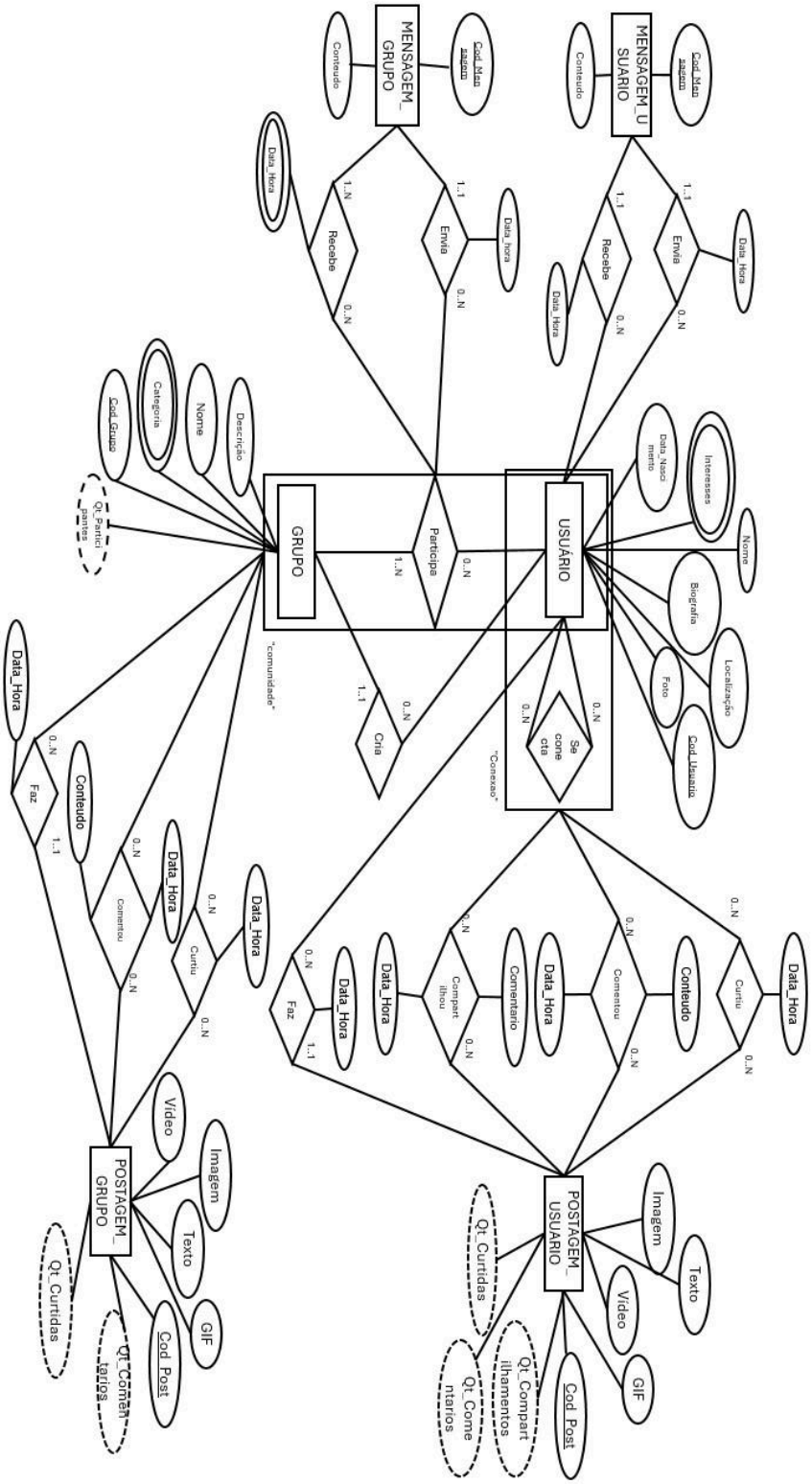
Nomes:

1. Nícolas Araújo Sampaio
2. Pedro Henrique Barros Mendonça
3. Thiago Vinicius Costa Guimarães

1. Modelo EER



2. Esquema Relacional

usuario(cod_usuario, nome, data_nascimento, biografia, localizacao, foto)

mensagem_usuario(cod_msg, #cod_usu_remetente, #cod_usuario_destinatario, conteudo, envio_data_hora, recebe_data_hora)
mensagem_usuario[cod_usuario_remetente] -> usuario[cod_usuario]
mensagem_usuario[cod_usuario_destinatario] -> usuario[cod_usuario]

interesse(#cod_interesse, nome)

interesse_usuario(#cod_interesse, #cod_usuario)
interesse_usuario[cod_usuario] -> usuario[cod_usuario]
interesse_usuario[cod_interesse] -> interesse[cod_interesse]

grupo(cod_grupo, #cod_criador, nome, descricao)
grupo[cod_criador] -> usuario[cod_usuario]

mensagem_grupo(cod_msg, #cod_grupo, data_hora_envio, conteudo)
mensagem_grupo[cod_grupo] -> grupo[cod_grupo]
mensagem_grupo[cod_usuario] -> usuario[cod_usuario]

dthr_msg_destinatario(#cod_msg, #cod_gru, #cod_usuario_destin, data_hora_recebe)
dthr_msg_destinatario[cod_gru] -> grupo[cod_grupo]
dthr_msg_destinatario[cod_usuario_destin] -> usuario[cod_usuario]
dthr_msg_destinatario[cod_msg] -> mensagem_grupo[cod_msg]

