



**Faculdade de Design,
Tecnologia e Comunicação**
Universidade Europeia

Think Toilet

Base de Dados

Curso: Engenharia Informática

Semestre: 2024/2025 - Terceiro Semestre

Nycolas Souza - 20230989

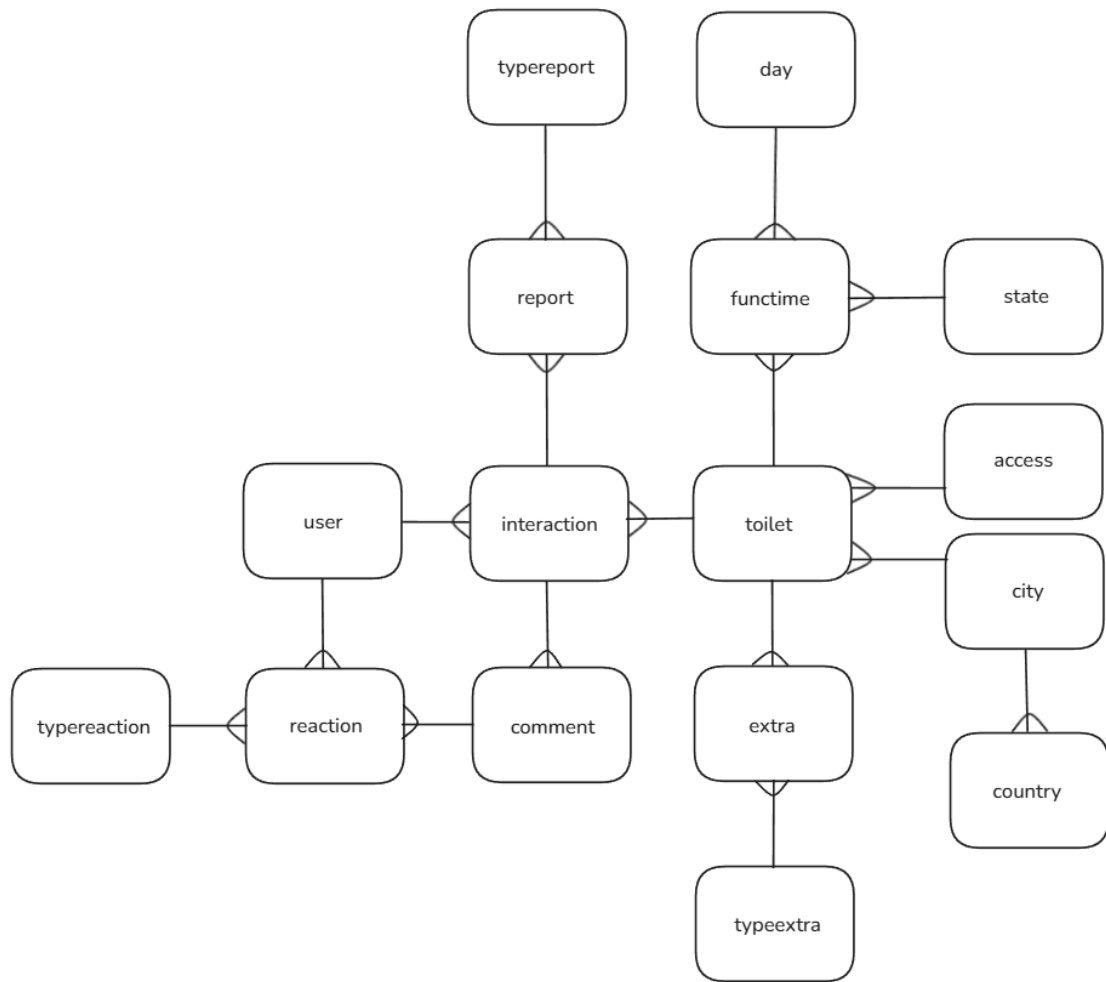
Luan Ribeiro - 20230692

Lohanne Guedes - 20220085

Introdução

O projeto **Think Toilet** é uma aplicação móvel que ajuda os utilizadores a encontrar e avaliar casas de banho próximas. A aplicação exibe um mapa interativo com as casas de banho mais bem avaliadas e fornece informações como preço, acessibilidade e restrições de uso (gratuito, público ou para clientes). Os utilizadores podem avaliar critérios como limpeza, acessibilidade, papel disponível e estrutura, além de deixar comentários. A aplicação também permite sugerir novas casas de banho, denunciar locais ou comentários inadequados e visualizar seu histórico de avaliações. Com integração ao Google Maps, oferece rotas para facilitar o acesso aos locais.

