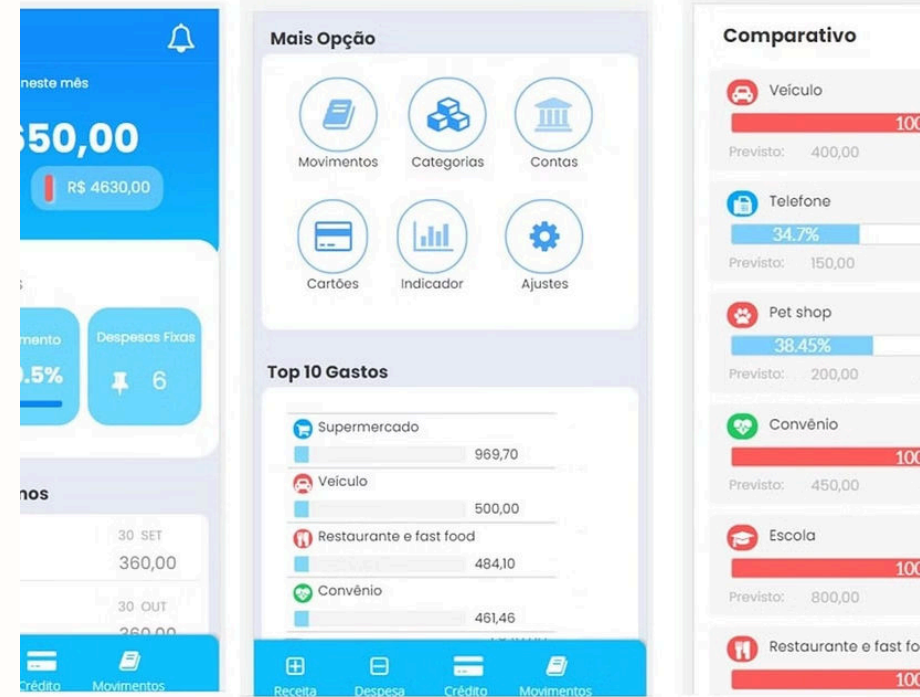


Controle de Gastos

Um aplicativo móvel para controle financeiro pessoal, com recursos de registro de gastos, metas e orçamentos. O usuário precisa de um login com senha para acessar suas informações financeiras.

Ana Beatriz e Thaís



Regras do Aplicativo

1 Usuário

O usuário tem um username (apelido), nome, senha, e-mail e um documento. Ele pode ter 0 ou mais bancos cadastrados, com nome e valor total.

2 Transações

O usuário registra a alocação do dinheiro, associando ou não a uma categoria, o valor, nome da transação e a data (que pode ser editada ou ter valor padrão do dia do registro).

3 Categorias

As categorias têm nome e podem ter subcategorias. Podem ser criadas pelo usuário de acordo com sua própria organização.

4 Orçamentos

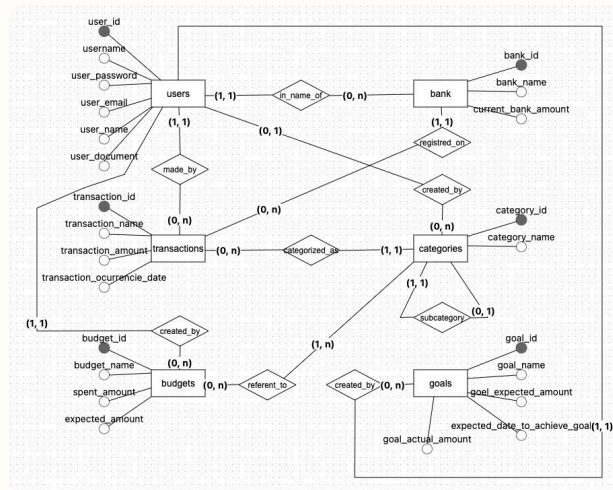
O usuário pode criar orçamentos, com nome, valor gasto, valor esperado, e se refere a uma ou mais categorias.

5 Metas

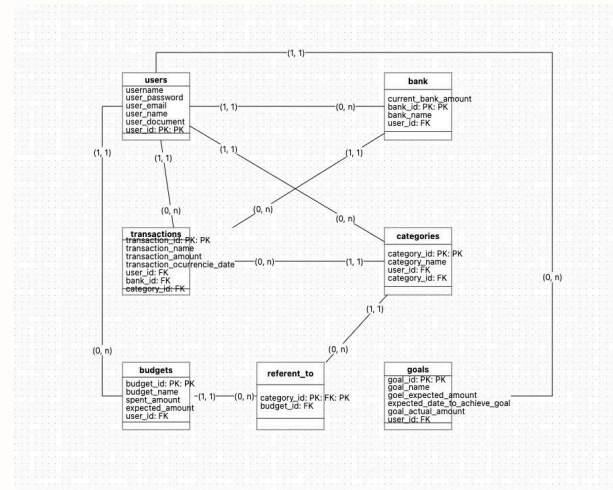
O usuário pode criar metas, com nome, dinheiro esperado para concluí-la, data esperada de cumprimento e dinheiro guardado total.

