**IFSP Instituto Federal de Educação Ciência e Tecnologia**

**Análise e Desenvolvimento de Sistemas**

**Mauro Sérgio Jircik A. M. R. L. Pereira BP3032116**

**Pedro Henrique Nishimura Bachaalani BP3032884**

**Victor Gomes da Cruz BP303285X**

**“projeto interdisciplinaR”**

**BANCO DE DADOS**

bragança paulista

2023

**Mauro Sérgio Jircik ARRUDA MENDES RIBEIRO LEITE Pereira**

**PEDRO HENRIQUE NISHIMURA BACHAALANI**

**VICTOR GOMES DA CRUZ**

**projeto interdisciplinaR**

**BANCO DE DADOS**

Trabalho apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista, como requisito parcial da disciplina Banco de Dados I sob a orientação da Profa. Letícia Souza Netto Brandi.

bragança paulista

2023

**RESUMO**

O presente trabalho tem como objetivo desenvolver e implementar um aplicativo visando atender as fases de desenvolvimento de um software aplicando os conteúdos abordados.

**Palavras-chave:** Banco de dados, Universo de discurso, Diagrama entidade relacionamento, Notação textual, Script, MySQL

**SUMÁRIO**

|  |  |
| --- | --- |
| **RESUMO ........................................................................................................................................** | **2** |
|  |  |
| **1 INTRODUÇÃO ...................................................................................................................** | **4** |
| **1.1 Contextualização .................................................................................................................** | **4** |
|  |  |
| **2 DESENVOLVIMENTO .....................................................................................................** | **4** |
| **2.1 Universo do discurso ...........................................................................................................** | **4** |
| **2.2 Diagrama entidade relacionamento ..................................................................................** | **5** |
| **2.3 Notação textual ....................................................................................................................** | **5** |
| **2.4 Script para implementação ................................................................................................** | **6** |
| **2.5 Consultas com queries ........................................................................................................** | **10** |
|  |  |
| **3 CONCLUSÃO .....................................................................................................................** | **12** |
|  |  |
| **REFERÊNCIAS ..................................................................................................................** | **13** |
|  |  |

**1 INTRODUÇÃO**

**1.1 Contextualização**

A loja de aplicativos identificou a necessidade de desenvolver um aplicativo de mensagens para atender às demandas de comunicação dos usuários.

Atualmente, existem várias opções de aplicativos de mensagens disponíveis, no entanto, deseja-se oferecer aos usuários uma solução própria e exclusiva que seja segura, fácil de usar e independente de plataformas.

O objetivo principal do aplicativo de mensagens é permitir que os usuários se comuniquem de forma eficiente, troquem mensagens instantâneas e compartilhem informações de maneira conveniente.

O aplicativo deve ser desenvolvido levando em consideração os principais sistemas operacionais móveis, como iOS, macOS, Windows e Android, para garantir uma ampla compatibilidade e acessibilidade aos usuários.

**2 DESENVOLVIMENTO**

**2.1 Universo de discurso**

O software deve possuir um banco de dados para servir a uma plataforma de comunicação social, onde os usuários podem se cadastrar e interagir uns com os outros com a utilização de um meio conectado à internet podendo ser celular, tablet ou computador.