Modelo Entidade-Relacionamento



user

A tabela **user** armazena as informações dos usuários registrados na plataforma. Ela é fundamental, pois serve como base para todas as interações dentro do sistema. Cada usuário é identificado de forma única, o que permite rastrear suas ações e feedbacks.

Colunas	Tipo de Dado	Observação
user_id	int	ID do usuário
user_name	varchar(50)	Nome do usuário
user_email	varchar(100)	Email do usuário
user_pwd	varchar(255)	Senha do usuário
user_points	int?	Pontos do usuário
user_iconid	varchar(255)	ID do ícone do usuário
user_bdate	date?	Data de nascimento do usuário
user_cdate	date	Data de criação da conta do usuário

A coluna `user_pwd` utiliza um `varchar(255)` pois idealmente a ideia é trabalhar com senhas em hash, e essa quantidade de caracteres é suficiente para utilizar qualquer tipo de hash. Já a `user_iconid` serve para guardar um endereço URL, equivalente ao arquivo onde consta o ícone.

Scripts - Create:

```
create table user (  
    user_id INT NOT NULL auto_increment,  
    user_name VARCHAR(50) NOT NULL,  
    user_email VARCHAR(100) NOT NULL,  
    user_pwd VARCHAR(255) NOT NULL,  
    user_points INT,  
    user_iconid VARCHAR(255),  
    user_bdate DATE,  
    user_cdate DATE NOT NULL,  
    primary key (user_id)  
);
```

Scripts - Populate:

```
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Alice Silva', 'alice@gmail.com', 'password123', 120,
'icon1', STR_TO_DATE('1990-05-14', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Bruno Costa', 'bruno@hotmail.com', 'securepass456',
300, 'icon2', STR_TO_DATE('1985-09-25', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Carla Nunes', 'carla@gmail.com', 'mypassword789', 200,
'icon3', STR_TO_DATE('1992-03-18', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Diego Martins', 'diego@yahoo.com', 'diego_pass', 150,
'icon4', STR_TO_DATE('1988-12-22', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Eva Rocha', 'eva.rocha@gmail.com', 'password_12345',
220, 'icon5', STR_TO_DATE('1995-08-12', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Fábio Souza', 'fabio.souza@gmail.com', 'fabio_pass',
180, 'icon6', STR_TO_DATE('1991-11-30', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Gisele Almeida', 'gisele.almeida@gmail.com',
'gisele123', 250, 'icon7', STR_TO_DATE('1989-07-05', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Henrique Lima', 'henrique.lima@hotmail.com',
'henrique456', 100, 'icon8', STR_TO_DATE('1993-01-15', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Isabel Ferreira', 'isabel.ferreira@gmail.com',
'isabel789', 350, 'icon9', STR_TO_DATE('1984-04-28', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('João Pedro', 'joao.pedro@gmail.com', 'joao_pass', 275,
'icon10', STR_TO_DATE('1990-10-02', '%Y-%m-%d'), CURDATE());
insert into user (user_name, user_email, user_pwd, user_points, user_iconid,
user_bdate, user_cdate) values('Larissa Gomes', 'larissa.gomes@hotmail.com',
'larissa123', 320, 'icon11', STR_TO_DATE('1994-06-17', '%Y-%m-%d'), CURDATE());
```

toilet

A tabela **toilet** contém informações sobre as casas de banho disponíveis para os usuários. Essa tabela é crucial, pois fornece a localização e características das casas de banho, facilitando a busca e a interação dos usuários com o sistema.