Modelagem do Banco de Dados

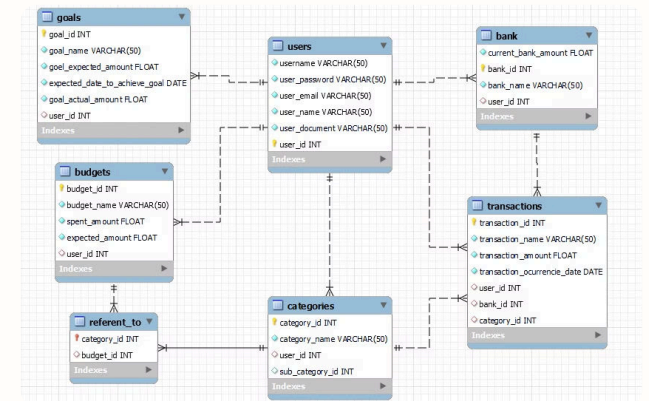
Modelo Conceitual



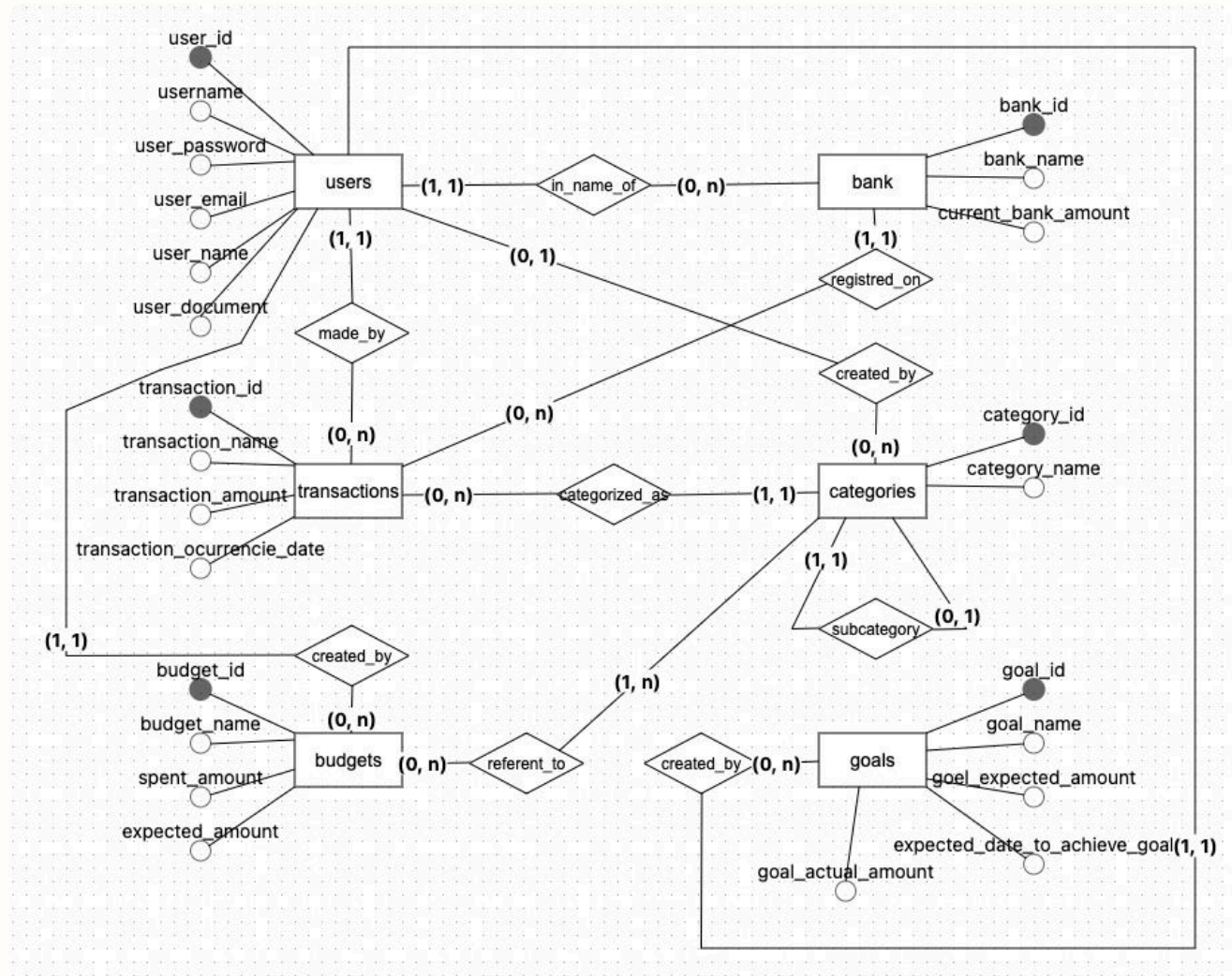
Modelo Lógico



Modelo Físico



Modelo Conceitual



Implementação do Banco de Dados

BRB modelo	MySQL
<pre>CREATE TABLE users (username VARCHAR(n) NOT NULL, user_password VARCHAR(n) NOT NULL, user_email VARCHAR(n) NOT NULL, user_name VARCHAR(n) NOT NULL, user_document VARCHAR(n) NOT NULL, user_id INT PRIMARY KEY AUTO_INCREMENT,); CREATE TABLE bank (current_bank_amount FLOAT NOT NULL, bank_id INT PRIMARY KEY AUTO_INCREMENT, bank_name VARCHAR(n) NOT NULL, user_id INT,); CREATE TABLE categories (category_id INT PRIMARY KEY AUTO_INCREMENT, category_name VARCHAR(n) NOT NULL, user_id INT, category_id INT,); CREATE TABLE transactions (transaction_id INT PRIMARY KEY AUTO_INCREMENT, transaction_name VARCHAR(n) NOT NULL, transaction_amount FLOAT NOT NULL, transaction_ocurrence_date DATE NOT NULL DEFAULT 'CURRENT_TIMESTAMP', user_id INT, bank_id INT, category_id INT,); CREATE TABLE budgets (budget_id INT PRIMARY KEY AUTO_INCREMENT, budget_name VARCHAR(n) NOT NULL, spent_amount FLOAT NOT NULL, expected_amount FLOAT NOT NULL, user_id INT,); CREATE TABLE goals (goal_id INT PRIMARY KEY AUTO_INCREMENT, goal_name VARCHAR(n) NOT NULL, goel_expected_amount FLOAT NOT NULL, expected_date_to_achieve_goal DATE NOT NULL, goal_actual_amount FLOAT NOT NULL, user_id INT,); CREATE TABLE referent_to (category_id INT PRIMARY KEY, budget_id INT,); ALTER TABLE bank ADD FOREIGN KEY(user_id) REFERENCES users (user_id) ALTER TABLE categories ADD FOREIGN KEY(user_id) REFERENCES users (user_id) ALTER TABLE categories ADD FOREIGN KEY(category_id) REFERENCES users (category_id) ALTER TABLE transactions ADD FOREIGN KEY(user_id) REFERENCES users (user_id) ALTER TABLE transactions ADD FOREIGN KEY(bank_id) REFERENCES bank (bank_id) ALTER TABLE transactions ADD FOREIGN KEY(category_id) REFERENCES categories (category_id) ALTER TABLE budgets ADD FOREIGN KEY(user_id) REFERENCES users (user_id) ALTER TABLE goals ADD FOREIGN KEY(user_id) REFERENCES users (user_id) ALTER TABLE referent_to ADD FOREIGN KEY(category_id) REFERENCES