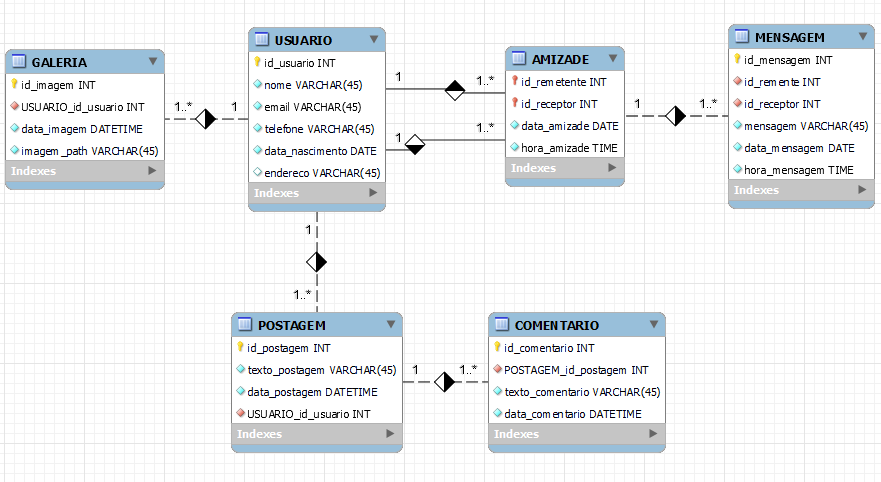
Em outras palavras, uma pessoa, poderá interagir com seus amigos através da utilização do aplicativo de forma unificada e automatizada.

Informações adicionais:

* O usuário pode criar uma conta no software, fornecendo informações como nome, endereço de e-mail, telefone, data de nascimento e endereço;
* Após o cadastro, essas informações são usadas para identificar e autenticar o usuário no sistema;
* O usuário tem a opção de escolher para o receptor da mensagem que será enviada por ele;
* O usuário cadastrado pode enviar mensagens para outro usuário cadastrado por ele através do software;
* O usuário pode enviar mensagens individuais para usuários específicos, selecionando o nome do receptor e redigindo a mensagem;
* O usuário cadastrado pode realizar postagens na plataforma, compartilhando suas ideias, pensamentos, notícias ou qualquer outro conteúdo relevante;
* As postagens podem ser visualizadas por outros usuários cadastrados no software;
* Os usuários podem comentar nas postagens de outros usuários, gerando discussões e interações na plataforma;
* O usuário cadastrado pode disponibilizar imagens para serem vistas por outros usuários;
* O usuário pode fazer upload de imagens para o software;
* Outros usuários cadastrados podem visualizar as imagens.

**2.2 Diagrama entidade relacionamento**

Para essa aplicação, apresenta-se o diagrama entidade relacionamento para o esquema, com a especificação dos atributos de cada entidade e as restrições estruturais de cada tipo de relacionamento.



**2.3 Notação textual**

Do diagrama entidade relacionamento apresentado, obtém-se a seguinte notação textual:

USUARIO (id\_usuario, nome, email, telefone, data\_nasc, endereco)

PRIMARY KEY (id\_usuario)

POSTAGEM (id\_postagem, texto\_postagem, data\_postagem, **id\_usuario**)

PRIMARY KEY (id\_postagem)

FOREIGN KEY (**id\_usuario**) REFENRENCES USUARIO (id\_usuario)

COMENTARIO (id\_comentario, texto\_comentario, data\_comentario, **id\_postagem**)

PRIMARY KEY (id\_comentario)

FOREIGN KEY (**id\_postagem**) REFENRENCES POSTAGEM (id\_postagem)

GALERIA (id\_imagem, imagem\_path, data\_imagem, **id\_usuario**)

PRIMARY KEY (id\_imagem)

FOREIGN KEY (**id\_usuario**) REFENRENCES USUARIO (id\_usuario)

AMIZADE (**id\_remetente**, **id\_receptor**, data\_amizade, hora\_amizade)

FOREIGN KEY (**id\_remetente**) REFENRENCES USUARIO (id\_usuario)

FOREIGN KEY (**id\_receptor**) REFENRENCES USUARIO (id\_usuario)

MENSAGEM (id\_mensagem, mensagem, data\_mensagem, hora\_mensagem, **id\_remetente**, **id\_receptor**)

PRIMARY KEY (id\_mensagem)

FOREIGN KEY (**id\_remetente**) REFENRENCES AMIZADE (id\_remetente)

FOREIGN KEY (**id\_receptor**) REFENRENCES AMIZADE (id\_receptor)