Colunas	Tipo de Dado	Observação
toil_id	int	ID da casa de banho
toil_city_id	int	ID estrangeiro para a cidade da casa de banho
toil_acs_id	int	ID estrangeiro do tipo de acesso a casa de banho
toil_name	varchar(50)	Nome da casa de banho ou estabelecimento
toil_lat	double	Latitude da casa de banho
toil_long	double	Longitude da casa de banho
toil_address	varchar(255)	Endereço da casa de banho
toil_mapsid	varchar(255)?	PlaceID da casa de banho
toil_cdate	date	Data da criação da casa de banho
toil_image	varchar(255)?	URL da imagem da casa de banho ou estabelecimento

Scripts - Create:

```
create table toilet (  
    toil_id INT NOT NULL auto_increment,  
    toil_city_id INT NOT NULL,  
    toil_acs_id INT NOT NULL,  
    toil_name VARCHAR(50) NOT NULL,  
    toil_lat DOUBLE NOT NULL,  
    toil_long DOUBLE NOT NULL,  
    toil_address VARCHAR(255) NOT NULL,
```

```
        toil_mapsid VARCHAR(255),  
        toil_cdate DATE NOT NULL,  
        toil_image VARCHAR(255),  
        primary key (toil_id)  
);
```

Scripts - Populate:

```
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (1, 1, 'Main Mall Restroom',  
40.7128, -74.0060, '123 Main St', 'maplink1', CURDATE(), 'image1.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (1, 2, 'Library Restroom',  
40.7129, -74.0059, '456 Library Ave', 'maplink2', CURDATE(), 'image2.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (2, 1, 'Park Restroom',  
40.7130, -74.0058, '789 Park St', 'maplink3', CURDATE(), 'image3.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (2, 2, 'Office Restroom',  
40.7131, -74.0057, '101 Office Ave', 'maplink4', CURDATE(), 'image4.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (3, 1, 'Stadium Restroom',  
40.7132, -74.0056, '102 Stadium Blvd', 'maplink5', CURDATE(), 'image5.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (3, 2, 'Shopping Restroom',  
40.7133, -74.0055, '103 Shopping Plaza', 'maplink6', CURDATE(), 'image6.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (1, 1, 'Central Square  
Restroom', 40.7140, -74.0040, '150 Central Sq', 'maplink7', CURDATE(), 'image7.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (2, 2, 'City Park Restroom',  
40.7150, -74.0030, '200 City Park', 'maplink8', CURDATE(), 'image8.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (3, 1, 'Riverfront  
Restroom', 40.7160, -74.0020, '300 Riverfront Blvd', 'maplink9', CURDATE(),  
'image9.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (1, 2, 'Museum Restroom',  
40.7170, -74.0010, '400 Museum Ave', 'maplink10', CURDATE(), 'image10.jpg');  
insert into toilet (toil_city_id, toil_acs_id, toil_name, toil_lat, toil_long,  
toil_address, toil_mapsid, toil_cdate, toil_image) values (2, 1, 'Zoo Restroom',  
40.7180, -74.0000, '500 Zoo Ln', 'maplink11', CURDATE(), 'image11.jpg');
```

interaction

A tabela **interaction** é responsável por registrar as interações entre usuários e casas de banho. Essa tabela é vital para entender o uso e a experiência dos usuários em relação às casas de banho. Além disso, ela possibilita a ligação entre os usuários e suas ações, como comentários e denúncias.

Colunas	Tipo de Dado	Observação
int_id	int	ID da interação
int_user_id	int	ID estrangeiro para o usuário
int_toil_id	int	ID estrangeiro para a casa de banho

Scripts - Create:

```
create table interaction (  
    int_id INT NOT NULL auto_increment,  
    int_user_id INT NOT NULL,  
    int_toil_id INT NOT NULL,  
    primary key (int_id)  
);
```