categoria_grupo(#cod_grupo, categoria)
categoria_grupo[cod_grupo] -> grupo[cod_grupo]
postagem_grupo(cod_post, #cod_grupo, #cod_usuario, imagem, texto, video, gif, data_hora)
postagem_grupo[cod_grupo] -> grupo[cod_grupo]
postagem_grupo[cod_usuario] -> usuario[cod_usuario]

comentario_grupo(#cod_grupo, #cod_usuario, #cod_post, data_hora, conteudo)
comentario_grupo[cod_grupo] -> grupo[cod_grupo]
comentario_grupo[cod_usuario] -> usuario[cod_usuario]
comentario_grupo[cod_post] -> postagem_grupo[cod_post]

curtida_grupo(#cod_grupo, #cod_usuario, #cod_post , data_hora)
curtida_grupo[cod_grupo] -> grupo[cod_grupo]
curtida_grupo[cod_usuario] -> usuario[cod_usuario]
curtida_grupo[cod_post] -> postagem_grupo[cod_post]

postagem_usuario(cod_post, #cod_usuario, imagem, texto, video, gif, data_hora)
postagem_usuario[cod_usuario] -> usuario[cod_usuario]

compartilhamento(#cod_usuario, #cod_post, comentario, data_hora)

compartilhamento[cod_usuario] -> usuario[cod_usuario]
compartilhamento[cod_post] -> postagem_usuario[cod_post]

comentario(#cod_usuario, #cod_post, data_hora, conteudo, data_hora)
comentario[cod_usuario] -> usuario[cod_usuario]
comentario[cod_post] -> postagem_usuario[cod_post]

curtida(#cod_usuario, #cod_post, data_hora)
curtida[cod_usuario] -> usuario[cod_usuario]
curtida[cod_post] -> postagem_usuario[cod_post]

conexao(#cod_usuario1, #cod_usuario2)
conexao[cod_usuario1] -> usuario[cod_usuario]
conexao[cod_usuario2] -> usuario[cod_usuario]

participante_comunidade(#cod_usuario, #cod_grupo)
participante_comunidade[cod_usuario] -> usuario[cod_usuario]
participante_comunidade[cod_post] -> grupo[cod_grupo]