**2.4 Script**

# Criar o banco de dados

create database if not exists MPV;

# Remover o banco de dados

drop database MPV;

# Abrir o banco de dados

use MPV;

# Exibir o banco de dados

show databases;

# Criar a tabela USUÁRIO

create table if not exists USUARIO

(

id\_usuario int not null,

nome varchar(100) not null,

email varchar(100) not null,

telefone varchar(45) not null,

data\_nascimento date,

endereço varchar (100),

primary key (id\_usuario)

);

# Criar a tabela GALERIA

create table if not exists GALERIA

(

id\_imagem int not null,

id\_usuario int,

data\_imagem datetime not null,

imagem\_path varchar(100),

primary key (id\_imagem),

foreign key (id\_usuario) references USUARIO(id\_usuario)

);

# Criar a tabela POSTAGEM

create table if not exists POSTAGEM

(

id\_postagem int not null,

id\_usuario int,

texto\_postagem varchar(150) not null,

data\_postagem datetime not null,

primary key (id\_postagem),

foreign key (id\_usuario) references USUARIO(id\_usuario)

);

# Criar a tabela COMENTARIO

create table if not exists COMENTARIO

(

id\_comentario int not null,

id\_postagem int,

texto\_comentario varchar(150) not null,

data\_comentario datetime,

primary key (id\_comentario),

foreign key (id\_postagem) references POSTAGEM(id\_postagem)

);

# Criar a tabela AMIZADE

create table if not exists AMIZADE

(

id\_remetente int not null,

id\_receptor int not null,

data\_amizade date not null,

hora\_amizade time not null,

foreign key (id\_remetente) references USUARIO(id\_usuario),

foreign key (id\_receptor) references USUARIO(id\_usuario)

);

# Criar a tabela MENSAGEM

create table if not exists MENSAGEM

(

id\_mensagem int not null,

mensagem varchar (100) not null,

data\_mensagem date not null,

hora\_mensagem time not null,

id\_remetente int,

id\_receptor int,

primary key (id\_mensagem),

foreign key (id\_remetente) references AMIZADE(id\_remetente),

foreign key (id\_receptor) references AMIZADE(id\_receptor)

);

# Exibir tabelas

show tables;

# Inserir dentro da tabela USUARIO os valores

INSERT INTO USUARIO VALUES

(1, 'Bruce Wayne','souobatman@gmail.com','11985632545','1972-02-19','Gotham City'),

(2, 'Clark Kent','souosuperman@gmail.com','11963215687','1970-03-21','Metropolis'),

(3, 'Barry Allen','souoflash@gmail.com','1198524678','1992-09-30','Central City'),

(4, 'Diana Prince','souamulhermaravilha@gmail.com','11978451236', '1980-05-12','Themyscira'),

(5, 'Hal Jordan','souolanternaverde@gmail.com','11985245678','1971-11-01', 'Coast City');

# Inserir dentro da tabela GALERIA os valores

INSERT INTO GALERIA VALUES

(1,1,'2023-02-20 15:30:00','/caminho/para/imagem.jpg'),

(2,2,'2023-05-04 10:45:00','/caminho/para/cachorro.png'),

(3,3,'2023-04-15 18:20:00','/caminho/para/foto.jpg'),

(4,4,'2023-03-06 14:15:00','/caminho/para/barco.jpg'),

(5,5,'2023-01-27 09:30:00','/caminho/para/foto2.jpg'),

(6,1,'2023-01-18 16:45:00','/caminho/para/dracula.jpg'),

(7,2,'2023-02-19 11:20:00','/caminho/para/loop.jpg'),

(8,3,'2023-03-30 13:45:00','/caminho/para/casa.jpg'),

(9,4,'2023-05-01 08:30:00','/caminho/para/batcaverna.jpg'),

(10,5,'2023-04-21 17:15:00','/caminho/para/anel.jpg'),

(11,1,'2023-04-12 12:40:00','/caminho/para/capa.jpg'),

(12,2,'2023-03-23 14:25:00','/caminho/para/lua.jpg'),

(13,3,'2023-02-14 09:50:00','/caminho/para/sol.jpg'),

(14,4,'2023-01-05 18:35:00','/caminho/para/energico.jpg'),

(15,5,'2023-05-16 13:00:00','/caminho/para/bala.jpg');

