRODUÇÃO	5
SPOTED	5
LISTA DE REQUISITOS	5
MODELO ENTIDADE-RELACIONAMENTO	6
SCRIPT DO BANCO DE DADOS	7
DESENVILVIMENTO DO APLICATIVO	7
RELATÓRIOS	7
DICIONÁRIO DE DADOS	8
CONCLUSÃO	9

INTRODUÇÃO

A escolha do nosso trabalho veio quando a nós decidimos em fazer um aplicativo parecido com o instagram mas só para alunos da UNOESC.

O projeto consiste na habilidade de cadastrar posts para os outros verem, comentarem e curtirem, os usuários poderão visualizar os posts sem ter uma conta no aplicativo mas para criar posts, comentarem ou curtirem é necessário se cadastra no aplicativo, os usuários também poderão excluir os posts e comentários postados por eles.

SPOTED

O nosso projeto tem como objetivo principal cadastrar posts feitos por usuários cadastrados no aplicativo

LISTA DE REQUISITOS

Inicialmente foram levantados os requisitos sobre o que o sistema deveria atender, dos principais aos secundários.

1	O sistema deve permitir a inclusão, alteração e remoção de usuários do sistema.
2	O sistema deve permitir aos usuários removerem apenas seus próprios cadastros.
3	O sistema deve permitir o cadastro de apenas um usuário por Email.
4	O sistema deve emitir mensagens de erro caso algum dos dados estejam incompletos.
5	O sistema deve gerar um relatório com todos os usuários cadastrados.
6	O sistema deve gerar um relatório com todas os post cadastradas.
7	O sistema deve mostrar todos os post anônimos.
8	O sistema deve mostrar permitir comentar quantas vezes que quiser um post.
9	O sistema deve cadastrar um historico de todos os post cadastrados.

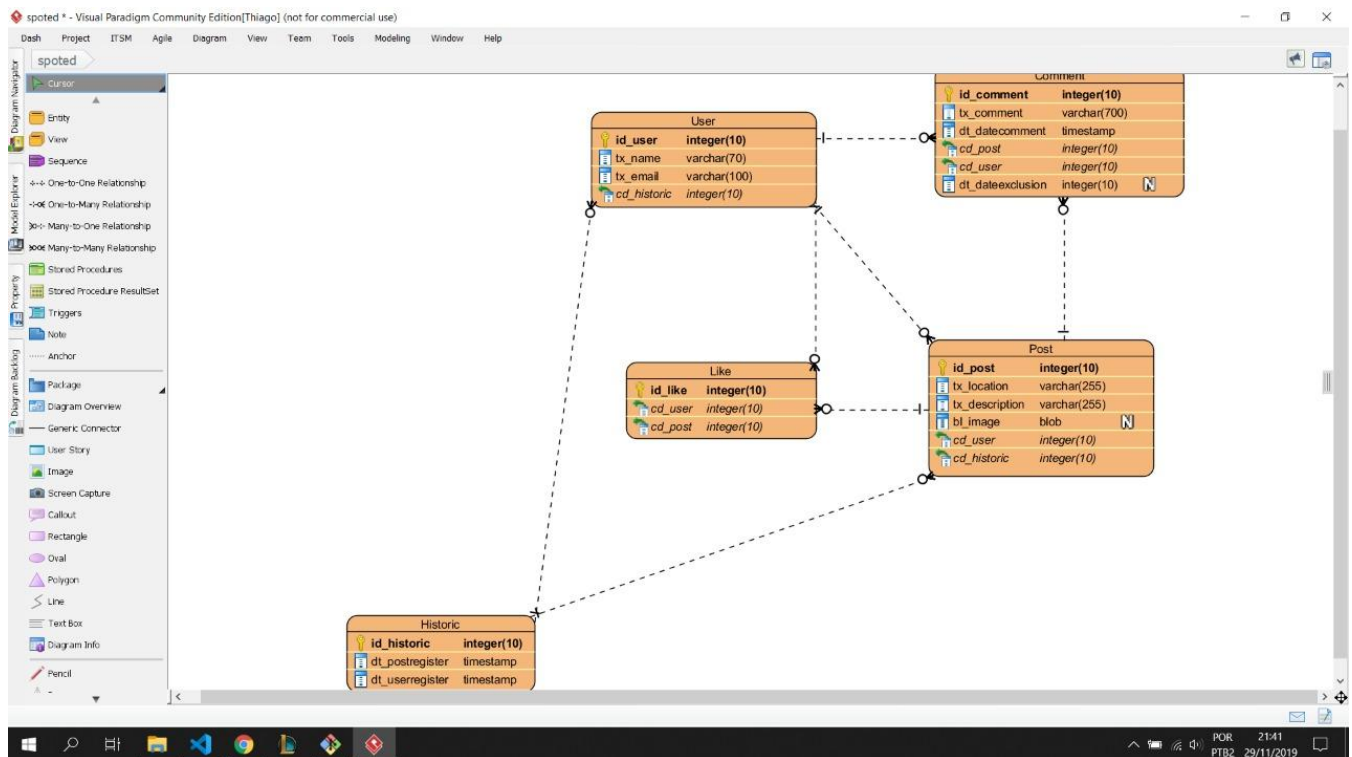
10	O sistema de permitir o usuário dar likes somente uma vez em cada post.
11	O sistema deve permitir o usuário retirar o like de uma post.
12	O sistema deve permitir o usuário retirar seu comentário no post.
13	O sistema deve permitir o usuário cadastrar um novo post.
14	O sistema deve permitir o usuário editar um novo post.
15	O sistema deve permitir o usuário editar somente a descrição de um post.
16	O sistema deve permitir o usuário editar somente a localização de um post.
17	O sistema deve permitir o usuário editar somente a imagem de um post.
18	O sistema deve permitir o usuário editar somente o título de um post.