3. Script de Criação do Banco

```
DROP DATABASE if EXISTS `ibd-pratical-work`;
CREATE DATABASE `ibd-pratical-work`;
USE `ibd-pratical-work`;

CREATE TABLE `usuario` (
    `cod_usuario` INT NOT NULL AUTO_INCREMENT,
    `nome` VARCHAR(255) NOT NULL,
    `data_nascimento` DATE NOT NULL,
    `biografia` VARCHAR(255),
    `localizacao` VARCHAR(255),
    `foto_perfil` BLOB,
    PRIMARY KEY (cod_usuario)
)  ENGINE=INNODB;

CREATE TABLE `mensagem_usuario` (
    `cod_mensagem` INT NOT NULL AUTO_INCREMENT,
    `cod_usu_remetente` INT DEFAULT NULL,
```

```
    `cod_usu_destinatario` INT DEFAULT NULL,
    `conteudo` VARCHAR(255) NOT NULL,
    `envio_data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    `recebe_data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (cod_mensagem),
    FOREIGN KEY (cod_usu_remetente)
        REFERENCES usuario (cod_usuario)
        ON DELETE SET NULL,
    FOREIGN KEY (cod_usu_destinatario)
        REFERENCES usuario (cod_usuario)
        ON DELETE SET NULL
)  ENGINE=INNODB;

CREATE TABLE `interesse` (
    `cod_interesse` INT NOT NULL AUTO_INCREMENT,
    `nome` VARCHAR(255),
    PRIMARY KEY (cod_interesse)
)  ENGINE=INNODB;

CREATE TABLE `interesse_usuario` (
    `cod_interesse` INT NOT NULL,
    `cod_usuario` INT NOT NULL,
    PRIMARY KEY (cod_interesse , cod_usuario),
    FOREIGN KEY (cod_interesse)
        REFERENCES interesse (cod_interesse)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `grupo` (
    `cod_grupo` INT NOT NULL AUTO_INCREMENT,
    `cod_criador` INT NOT NULL,
    `nome` VARCHAR(255) NOT NULL,
    `descricao` VARCHAR(255),
    PRIMARY KEY (cod_grupo),
    FOREIGN KEY (cod_criador)
        REFERENCES usuario (cod_usuario)
)  ENGINE=INNODB;

CREATE TABLE `mensagem_grupo` (
    `cod_msg` INT NOT NULL AUTO_INCREMENT,
    `cod_grupo` INT NOT NULL,
```

```
    `cod_usuario` INT DEFAULT NULL,
    `conteudo` VARCHAR(255) NOT NULL,
    `data_hora_envio` DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (cod_msg),
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE SET NULL
)  ENGINE=INNODB;

CREATE TABLE `dthr_msg_destinatario` (
    `cod_msg` INT NOT NULL,
    `cod_grupo` INT NOT NULL,
    `cod_usuario` INT NOT NULL,
    `data_hora_recebe` DATETIME DEFAULT NULL,
    PRIMARY KEY (cod_msg , cod_usuario , cod_grupo),
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo),
    FOREIGN KEY (cod_msg)
        REFERENCES mensagem_grupo (cod_msg)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `categoria_grupo` (
    `cod_categoria` INT NOT NULL,
    `cod_grupo` INT NOT NULL,
    PRIMARY KEY (cod_categoria , cod_grupo),
    FOREIGN KEY (cod_categoria)
        REFERENCES interesse (cod_interesse)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `postagem_grupo` (
    `cod_post` INT NOT NULL AUTO_INCREMENT,
    `cod_grupo` INT NOT NULL,
    `cod_usuario` INT NOT NULL,
```

```
    `texto` VARCHAR(1000) NOT NULL,
    `imagem` MEDIUMBLOB,
    `video` LONGBLOB,
    `gif` BLOB,
    `data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (cod_post , cod_grupo , cod_usuario),
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `comentario_grupo` (
    `cod_grupo` INT NOT NULL,
    `cod_usuario` INT NOT NULL,
    `cod_post_grupo` INT NOT NULL,
    `data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    `conteudo` VARCHAR(255) NOT NULL,
    PRIMARY KEY (cod_grupo , cod_usuario , cod_post_grupo ,
data_hora),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_post_grupo)
        REFERENCES postagem_grupo (cod_post)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `curtida_grupo` (
    `cod_grupo` INT NOT NULL,
    `cod_usuario` INT NOT NULL,
    `cod_post_grupo` INT NOT NULL,
    `data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (cod_grupo , cod_usuario , cod_post_grupo),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
```

```
        ON DELETE CASCADE,
    FOREIGN KEY (cod_post_grupo)
        REFERENCES postagem_grupo (cod_post)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `postagem_usuario` (
    `cod_post` INT NOT NULL AUTO_INCREMENT,
    `cod_usuario` INT NOT NULL,
    `texto` VARCHAR(256) NOT NULL,
    `imagem` MEDIUMBLOB,
    `video` LONGBLOB,
    `gif` BLOB,
    `data_hora` DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (cod_post , cod_usuario),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `compartilhamento` (
    `cod_usuario` INT NOT NULL,
    `cod_post` INT NOT NULL,
    `comentario` VARCHAR(255),
    `data_hora` DATETIME NOT NULL,
    PRIMARY KEY (cod_usuario , cod_post),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_post)
        REFERENCES postagem_usuario (cod_post)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `comentario` (
    `cod_usuario` INT NOT NULL,
    `cod_post` INT NOT NULL,
    `conteudo` VARCHAR(255) NOT NULL,
    `data_hora` DATETIME NOT NULL,
    PRIMARY KEY (cod_usuario , cod_post),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_post)
```

```
        REFERENCES postagem_usuario (cod_post)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `curtida` (
    `cod_usuario` INT NOT NULL,
    `cod_post` INT NOT NULL,
    `data_hora` DATETIME NOT NULL,
    PRIMARY KEY (cod_usuario , cod_post),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_post)
        REFERENCES postagem_usuario (cod_post)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `conexao` (
    `cod_usuario1` INT NOT NULL,
    `cod_usuario2` INT NOT NULL,
    PRIMARY KEY (cod_usuario1 , cod_usuario2),
    FOREIGN KEY (cod_usuario1)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_usuario2)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE
)  ENGINE=INNODB;

CREATE TABLE `participante_comunidade` (
    `cod_usuario` INT NOT NULL,
    `cod_grupo` INT NOT NULL,
    PRIMARY KEY (cod_usuario , cod_grupo),
    FOREIGN KEY (cod_usuario)
        REFERENCES usuario (cod_usuario)
        ON DELETE CASCADE,
    FOREIGN KEY (cod_grupo)
        REFERENCES grupo (cod_grupo)
        ON DELETE CASCADE
)  ENGINE=INNODB;
```

4. **Consultas SQL**

1. Consulta de perfil do usuário: Recuperar informações do perfil de um usuário específico, incluindo nome, foto, biografia, etc

```
SELECT
   *
FROM
   usuario
WHERE
   cod_usuario = ''; -- Substituir aspas pelo código do usuário
```

2. Consulta de conexões de um usuário: Obter a lista de amigos de um determinado usuário.

```
SELECT
   *
FROM
   usuario u
      INNER JOIN
   conexao c ON u.cod_usuario IN (c.cod_usuario1 , c.cod_usuario2)
      AND '' IN (c.cod_usuario1 , c.cod_usuario2) -- Substituir aspas pelo código
do usuário
      AND u.cod_usuario <> '' -- Substituir aspas pelo código do usuário
;
```

3. Consulta de postagens de um usuário: Recuperar todas as postagens feitas por um usuário específico, ordenadas por data de publicação (as mais recentes primeiro).

```
SELECT
   *
FROM
   postagem_usuario
WHERE
   cod_usuario = '' -- Substituir aspas pelo código do usuário
ORDER BY data_hora DESC;
```

4. Consulta de postagens em um grupo: Listar 20 as postagens mais recentes feitas em um grupo específico.

```
SELECT
   *
FROM
   postagem_grupo
```

```
WHERE
   cod_grupo = '' -- Substituir aspas pelo código do grupo
ORDER BY data_hora DESC
LIMIT 20;
```