categories (category_id) ALTER TABLE referent_to ADD FOREIGN KEY(budget_id) REFERENCES budgets (budget_id)</pre>	<pre>drop database db_trabalho; create database db_trabalho; use db_trabalho; CREATE TABLE user (user_id INT PRIMARY KEY AUTO_INCREMENT, username VARCHAR(50) NOT NULL, user_password VARCHAR(50) NOT NULL, user_email VARCHAR(50) NOT NULL, user_name VARCHAR(50) NOT NULL, user_document VARCHAR(50) NOT NULL); CREATE TABLE bank (bank_id INT PRIMARY KEY AUTO_INCREMENT, current_bank_amount FLOAT NOT NULL, bank_name VARCHAR(50) NOT NULL, user_id INT); CREATE TABLE category (category_id INT PRIMARY KEY AUTO_INCREMENT, category_name VARCHAR(50) NOT NULL, user_id INT, sub_category_id INT); CREATE TABLE transaction (transaction_id INT PRIMARY KEY AUTO_INCREMENT, transaction_name VARCHAR(50) NOT NULL, transaction_amount FLOAT NOT NULL, transaction_ocurrence_date timestamp NOT NULL DEFAULT current_timestamp , user_id INT, bank_id INT, category_id INT); CREATE TABLE budget (budget_id INT PRIMARY KEY AUTO_INCREMENT, budget_name VARCHAR(50) NOT NULL, spent_amount FLOAT NOT NULL, expected_amount FLOAT NOT NULL, user_id INT); CREATE TABLE goal (goal_id INT PRIMARY KEY AUTO_INCREMENT, goal_name VARCHAR(50) NOT NULL, goel_expected_amount FLOAT NOT NULL, expected_date_to_achieve_goal DATE NOT NULL, goal_actual_amount FLOAT NOT NULL, user_id INT); CREATE TABLE referent_to (category_id INT PRIMARY KEY, budget_id INT); ALTER TABLE bank ADD FOREIGN KEY(user_id) REFERENCES user (user_id); ALTER TABLE category ADD FOREIGN KEY(user_id) REFERENCES user (user_id); ALTER TABLE transaction ADD FOREIGN KEY(user_id) REFERENCES user (user_id); ALTER TABLE transaction ADD FOREIGN KEY(bank_id) REFERENCES bank (bank_id); ALTER TABLE transaction ADD FOREIGN KEY(category_id) REFERENCES category (category_id); ALTER TABLE budget ADD FOREIGN KEY(user_id) REFERENCES user (user_id); ALTER TABLE goal ADD FOREIGN KEY(user_id) REFERENCES user (user_id); ALTER TABLE referent_to ADD FOREIGN KEY(category_id) REFERENCES category (category_id); ALTER TABLE referent_to ADD FOREIGN KEY(budget_id) REFERENCES budget (budget_id);</pre>