MODELO ENTIDADE-RELACIONAMENTO

Com a análise dos requisitos foi desenvolvido o modelo entidade-relacionamento utilizando a ferramenta visual paradigm. O modelo possui 5 tabelas, todas elas comentadas e com a descrição de cada tabela para ajudar na compreensão do modelo.

O modelo construído é o seguinte:

Figura 1: modelo Entidade-Relacionamento



SCRIPT DO BANCO DE DADOS

Com o modelo criado foi desenvolvidos os scripts para a criação do banco utilizando o mongoDB, segue em anexo o backup do banco de dados, nomeado como BACKUPBANCO.

DESENVOLVIMENTO DO APLICATIVO

Para o desenvolvimento foi utilizado Node.js, mongoDB e reactjs, segue abaixo o aplicativo desenvolvido.

Como primeira página temos a página de cadastro, onde podemos colocar o título localização e descrição da imagem.



← *Spoted*

Compartilhar

Local da foto

Descrição

Selecionar imagen

Ainda na primeira página podemos realizar o upload da imagem.



← *Spoted*

Compartilhar

sao miguel

Descrição

Selecionar imagen



Na segunda página temos a listagem de post onde se encontra todos os post.



Spoted



sao miguel
alguem sabe o nome desse cara?



4 Curtidas

Na terceira caso o usuário queira ver mais sobre os post cadastrado ele pode clicar que irá abrir com mais informações.



Spoted



sao miguel
alguem sabe o nome
desse cara?

4 Curtidas



Comentários

user

é carlos sim

user

carlos, eu acho



RELATÓRIOS

Os relatórios foram definidos pelo professor baseado no modelo do projeto, os scripts dos relatórios estão a seguir:

- 1 - Relacione o nome e e-mail de usuários com uma postagem no mês de maio de 2019. Ordene o relatório pelo nome de forma descendente.

```
select public.user.tx_nome, public.user.tx_email from public.user inner join public.post on  
public.post.cd_user = public.user.id_user order by tx_nome desc;
```

- 2- Relacione o nome do usuário, o código do post e a quantidade de likes de posts feitos por usuários cujo email termine em 'com.br'. Ordene o relatório do post com mais likes para o post com menos likes.

```
select public.user.tx_nome, public.post.id_post, public.post.nr_like from public.user
inner join public.post on public.post.cd_user = public.user.id_user where
public.user.tx_nome like '%.com.br' order by nr_like desc;
```