5. Consulta de mensagens privadas: Listar as 10 mensagens privadas mais recentes trocadas entre 2 usuários dados como entrada.

```
SELECT
   *
FROM
   mensagem_usuario
WHERE
   cod_usu_remetente = '' -- Substituir aspas pelo código do usuário
      AND cod_usu_destinatario = '' -- Substituir aspas pelo código do usuário
UNION
SELECT
   *
FROM
   mensagem_usuario
WHERE
   cod_usu_remetente = '' -- Substituir aspas pelo código do usuário
      AND cod_usu_destinatario = '' -- Substituir aspas pelo código do usuário
ORDER BY envio_data_hora DESC , recebe_data_hora DESC
LIMIT 10;
```

6. Consulta de busca de usuários: Dado uma string como entrada, buscar os usuários cujos nomes contenham a string fornecida.

```
SELECT
   *
FROM
   usuario
WHERE
   nome LIKE "%"%"; -- Substituir aspas pelo nome do usuário
```

7. Consulta de tendências: Listar o identificador dos 5 posts com mais interações nos últimos 7 dias.

```
WITH post_usu AS (
   SELECT
      pu.cod_post AS cod_post,
      COUNT(ct.cod_usuario) AS qt_curtidas,
      COUNT(cm.cod_usuario) AS qt_comentarios,
```

```
        (COUNT(ct.cod_usuario) + COUNT(cm.cod_usuario)) AS indice_interacao
    FROM postagem_usuario pu
    LEFT JOIN curtida ct
        ON pu.cod_post = ct.cod_post
    LEFT JOIN comentario cm
        ON pu.cod_post = cm.cod_post
    GROUP BY pu.cod_post
),
post_group AS (
    SELECT
        pg.cod_post AS cod_post,
        COUNT(ct.cod_grupo) AS qt_curtidas,
        COUNT(cm.cod_grupo) AS qt_comentarios,
        (COUNT(ct.cod_grupo) + COUNT(cm.cod_grupo)) AS indice_interacao
    FROM postagem_grupo pg
    LEFT JOIN curtida_grupo ct
        ON pg.cod_post = ct.cod_post_grupo
    LEFT JOIN comentario_grupo cm
        ON pg.cod_post = cm.cod_post_grupo
    GROUP BY pg.cod_post
)
SELECT
    cod_post,
    qt_curtidas,
    qt_comentarios,
    indice_interacao
FROM post_usu
UNION ALL
SELECT
    cod_post,
    qt_curtidas,
    qt_comentarios,
    indice_interacao
FROM post_group
ORDER BY indice_interacao DESC
LIMIT 5;
```

8. Consulta de análise de engajamento: Dado um post de um usuário, fazer a contagem de quantos usuários interagiram com ele nos últimos 7 dias.

```
-- Consulta para usuários

SELECT
    SUM(quantidade) AS interacoes
FROM
    (SELECT
        COUNT(cod_usuario) AS quantidade
```

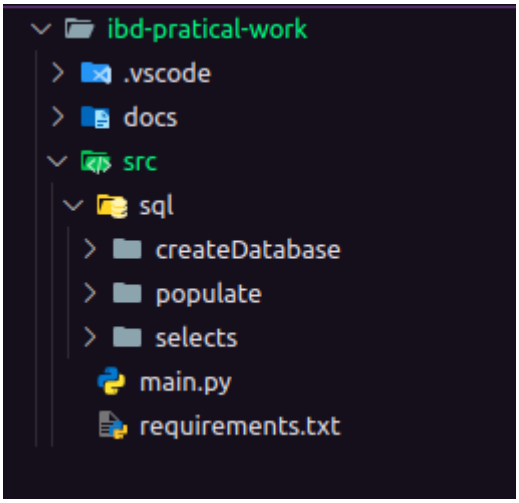
```
    FROM
        curtida
    WHERE
        cod_post = ''-- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY UNION SELECT
        COUNT(cod_usuario) AS quantidade
    FROM
        comentario
    WHERE
        cod_post = '' -- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY UNION SELECT
        COUNT(cod_usuario) AS quantidade
    FROM
        compartilhamento
    WHERE
        cod_post = '' -- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY) AS interacoes;

-- Consulta para grupos

SELECT
    SUM(quantidade) AS interacoes
FROM
    (SELECT
        COUNT(cod_usuario) AS quantidade
    FROM
        curtida_grupo
    WHERE
        cod_post_grupo = '' -- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY UNION SELECT
        COUNT(cod_usuario) AS quantidade
    FROM
        comentario_grupo
    WHERE
        cod_post_grupo = '' -- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY) AS interacoes;
```

Tutorial de como usar o programa:

vídeo: Apresentacao_IBD.mp4

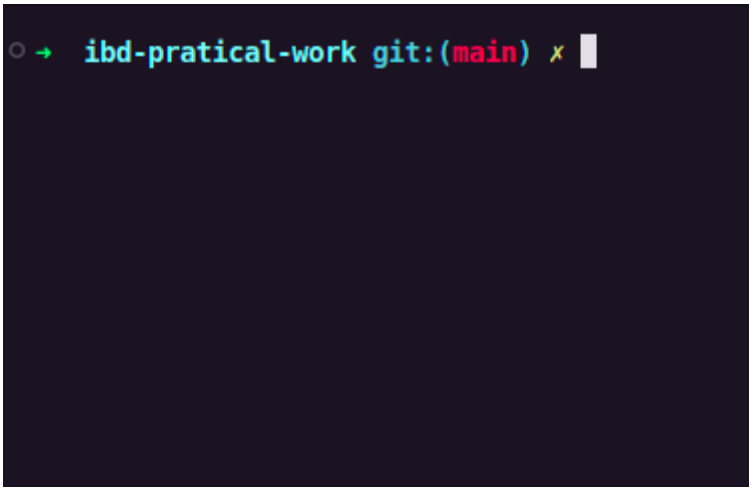
1. Estrutura das pastas:

- docs: consta o tutorial do programa e o modelo ER;
- src:
    - sql:
        - createDatabase: script sql que cria o banco de dados
        - populate: scripts sql que populam o banco de dados
        - selects: consultas sql solicitadas
    - Arquivo main.py: programa que cria e popula o banco de dados
    - Arquivo requirements.txt: arquivo que contém as bibliotecas python necessárias para rodar o programa



2. Passo a passo

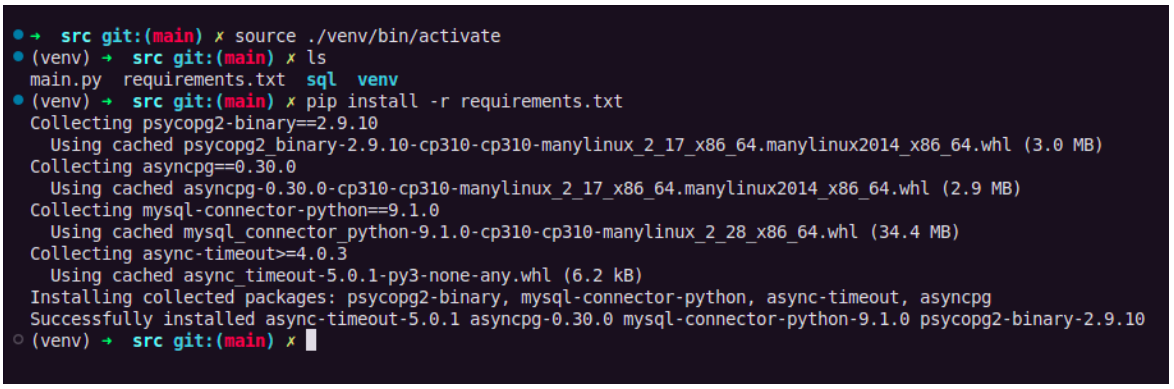
    1. Em primeiro momento, tendo o python e o SGBD instalados na máquina, abra o terminal:



2. Entre no nível dentro da pasta src:

3. Digite o comando “source ./venv/bin/activate”:

4. Digite o comando “pip install -r requirements.txt”:



5. Acesse o arquivo main.py e coloque as credenciais do banco na linha 118:

```
# Usando a função
host = ''
user = ''
password = ''
sql_file_create = './sql/createDatabase/create_database.sql'
database = "ibd-pratical-work"
```

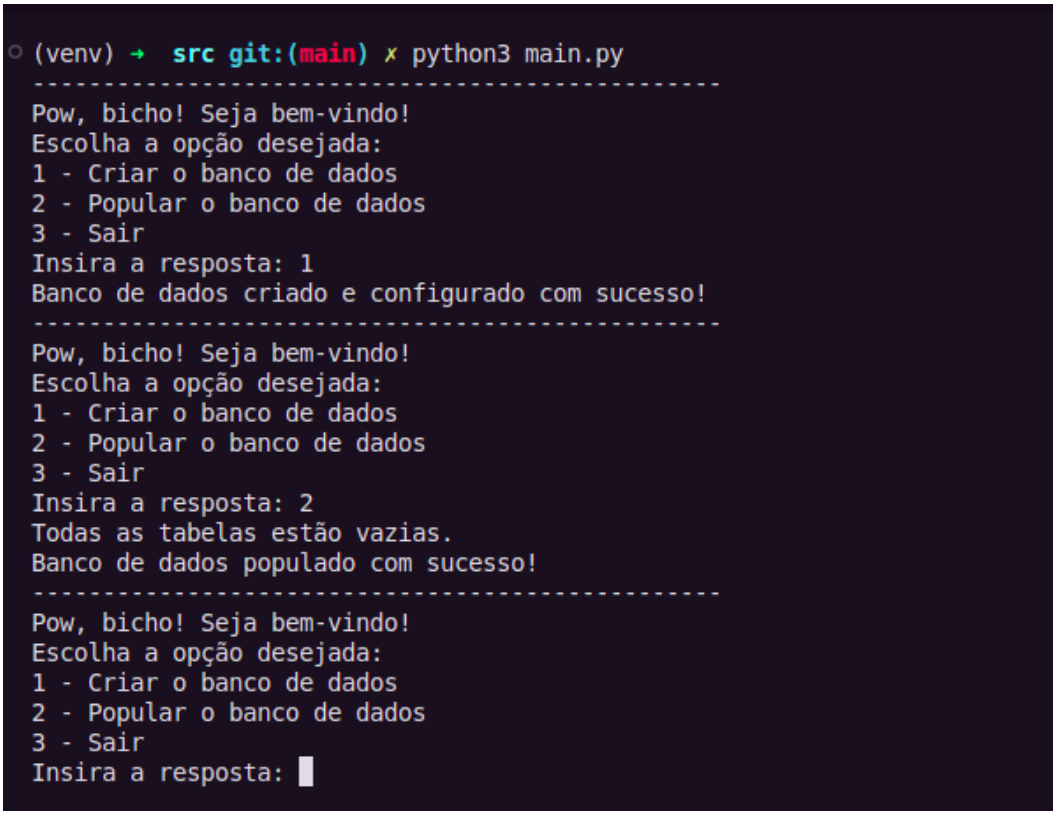
6. Dê o comando “python main.py”:

```
(venv) ➜  src git:(main) ✗ python3 main.py
-------------------------------------------------
Pow, bicho! Seja bem-vindo!
Escolha a opção desejada:
1 - Criar o banco de dados
2 - Popular o banco de dados
3 - Sair
Insira a resposta:
```