Scripts - Populate:

```
insert into interaction (int_user_id, int_toil_id) values (1, 1);  
insert into interaction (int_user_id, int_toil_id) values (1, 2);  
insert into interaction (int_user_id, int_toil_id) values (2, 1);  
insert into interaction (int_user_id, int_toil_id) values (2, 3);  
insert into interaction (int_user_id, int_toil_id) values (3, 2);  
insert into interaction (int_user_id, int_toil_id) values (3, 3);  
insert into interaction (int_user_id, int_toil_id) values (4, 4);  
insert into interaction (int_user_id, int_toil_id) values (5, 5);  
insert into interaction (int_user_id, int_toil_id) values (6, 6);  
insert into interaction (int_user_id, int_toil_id) values (1, 4);  
insert into interaction (int_user_id, int_toil_id) values (2, 5);  
insert into interaction (int_user_id, int_toil_id) values (3, 6);
```


report

A tabela **report** registra as denúncias feitas pelos usuários sobre as casas de banho, permitindo que problemas e melhorias sejam identificados. Sua importância reside na capacidade de coletar feedback estruturado, contribuindo para a manutenção e aprimoramento das casas de banho.

Colunas	Tipo de Dado	Observação
rep_id	int	ID da denúncia
rep_trp_id	int	ID estrangeiro para o tipo da denúncia
rep_int_id	int	ID estrangeiro para a interação
rep_cdate	date	Data de criação da denúncia

Scripts - Create:

```
create table report (  
    rep_id INT NOT NULL auto_increment,  
    rep_trp_id INT NOT NULL,  
    rep_int_id INT NOT NULL,  
    rep_cdate DATE NOT NULL,  
    primary key (rep_id)  
);
```

Scripts - Populate:

```
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (1, 1, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (2, 1, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (1, 2, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (3, 3, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (2, 4, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (1, 5, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (2, 6, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (3, 1, CURDATE());  
insert into report (rep_trp_id, rep_int_id, rep_cdate) values (1, 4, CURDATE());
```

reaction

A tabela **reaction** registra as reações dos usuários aos comentários. Sua importância reside na capacidade de coletar e categorizar as opiniões dos usuários sobre o que outros usuários estão dizendo.

Colunas	Tipo de Dado	Observação
react_id	int	ID da reação
react_user_id	int	ID estrangeiro para o usuário
react_cmm_id	int	ID estrangeiro para o comentário
react_trc_id	int	ID estrangeiro para o tipo de reação
react_cdate	date	Data de criação da reação

Scripts - Create:

```
create table reaction (  
    react_id INT NOT NULL auto_increment,  
    react_user_id INT NOT NULL,  
    react_cmm_id INT NOT NULL,  
    react_trc_id INT NOT NULL,  
    react_cdate DATE NOT NULL,  
    primary key (react_id)  
);
```

Scripts - Populate:

```
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(1, 1, 1, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(2, 2, 2, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(3, 3, 3, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(4, 4, 4, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(5, 5, 5, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(6, 6, 6, CURDATE());
```

```
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(1, 7, 1, CURDATE());  
insert into reaction (react_user_id, react_cmm_id, react_trc_id, react_cdate) values  
(2, 8, 2, CURDATE());
```

comment

A tabela **comment** registra os comentários feitos pelos usuários sobre suas interações com os banheiros. Essa tabela é importante para coletar feedback direto e experiências dos usuários, ajudando a melhorar os serviços.

Colunas	Tipo de Dado	Observação
cmm_id	int	ID do comentário
cmm_int_id	int	ID estrangeiro para a interação
cmm_text	varchar(280)	Texto do comentário
cmm_rclean	int	Nota da categoria "Limpeza"
cmm_rpaper	int	Nota da categoria "Papel"
cmm_rstructure	int	Nota da categoria "Estrutura"
cmm_raccessibility	int	Nota da categoria "Acessibilidade"
cmm_cdatetime	datetime	Data e hora da criação do comentário

As colunas `cmm_rclean`, `cmm_rpaper`, `cmm_rstructure` e `cmm_accessibility`, apresentam esse prefixo "r" de "rating". Além disso, essas categorias podem mudar ao longo do desenvolvimento do projeto, não são definitivas.