# Inserir dentro da tabela POSTAGEM os valores

INSERT INTO POSTAGEM VALUES

(1,1,'Olá, mundo!','2023-05-24 10:00:00'),

(2,2,'Bom dia!','2023-03-20 09:30:00'),

(3,3,'Boa tarde!','2023-04-22 15:45:00'),

(4,4,'Boa noite!','2023-02-23 20:15:00'),

(5,5,'Ótimo dia!','2023-01-21 11:30:00'),

(6,1,'Tenham um bom final de semana!','2023-05-08 18:00:00'),

(7,2,'Excelente semana para todos!','2023-04-09 09:45:00'),

(8,3,'Aproveitem o feriado!','2023-05-30 14:20:00'),

(9,4,'Boas festas!','2023-01-31 12:00:00'),

(10,5,'Feliz aniversário!','2023-01-11 16:30:00'),

(11,1,'Tenham uma ótima semana!','2023-03-02 09:15:00'),

(12,2,'Aproveitem o dia de hoje!','2023-04-23 14:45:00'),

(13,3,'Vamos celebrar!','2023-05-14 11:30:00'),

(14,4,'Hora de relaxar!','2023-05-05 17:20:00'),

(15,5,'Aproveitem o final de semana!','2023-03-16 13:45:00');

# Inserir dentro da tabela COMENTARIO os valores

INSERT INTO COMENTARIO VALUES

(1,1,'Excelente postagem!','2023-04-23 10:00:00'),

(2,2,'Parabéns pelo conteúdo!','2023-05-04 09:30:00'),

(3,3,'Adorei essa foto!','2023-03-15 15:45:00'),

(4,4,'Muito inspirador!','2023-02-26 20:15:00'),

(5,5,'Que bela paisagem!','2023-05-07 11:30:00'),

(6,1,'Ótimo ponto de vista!','2023-02-18 18:00:00'),

(7,2,'Parabéns pela conquista!','2023-04-29 09:45:00'),

(8,3,'Esse lugar é incrível!','2023-05-03 14:20:00'),

(9,4,'Que mensagem poderosa!','2023-03-01 12:00:00'),

(10,5,'Parabéns pelo talento!','2023-01-11 16:30:00'),

(11,1,'Muito bem explicado!','2023-05-22 09:15:00'),

(12,2,'Que foto maravilhosa!','2023-04-03 14:45:00'),

(13,3,'Parabéns pela iniciativa!','2023-05-04 11:30:00'),

(14,4,'Aproveitem esse momento!','2023-03-15 17:20:00'),

(15,5,'Parabéns pelo sucesso!','2023-06-21 13:45:00');

# Inserir dentro da tabela AMIZADE os valores

INSERT INTO AMIZADE VALUES

(1,2,'2023-04-02','10:00:00'),

(2,3,'2023-05-04','09:30:00'),

(3,4,'2023-03-15','15:45:00'),

(4,5,'2023-04-26','20:15:00'),

(5,1,'2023-02-17','11:30:00'),

(1,3,'2023-01-08','18:00:00'),

(2,4,'2023-04-09','09:45:00'),

(3,5,'2023-05-03','14:20:00'),

(4,1,'2023-02-10','12:00:00'),

(5,2,'2023-04-1','16:30:00'),

(1,4,'2023-05-22','09:15:00'),

(2,5,'2023-02-23','14:45:00'),

(3,1,'2023-01-14','11:30:00'),

(4,2,'2023-03-15','17:20:00'),

(5,3,'2023-02-06','13:45:00');