- 3- Relacione o código do post, nome do post e somente o último comentário de todos os posts feitos em 2019. Posts sem comentários devem ser relacionados também. Ordene o relatório do post mais antigo para o post mais recente

```
select public.user.tx_nome, public.post.id_post, (SELECT * FROM public."coment"
where public."coment".dt_cadastro BETWEEN '2019-01-01' and '2019-12-31' order by
dt_cadastro desc LIMIT 1) from public.user inner join public.post on
public.post.cd_user = public.user.id_user where public.user.tx_nome like '%.com.br'
order by dt_cadastro asc;
```

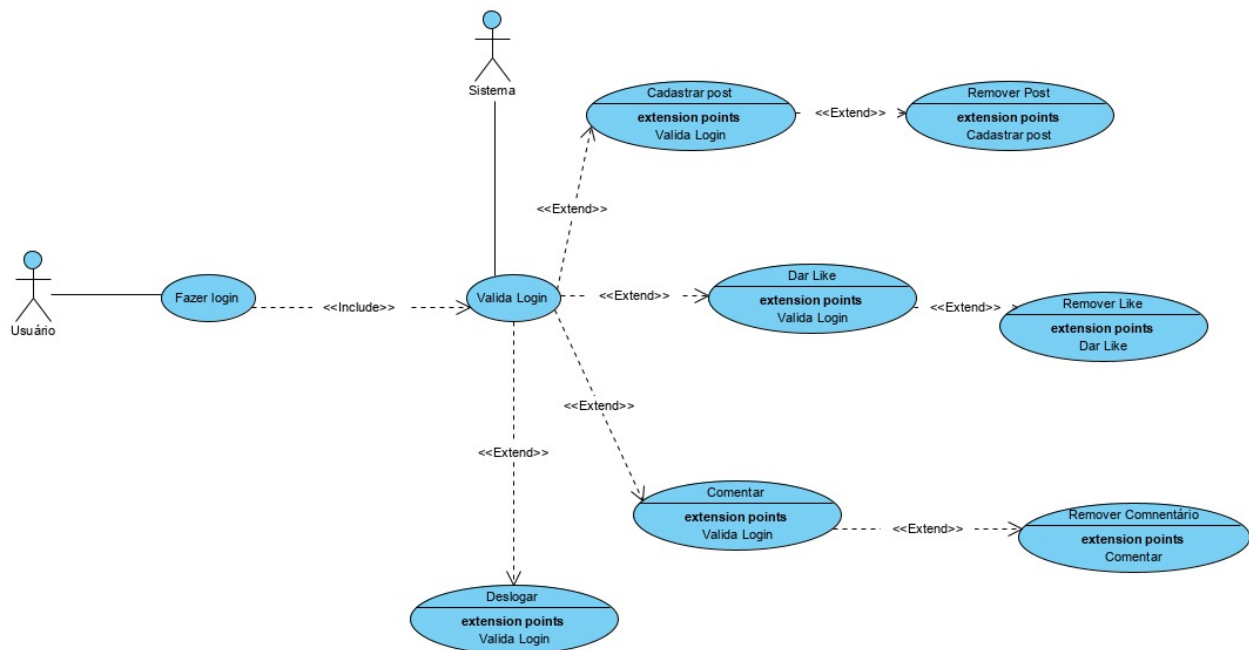
- 4- Relacione o código do usuário, nome do usuário e descrição do post para posts realizados em São Miguel do Oeste antes de 2020 e que possuem mais de 100 likes. Ordene o relatório pela descrição do post de forma descendente.

```
select public.user.tx_nome, public.user.id_user, public.post.tx_descricao from
public.user inner join public.post on public.post.cd_user = public.user.id_user inner
join public.historico on public.post.cd_user = public.user.id_user where
public.post.tx_localizacao like 'São Miguel do Oeste' and public.post.nr_like > 100 and
public.historico.dt_cadastro < '2020-01-01 00:00:00' order by tx_descricao desc;
```

DIAGRAMAS

Para facilitar o desenvolvimento do aplicativo foi criado diagramas, segue abaixo os diagramas usados para realização do aplicativo.