Scripts - Create:

```
create table comment (  
    cmm_id INT NOT NULL auto_increment,  
    cmm_int_id INT NOT NULL,  
    cmm_text VARCHAR(280) NOT NULL,  
    cmm_rclean INT NOT NULL,  
    cmm_rpaper INT NOT NULL,  
    cmm_rstructure INT NOT NULL,  
    cmm_raccessibility INT NOT NULL,  
    cmm_cdatetime DATETIME NOT NULL,  
    primary key (cmm_id)  
);
```

Scripts - Populate:

```
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (1, 'This restroom was impressively clean
and accessible. Its spacious and well-maintained, perfect for wheelchair users!', 5,
5, 4, 4, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (1, 'There was no toilet paper available
during my visit. Otherwise, it seemed fine, but supply tracking needs improvement.',
3, 1, 3, 2, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (2, 'The restroom was clean, but the hand
dryer was broken, and I didnt find any paper towels. Regular checks are necessary.',
4, 5, 2, 3, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (3, 'The restroom needs more attention; it
was not very clean, with stains on the floor and an overflowing trash bin.', 2, 4, 2,
4, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (4, 'Accessible but lacking supplies; no
hand sanitizer and an empty soap dispenser. These details matter!', 4, 3, 4, 5,
NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (5, 'Decent experience, but it could be
cleaner, especially around the sinks where water accumulates.', 3, 2, 1, 1, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (6, 'Great overall! Clean, well-stocked, and
excellent accessibility features made my visit comfortable.', 5, 4, 5, 5, NOW());
insert into comment (cmm_int_id, cmm_text, cmm_rclean, cmm_rpaper, cmm_rstructure,
cmm_raccessibility, cmm_cdatetime) values (5, 'Good accessibility but lacking in
supplies. The paper towels were out, and the soap dispenser was low.', 3, 2, 2, 3,
NOW());
```

funcitime

A tabela **funcitime** gerencia os horários de funcionamento das casas de banho, permitindo que os usuários saibam quando um banheiro está acessível. Sua importância está em fornecer informações relevantes que ajudam os usuários a planejem suas visitas.

Colunas	Tipo de Dado	Observação
ft_id	int	ID do horário de funcionamento
ft_toil_id	int	ID estrangeiro para a casa de banho
ft_day_id	int	ID estrangeiro para o dia
ft_state_id	int	ID estrangeiro para o estado da casa de banho nesse horário
ft_timestart	time	Horário de início
ft_timeend	time	Horário de fim
ft_cdate	date	Data da criação do horário de funcionamento

Scripts - Create:

```
create table funcitime (  
    ft_id INT NOT NULL auto_increment,  
    ft_toil_id INT NOT NULL,  
    ft_day_id INT NOT NULL,  
    ft_state_id INT NOT NULL,  
    ft_timestart TIME NOT NULL,  
    ft_timeend TIME NOT NULL,  
    ft_cdate DATE NOT NULL,  
    primary key (ft_id)  
);
```

Scripts - Populate:

```
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (1, 1, 1, '09:00:00', '15:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (1, 1, 1, '18:00:00', '22:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (2, 2, 2, '09:00:00', '17:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (3, 3, 1, '07:00:00', '19:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (4, 4, 2, '06:00:00', '20:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (5, 5, 1, '10:00:00', '16:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (6, 1, 1, '08:00:00', '14:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (6, 3, 2, '10:00:00', '16:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (7, 2, 1, '09:30:00', '15:30:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (7, 4, 2, '11:00:00', '20:00:00', CURDATE());
insert into functime (ft_toil_id, ft_day_id, ft_state_id, ft_timestart, ft_timeend,
ft_cdate) values (8, 5, 1, '12:00:00', '18:00:00', CURDATE());
```

extra

A tabela **extra** relaciona características ou amenidades adicionais disponíveis em casas de banho, como fraldário ou acesso para deficientes. Essa tabela é importante para melhorar a experiência do usuário, fornecendo informações sobre o que cada banheiro oferece além do básico.