# Inserir dentro da tabela MENSAGEM os valroes

INSERT INTO MENSAGEM VALUES

(1,'Olá, como você está?', '2023-05-23', '10:00:00',1, 2),

(2,'Tudo bem, e com você?', '2023-05-24', '09:30:00',2, 3),

(3,'Estou ótimo!', '2023-05-25', '15:45:00',3, 4),

(4,'Que bom!', '2023-05-26', '20:15:00',4, 5),

(5,'Vamos sair amanhã?', '2023-05-27', '11:30:00',5, 1),

(6,'Claro, que horas?', '2023-05-28', '18:00:00',1, 3),

(7,'Podemos nos encontrar às 19h.', '2023-05-29', '09:45:00',2, 4),

(8,'Perfeito, nos vemos então.', '2023-05-30', '14:20:00',3, 5),

(9,'Estou ansioso!', '2023-05-31', '12:00:00',4, 1),

(10,'Também estou!', '2023-06-01', '16:30:00',5, 2),

(11,'Até amanhã!', '2023-06-02', '09:15:00',1, 4),

(12,'Nos vemos lá!', '2023-06-03', '14:45:00',2, 5),

(13,'Mal posso esperar!', '2023-06-04', '11:30:00',3, 1),

(14,'Será incrível!', '2023-06-05', '17:20:00',4, 2),

(15,'Com certeza!', '2023-06-06', '13:45:00',5, 3);

# Selecionar as todos os usuários cadastrados:

SELECT \* FROM USUARIO;

SELECT \* FROM GALERIA;

SELECT \* FROM POSTAGEM;

SELECT \* FROM COMENTARIO;

SELECT \* FROM AMIZADE;

SELECT \* FROM MENSAGEM;

**2.5 Consultas com queries**

# 1) Selecione o nome, email e telefone dos usuários com mais de 45 anos:

SELECT nome, email, telefone, round(DATEDIFF(CURDATE(), data\_nascimento)/365, 0) as idade

FROM USUARIO

WHERE DATEDIFF(CURDATE(), data\_nascimento)/365 > 45;

# 2) Selecione as postagens feitas pelos usuários em ordem decrescente de data:

SELECT \* FROM POSTAGEM

ORDER BY data\_postagem DESC;

# 3) Seleciona as postagens feitas pelo usuário de ID 3:

SELECT \* FROM POSTAGEM

WHERE id\_usuario = 3;

# 4) Seleciona as postagens que possuem "dia" em alguma parte do texto:

SELECT \* FROM POSTAGEM

WHERE texto\_postagem LIKE '%dia%';

# 5) Seleciona todas as mensagens enviadas pelo usuário de ID 1 (Bruce Wayne), incluindo a data de envio e o nome do receptor:

SELECT mensagem.id\_remetente, mensagem.mensagem, mensagem.data\_mensagem, usuario.nome as nome\_receptor

FROM MENSAGEM inner join usuario on mensagem.id\_receptor = usuario.id\_usuario

WHERE id\_remetente = 1;

# 6) Seleciona todas as postagens, incluindo do nome de quem postou, ordenadas pela quantidade de caracteres da postagem:

SELECT id\_postagem, usuario.nome as "usuário\_que\_postou", texto\_postagem, length(texto\_postagem) as "qtd\_caracteres"

FROM postagem inner join usuario on postagem.id\_usuario = usuario.id\_usuario

ORDER BY 4;

# 7) Mostra quantos comentários cada postagem teve:

SELECT postagem.id\_postagem, count(comentario.id\_postagem) as "qtd\_comentarios"

FROM postagem inner join comentario on postagem.id\_postagem = comentario.id\_postagem

GROUP BY postagem.id\_postagem;

# 8) Mostra todas as postagens feitas em um Domingo, com as datas no formato brasileiro (dia, mês, ano):

SELECT postagem.id\_postagem, usuario.nome as "usuário\_que\_postou", postagem.texto\_postagem, date\_format(postagem.data\_postagem, "%d/%m/%Y") as data\_postagem, dayname(data\_postagem) as dia