7. Escolha a opção desejada. Caso escolha a 1, o programa irá criar o banco de dados quando ele não existir.

```
(venv) ➜  src git:(main) ✗ python3 main.py
-------------------------------------------------
Pow, bicho! Seja bem-vindo!
Escolha a opção desejada:
1 - Criar o banco de dados
2 - Popular o banco de dados
3 - Sair
Insira a resposta: 1
Banco de dados criado e configurado com sucesso!
-------------------------------------------------
Pow, bicho! Seja bem-vindo!
Escolha a opção desejada:
1 - Criar o banco de dados
2 - Popular o banco de dados
3 - Sair
Insira a resposta:
```

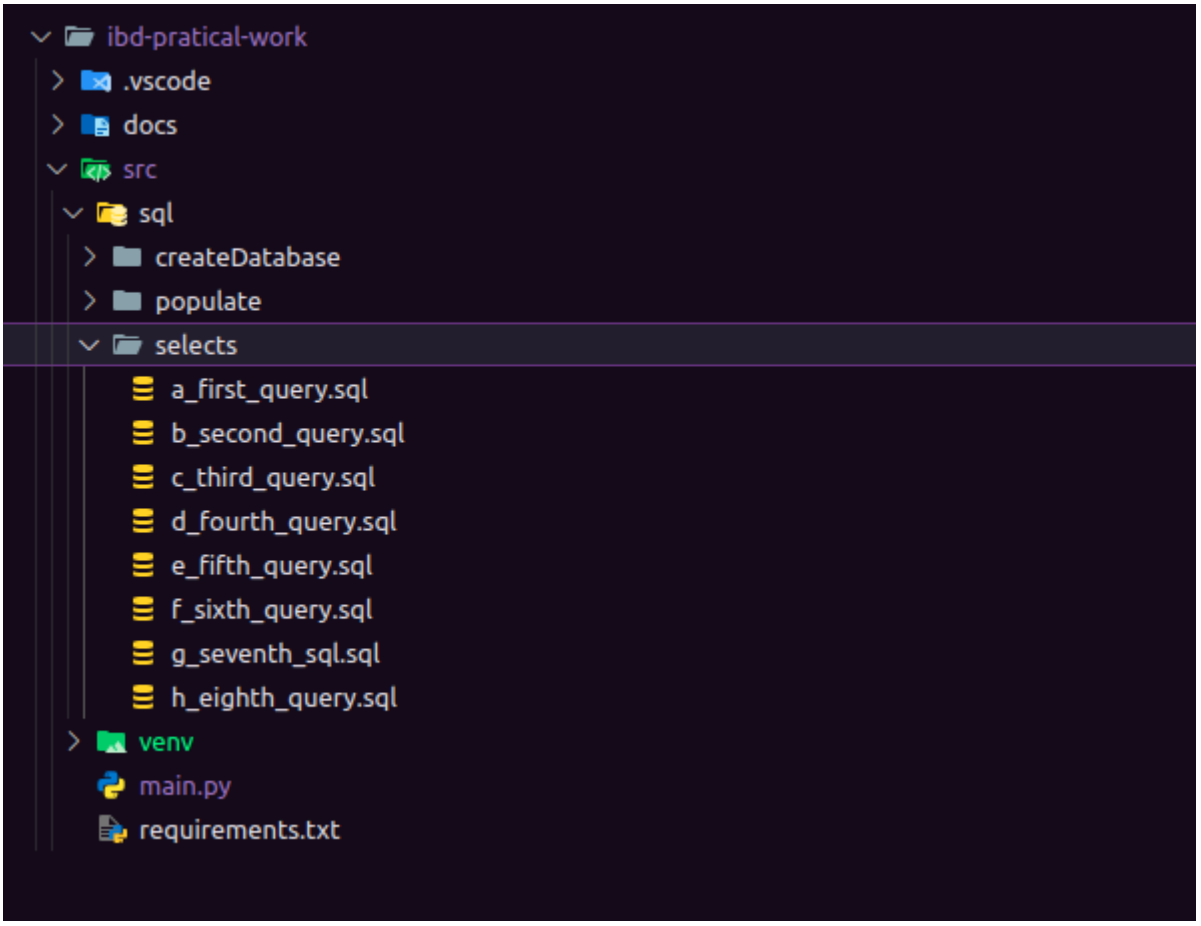
8. Em seguida, caso queira popular o banco, pode selecionar a opção 2:



9. Por último, pode apertar a opção 3 para sair.

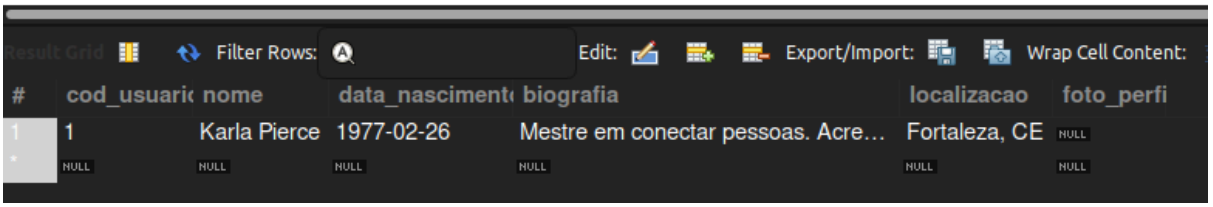
Tutorial de como executar as consultas:

Olhando dentro da pasta selects, as consultas podem ser coletadas e executadas dentro do sgbd:



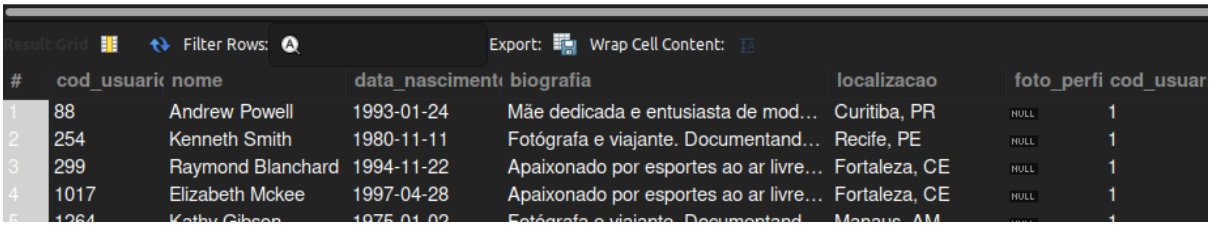
Exemplo de execução da primeira consulta:

```
3 •  SELECT
4        *
5    FROM
6        usuario
7    WHERE
8        cod_usuario = 1; -- Substituir aspas pelo código do usuário
```



Exemplo de execução da segunda consulta:

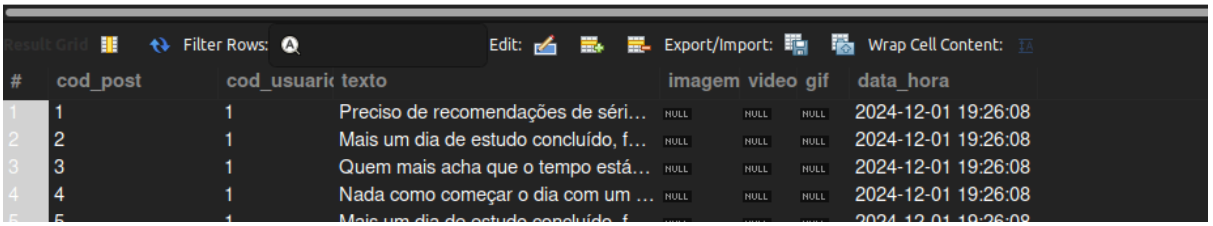
```sql
SELECT
    *
FROM
    usuario u
        INNER JOIN
    conexao c ON u.cod_usuario IN (c.cod_usuario1 , c.cod_usuario2)
        AND 1 IN (c.cod_usuario1 , c.cod_usuario2) -- Substituir aspas pelo código do usuário
        AND u.cod_usuario <> 1 -- Substituir aspas pelo código do usuário
```



#	cod_usuari	nome	data_nascimento	biografia	localizacao	foto_perfi	cod_usuar
1	88	Andrew Powell	1993-01-24	Mãe dedicada e entusiasta de mod…	Curitiba, PR	NULL	1
2	254	Kenneth Smith	1980-11-11	Fotógrafa e viajante. Documentand…	Recife, PE	NULL	1
3	299	Raymond Blanchard	1994-11-22	Apaixonado por esportes ao ar livre…	Fortaleza, CE	NULL	1
4	1017	Elizabeth Mckee	1997-04-28	Apaixonado por esportes ao ar livre…	Fortaleza, CE	NULL	1

Exemplo de execução da terceira consulta:

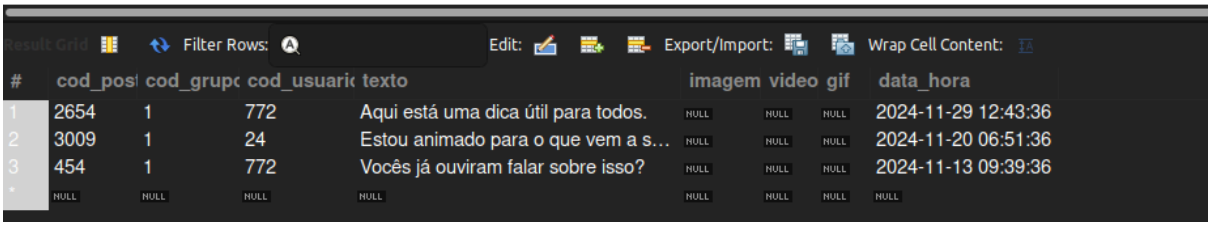
```sql
SELECT
    *
FROM
    postagem_usuario
WHERE
    cod_usuario = 1 -- Substituir aspas pelo código do usuário
ORDER BY data_hora DESC;
```