Diagrama de uso



Fluxo de caso

Descrição: Usuário usa uma conta para entrar no aplicativo, lá ele poderá ver posts dar like nos posts, comentar e postar suas próprias.

Fluxo Básico:

Usuário cria uma conta no aplicativo

Usuário insere o e-mail e senha para cadastro

Após o cadastro o usuário entra no aplicativo com o e-mail e a senha cadastrada

Usuário faz um post no aplicativo

Usuário faz um comentário em um outro post

Usuário dá um gostei em um post

Usuário deleta o post que ele criou

Usuário sai do aplicativo

Fluxo Alternativo 1:

Usuário cria uma conta no aplicativo

Usuário insere o e-mail e senha para cadastro

Após o cadastro o usuário entra no aplicativo com o e-mail e a senha cadastrada

Usuário faz um post no aplicativo

Usuário sai do aplicativo

Fluxo Alternativo 2:

Usuário cria uma conta no aplicativo

Usuário insere o e-mail e senha para cadastro

Após o cadastro o usuário entra no aplicativo com o e-mail e a senha cadastrada

Usuário faz um post no aplicativo

Usuário faz um post no aplicativo

Usuário faz um comentário em um outro post

Usuário sai do aplicativo

Fluxo Alternativo 3:

Usuário cria uma conta no aplicativo

Usuário insere o e-mail e senha para cadastro

Após cadastro o usuário tenta entrar no aplicativo e recebe mensagem de conta desativada

Diagrama de sequência

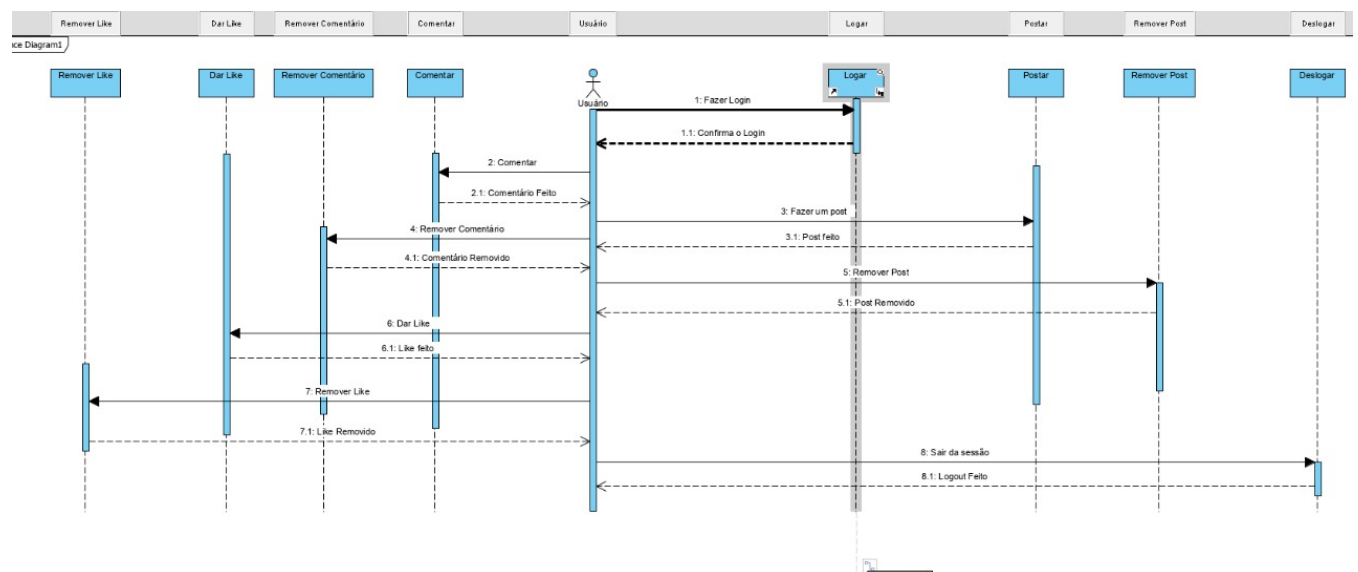


Diagrama de atividade.