Colunas	Tipo de Dado	Observação
extra__id	int	ID do extra
extra_toil_id	int	ID estrangeiro para a casa de banho
extra_tex_id	int	ID estrangeiro para tipo de extra

Scripts - Create:

```
create table extra (  
    extra_id INT NOT NULL auto_increment,  
    extra_toil_id INT NOT NULL,  
    extra_tex_id INT NOT NULL,  
    primary key (extra_id)  
);
```

Scripts - Populate:

```
insert into extra (extra_toil_id, extra_tex_id) values (1, 1);  
insert into extra (extra_toil_id, extra_tex_id) values (2, 2);  
insert into extra (extra_toil_id, extra_tex_id) values (3, 3);  
insert into extra (extra_toil_id, extra_tex_id) values (4, 4);  
insert into extra (extra_toil_id, extra_tex_id) values (5, 5);
```


typereport

A tabela `typereport` define os tipos de denúncias que os usuários podem fazer sobre as casas de banho. Isso é crucial para categorizar feedbacks e facilitar a análise.

Colunas	Tipo de Dado	Observação
trp_id	int	ID do tipo de denúncia da casa de banho
trp_name	varchar(50)	Nome do tipo de denúncia da casa de banho

Scripts - Create:

```
create table typereport (  
    trp_id INT NOT NULL auto_increment,  
    trp_name VARCHAR(50) NOT NULL,  
    primary key (trp_id)  
);
```

Scripts - Populate:

```
insert into typereport (trp_name) values ('False Information');  
insert into typereport (trp_name) values ('Unsanitary Conditions');  
insert into typereport (trp_name) values ('Privacy Violation');  
insert into typereport (trp_name) values ('Maintenance Needed');  
insert into typereport (trp_name) values ('Broken Facilities');  
insert into typereport (trp_name) values ('Other Concerns');
```

typereaction

A tabela **typereaction** lista os tipos de reações que os usuários podem ter em relação a comentários. Sua importância é alta, pois permite a coleta de feedback qualitativo sobre as experiências dos usuários.

Colunas	Tipo de Dado	Observação
trc_id	int	ID do tipo de reação ao comentário
trc_name	varchar(50)	Nome do tipo de reação ao comentário

Scripts - Create:

```
create table typereaction (  
    trc_id INT NOT NULL auto_increment,  
    trc_name VARCHAR(50) NOT NULL,  
    primary key (trc_id)  
);
```

Scripts - Populate:

```
insert into typereaction (trc_name) values ('Liked');  
insert into typereaction (trc_name) values ('Disliked');  
insert into typereaction (trc_name) values ('Not Helpful');  
insert into typereaction (trc_name) values ('Misleading');  
insert into typereaction (trc_name) values ('Inappropriate');  
insert into typereaction (trc_name) values ('Offensive');  
insert into typereaction (trc_name) values ('Spam');  
insert into typereaction (trc_name) values ('Other Concerns');
```

typeextra

A tabela **typeextra** define os tipos de características ou amenidades adicionais que podem estar disponíveis nas casas de banho. É importante para categorizar e descrever os extras que um banheiro pode oferecer.

Colunas	Tipo de Dado	Observação
tex_id	int	ID do tipo de extra da casa de banho
tex_name	varchar(50)	Nome do tipo de extra da casa de banho

Scripts - Create:

```
create table typeextra (  
    tex_id INT NOT NULL auto_increment,  
    tex_name VARCHAR(50) NOT NULL,  
    primary key (tex_id)  
);
```

Scripts - Populate:

```
insert into typeextra (tex_name) values ('Wheelchair Accessible');  
insert into typeextra (tex_name) values ('Baby Changing Station');  
insert into typeextra (tex_name) values ('Handicap Parking');  
insert into typeextra (tex_name) values ('Visual Aids Available');  
insert into typeextra (tex_name) values ('Braille Signage');
```

day

A tabela **day** armazena informações sobre os dias da semana, o que é relevante para gerenciar horários de funcionamento. Sua importância está na estruturação dos horários de operação das casas de banho.