Criação do Banco de Dados

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	budget_id	budget_name	spent_amount	expected_amount	user_id
▶	1	budget	50	100	1
*	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	transaction_id	transaction_name	transaction_amount	transaction_ocurrence_date	user_id	bank_id	category_id
▶	1	viagem 1/10	50	2024-04-14 13:55:26	1	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: |

	bank_id	current_bank_amount	bank_name	user_id
▶	1	50	Banco Rico	1
*	NULL	NULL	NULL	NULL

Criação do Banco de Dados

Result Grid | Filter Rows: | Edit: | Export/Import:

	user_id	username	user_password	user_email	user_name	user_document
▶	1	tatitata	123	thais@mail.com	Thais Santos	1234567891
*	NULL	NULL	NULL	NULL	NULL	NULL

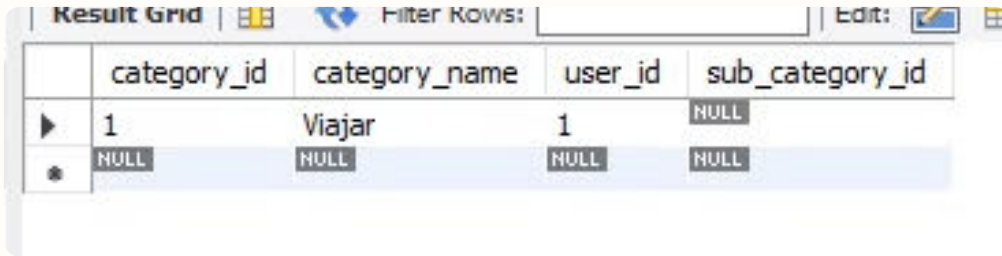
Result Grid | Filter Rows:

	category_id	budget_id
▶	1	1
*	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	goal_id	goal_name	goal_expected_amount	expected_date_to_achieve_goal	goal_actual_amount	user_id
▶	1	Viajar Urubici	2000	2025-01-01	30	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Criação do Banco de Dados



The image shows a screenshot of a database application's 'Result Grid'. The grid has a header row with the following columns: 'category_id', 'category_name', 'user_id', and 'sub_category_id'. The first row of data contains the values '1', 'Viajar', '1', and 'NULL'. The second row of data contains the values 'NULL', 'NULL', 'NULL', and 'NULL'. The second row is highlighted with a light blue background. Above the grid, there is a 'Filter Rows:' field and an 'Edit:' button.

	category_id	category_name	user_id	sub_category_id
▶	1	Viajar	1	NULL
✱	NULL	NULL	NULL	NULL

Conclusão e Referências



Ajustes Necessários

Durante a modelagem do banco, foi preciso fazer vários ajustes no script para que ele se adequasse à evolução das regras de negócio.



Modelagem Iterativa

No início, definimos as regras e depois modelamos o banco, mas mesmo assim foi preciso adequar tanto o modelo conceitual quanto o lógico para que o modelo físico se adequasse às expectativas.



Referências Utilizadas

Foram utilizadas as seguintes referências durante o processo de modelagem e implementação do banco de dados: BRModelo, Gamma e MySQL.