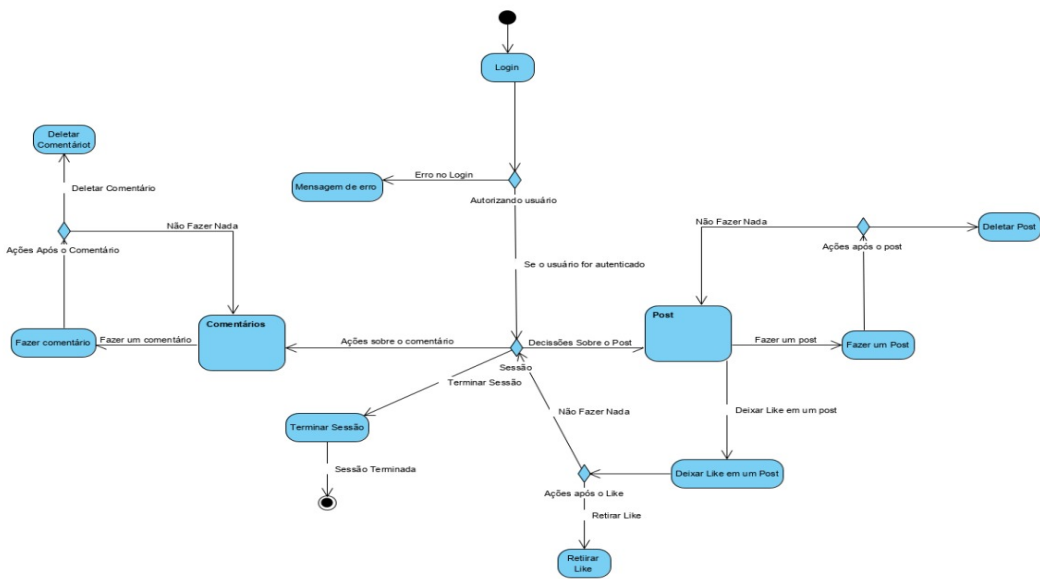


Diagrama de estados

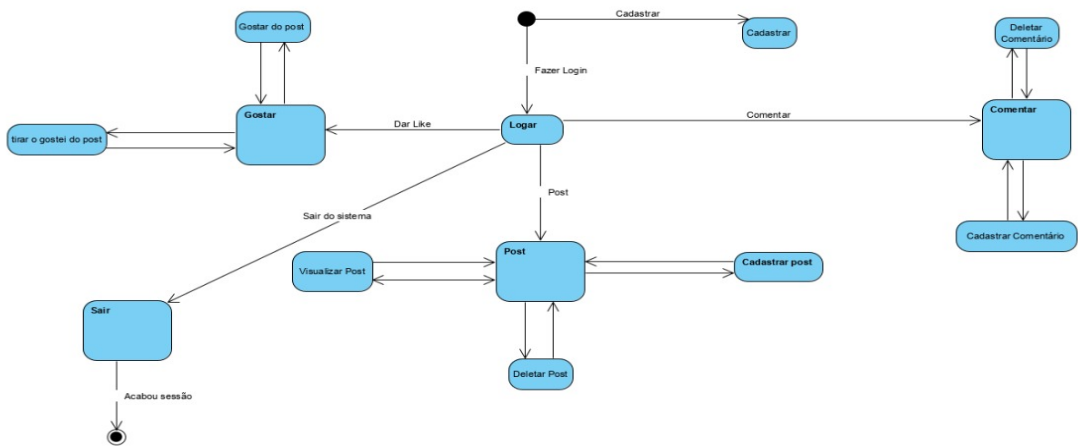
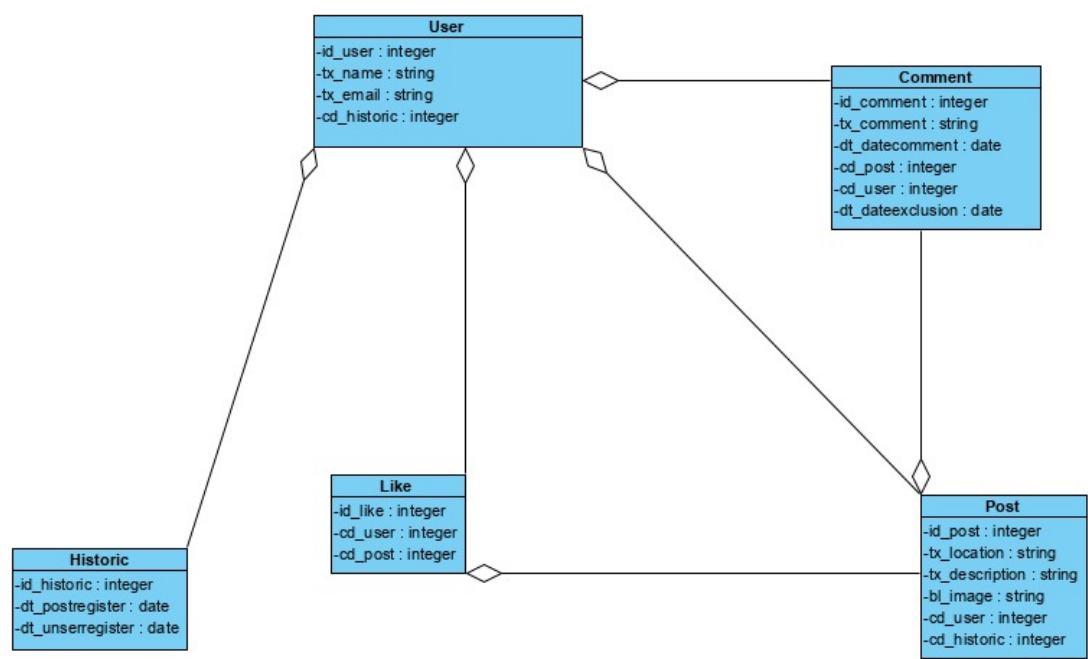




Diagrama de classes



DICIONÁRIO DE DADOS

Entity Name		Entity Description					
	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
Comment							
	cd_post	codigo do post	integer	10	false	false	false
	cd_user	codigo do usuário	integer	10	false	false	false
	dt_datecomment	dia do comentario	timestam p	0	false	false	false
	dt_dateexclusion	data de exclusão do comentario	integer	10	false	true	false
	id_comment	id do comentário	integer	10	true	false	false
	tx_comment	comentario do post	varchar	700	false	false	false
Historic							
	dt_postregister	data de cadastro do post	timestam p	0	false	false	false
	dt_userregister	data de cadastro do usuário	timestam p	0	false	false	false
	id_historic	id do historico	integer	10	true	false	false
Like							
	cd_post	codigo do post	integer	10	false	false	false

	cd_user	codigo do usuário	integer	10	false	false	false
	id_like	id do like	integer	10	true	false	false
 Post							
	bl_image	imagem do post	blob	0	false	true	false
	cd_historic	codigo do historico	integer	10	false	false	false
	cd_user	codigo do usuário	integer	10	false	false	false
	id_post	id do post	integer	10	true	false	false
	tx_description	descrição do post	varchar	255	false	false	false
	tx_location	local do post	varchar	255	false	false	false
 User							
	cd_historic	codigo do historico	integer	10	false	false	false
	id_user	id do usuario	integer	10	true	false	false
	tx_email	email do usuário	varchar	100	false	false	false
	tx_name	nome do usuário	varchar	70	false	false	false

CONCLUSÃO

O projeto ainda está pronto, podendo cadastrar usuários, posts, comentários e likes,

Com o resultado desse projeto a gente consegue fazer o que nos queríamos no começo e estamos melhorando o projeto todo dia deixando-o cada vez melhor.