#	cod_post	cod_usuari	texto	imagem	video	gif	data_hora
1	1	1	Preciso de recomendações de séri…	NULL	NULL	NULL	2024-12-01 19:26:08
2	2	1	Mais um dia de estudo concluído, f…	NULL	NULL	NULL	2024-12-01 19:26:08
3	3	1	Quem mais acha que o tempo está…	NULL	NULL	NULL	2024-12-01 19:26:08
4	4	1	Nada como começar o dia com um …	NULL	NULL	NULL	2024-12-01 19:26:08

Exemplo de execução da quarta consulta:

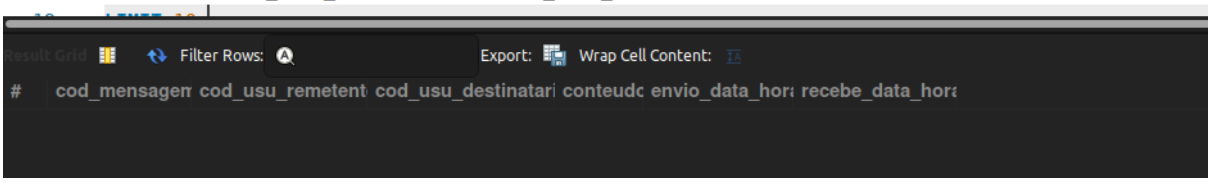
```sql
SELECT
    *
FROM
    postagem_grupo
WHERE
    cod_grupo = 1 -- Substituir aspas pelo código do grupo
ORDER BY data_hora DESC
LIMIT 20;
```



#	cod_post	cod_grupo	cod_usuari	texto	imagem	video	gif	data_hora
1	2654	1	772	Aqui está uma dica útil para todos.	NULL	NULL	NULL	2024-11-29 12:43:36
2	3009	1	24	Estou animado para o que vem a s…	NULL	NULL	NULL	2024-11-20 06:51:36
3	454	1	772	Vocês já ouviram falar sobre isso?	NULL	NULL	NULL	2024-11-13 09:39:36
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

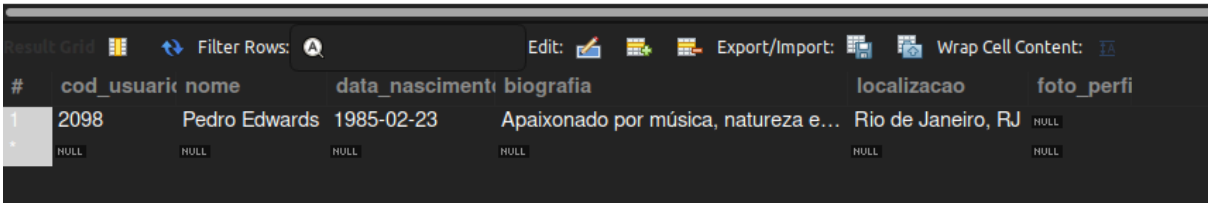
Exemplo de execução da quinta consulta:

```sql
SELECT
    *
FROM
    mensagem_usuario
WHERE
    cod_usu_remetente = 1 -- Substituir aspas pelo código do usuário
        AND cod_usu_destinatario = 2 -- Substituir aspas pelo código do usuário
UNION
SELECT
    *
FROM
    mensagem_usuario
WHERE
    cod_usu_remetente = 2 -- Substituir aspas pelo código do usuário
        AND cod_usu_destinatario = 1 -- Substituir aspas pelo código do usuário
ORDER BY envio_data_hora DESC , recebe_data_hora DESC
```



Exemplo de execução da sexta consulta:

```sql
SELECT
    *
FROM
    usuario
WHERE
    nome LIKE "%pedro%"; -- Substituir aspas pelo nome do usuário
```



Exemplo de execução da sétima consulta:

```sql
WITH post_usu AS (
    SELECT
        pu.cod_post AS cod_post,
        COUNT(ct.cod_usuario) AS qt_curtidas,
        COUNT(cm.cod_usuario) AS qt_comentarios,
        (COUNT(ct.cod_usuario) + COUNT(cm.cod_usuario)) AS indice_interacao
    FROM postagem_usuario pu
    LEFT JOIN curtida ct
        ON pu.cod_post = ct.cod_post
    LEFT JOIN comentario cm
        ON pu.cod_post = cm.cod_post
    GROUP BY pu.cod_post
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	cod_post	qt_curtidas	qt_comentarios	indice_interacao
1	138	60	60	120
2	557	60	60	120
3	2486	56	56	112
4	3966	54	54	108
5	4999	54	54	108

Exemplo de execução da oitava consulta:

```sql
-- Consulta para usuários

SELECT
    SUM(quantidade) AS interacoes
FROM
    (SELECT
        COUNT(cod_usuario) AS quantidade
    FROM
        curtida
    WHERE
        cod_post = 1 -- Substituir as aspas pelo Código do Post desejado
            AND data_hora >= NOW() - INTERVAL 7 DAY UNION SELECT
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	interacoes
1	4