FROM usuario inner join postagem on usuario.id\_usuario = postagem.id\_usuario

WHERE dayname(data\_postagem) = "Sunday";

# 9) Mostra todas as imagens postadas entre Abril e Maio, ordenadas por data de postagem:

SELECT usuario.nome, date\_format(galeria.data\_imagem, "%d/%m/%Y") as data\_imagem, galeria.imagem\_path

FROM usuario inner join galeria on usuario.id\_usuario = galeria.id\_usuario

WHERE month(galeria.data\_imagem) = 4 or month(galeria.data\_imagem) = 5

ORDER BY galeria.data\_imagem;

# 10) Mostra a quantidade média de caracteres das postagens, por usuário:

SELECT postagem.id\_usuario, usuario.nome, round(avg(length(texto\_postagem)), 2) as media\_caracteres\_postagem

FROM postagem inner join usuario on usuario.id\_usuario = postagem.id\_usuario

GROUP BY id\_usuario;

# 11) Mostra as 6 primeiras postagens feitas no aplicativo:

SELECT usuario.nome, texto\_postagem, data\_postagem

FROM usuario inner join postagem on usuario.id\_usuario = postagem.id\_usuario

ORDER BY data\_postagem limit 6;

# 12) Mostra todas as mensagens enviadas entre 12h00 e 18h00 ordenadas por hora:

SELECT usuario.nome, mensagem, hora\_mensagem

FROM mensagem inner join usuario on usuario.id\_usuario = mensagem.id\_remetente

WHERE hora\_mensagem between '12:00:00' and '18:00:00'

ORDER BY hora\_mensagem;

# 13) Comando que altera a tabela POSTAGEM para adicionar a coluna "qtd\_caracteres":

ALTER TABLE POSTAGEM

ADD COLUMN qtd\_caracteres int;

# 14) Comando que preenche os valores da coluna "qtd\_caracteres" na tabela POSTAGEM:

UPDATE postagem

SET qtd\_caracteres=length(texto\_postagem)

WHERE id\_postagem between 1 and 15;

UPDATE postagem

SET qtd\_caracteres=length(texto\_postagem)

WHERE id\_postagem IS NOT NULL;

**3 CONCLUSÃO**

Com o planejamento, o desenvolvimento e a construção do banco de dados, conclui-se que o armazenamento projetado e a organização de dados permitem que os valores sejam categorizados e relacionados, facilitando a sua recuperação para análise.

As consultas complexas, nas análises dos valores durante a estruturação dos dados, permitem a tomada de decisões de maneira adequada e fundamentada, levando a uma maior produtividade e rendimento no trabalho.

Assim , os bancos de dados desempenham um papel fundamental na integração de sistemas e na troca de dados entre diferentes aplicativos e plataformas.

**REFERÊNCIAS**

[1] DATE, C. J. **Introdução a Sistemas de Banco de Dados.** 9ª Ed. Campus, 2000.

[2] ELMASRI, R. & NAVATHE, S. **Sistemas de banco de dados.** 6ª Ed. São Paulo: Pearson Education, 2011.

[3] HEUSER, C. A. **Projeto de Banco de Dados.** 6ª Ed. Porto Alegre: Sagra Luzzatto, 2009.

[4] GUIMARÃES, Célio Cardoso. **Fundamentos de Banco de Dados: modelagem, projeto e linguagem SQL.** Campinas: Unicamp, 2003.

[5] MACHADO, Felipe Nery Rodrigues. **Projeto e Implementação de Banco de Dados.** 2ª Ed. São Paulo: Érica, 2008.

[6] SILBERSHATZ, A., KORTH, H. F., SUDARSHAN, S. **Sistema de Banco de Dados.** 5ª Ed. McGraw-Hill, 2006.