Colunas	Tipo de Dado	Observação
day_id	int	ID do dia
day_name	varchar(50)	Nome do dia

Scripts - Create:

```
create table day (  
    day_id INT NOT NULL auto_increment,  
    day_name VARCHAR(50) NOT NULL,  
    primary key (day_id)  
);
```

Scripts - Populate:

```
insert into day (day_name) values ('Monday');  
insert into day (day_name) values ('Tuesday');  
insert into day (day_name) values ('Wednesday');  
insert into day (day_name) values ('Thursday');  
insert into day (day_name) values ('Friday');  
insert into day (day_name) values ('Saturday');  
insert into day (day_name) values ('Sunday');
```

state

A tabela **state** armazena informações sobre estados, que podem estar relacionados ao status operacional dos banheiros. É fundamental para categorizar e descrever como as casas de banho estão operando em determinados momentos.

Colunas	Tipo de Dado	Observação
state_id	int	ID do tipo de estado da casa de banho em um determinado horário
state_name	varchar(50)	Nome do tipo de estado da casa de banho em um determinado horário

Scripts - Create:

```
create table state (  
    state_id INT NOT NULL auto_increment,  
    state_name VARCHAR(50) NOT NULL,  
    primary key (state_id)  
);
```

Scripts - Populate:

```
insert into state (state_name) values ('Available');  
insert into state (state_name) values ('Closed');  
insert into state (state_name) values ('Temporarily Closed');  
insert into state (state_name) values ('In Maintenance');  
insert into state (state_name) values ('Out of Order');
```

access

A tabela **access** define os tipos de acesso (público, privado e etc) disponíveis em casas de banho, permitindo que usuários com diferentes necessidades encontrem banheiros adequados. Sua importância está em promover inclusão e acessibilidade.

Colunas	Tipo de Dado	Observação
acs_id	int	ID do tipo de acesso a casa de banho
acs_name	varchar(50)	Nome do tipo de acesso a casa de banho

Scripts - Create:

```
create table access (  
    acs_id int not null auto_increment,  
    acs_name VARCHAR(50) not null,  
    primary key (acs_id)  
);
```

Scripts - Populate:

```
insert into access (acs_name) values ('Public');  
insert into access (acs_name) values ('Private');  
insert into access (acs_name) values ('Consumers Only');
```

city

A tabela **city** armazena informações sobre cidades onde as casas de banho estão localizados. É fundamental para a categorização geográfica dos banheiros, facilitando buscas e relatórios em diferentes áreas.

Colunas	Tipo de Dado	Observação
city_id	int	ID da cidade
city_country_id	int	ID estrangeiro para o país
city_name	varchar(50)	Nome da cidade

Scripts - Create:

```
create table city (  
    city_id INT NOT NULL auto_increment,  
    city_country_id INT NOT NULL,  
    city_name VARCHAR(50) NOT NULL,  
    primary key (city_id)  
);
```

Scripts - Populate:

```
insert into city (city_country_id, city_name) values (1, 'Lisbon');  
insert into city (city_country_id, city_name) values (1, 'Porto');  
insert into city (city_country_id, city_name) values (1, 'Braga');  
insert into city (city_country_id, city_name) values (1, 'Coimbra');  
insert into city (city_country_id, city_name) values (2, 'Madrid');  
insert into city (city_country_id, city_name) values (2, 'Barcelona');  
insert into city (city_country_id, city_name) values (2, 'Valencia');  
insert into city (city_country_id, city_name) values (5, 'Rome');  
insert into city (city_country_id, city_name) values (5, 'Milan');
```

country

A tabela **country** contém informações sobre países e é essencial para a estrutura geográfica do sistema. Ela possibilita a organização dos dados de cidade e, por consequência, das casas de banho.

Colunas	Tipo de Dado	Observação
country_id	int	ID do país
country_name	varchar(50)	Nome do país

Scripts - Create:

```
create table country (  
    country_id INT NOT NULL auto_increment,  
    country_name VARCHAR(50) NOT NULL,  
    primary key (country_id)  
);
```

Scripts - Populate:

```
insert into country (country_name) values ('Portugal');  
insert into country (country_name) values ('Spain');  
insert into country (country_name) values ('France');  
insert into country (country_name) values ('Germany');  
insert into country (country_name) values ('Italy');  
insert into country (country_name) values ('Brazil');  
insert into country (country_name) values ('United States');  
insert into country (country_name) values ('Australia');
```