

Trabalho final de Banco de Dados I

Professor:

- Alexandre Leopoldo Gonçalves

Alunos:

- Matheus Francisco Rodrigues Lima (19104064)
- Rômulo Emanuel Mendonça Schiavon (19102490)

Sumário

Trabalho final de Banco de Dados I	0
O que é o VAPOR:	2
Como executar o código:	2
Requisitos:	3
1. Descrição do objetivo geral do sistema	3
2. Uma descrição mais detalhada do sistema	4
3. Inclusão da modelagem conceitual	5
4. Inclusão da modelagem lógica	6
5. Inclusão do script DDL obtido a partir do item 4	7
6. Gráficos relacionados às consultas	12

O que é o VAPOR:

Vapor é uma loja online voltada para o público gamer. Vapor não é somente uma loja de jogos e expansões como também é uma loja de itens colecionáveis. O vapor conecta o jogador ao seu jogo preferido, independente da classificação, categoria, linguagem, etc... O vapor mostra para o seu usuário tudo aquilo que ele precisa saber sobre o jogo. Ou seja, é uma loja gamer feita por gamers para gamers.

O vapor também tem comentários avaliativos sobre os jogos, ou seja, no vapor você consegue ver quais jogos os usuários mais gostaram e assim aumenta a sua chance de gostar de determinado jogo.

O vapor mostra e deixa visível para os usuários também a distribuidora dos jogos e os seus desenvolvedores, ou seja, o vapor dá o crédito aos criadores dos jogos pelas suas criações. Ajudando a movimentar a indústria dos jogos!

Como executar o código:

Instalar as seguintes bibliotecas:

- [Matplotlib](#)
- [NumPy](#) (vem com o matplotlib, mas podem haver erros)
- [Pycopg2](#)

Após instalar executar em seu prompt, alterar as variáveis para o banco de dados no arquivo `db.py`.

Depois execute o `main.py` no seu prompt utilizando:

```
python main.py
```

Para criar o banco, inserir, atualizar e realizar exclusões de dados utilize a interface gráfica. Todos os "returns" exigidos nos requisitos terão sua saída direto no prompt (prints no console).

As consultas funcionam mesmo que as atualizações e exclusões não tenham sido realizadas no banco, no entanto os gráficos gerados podem conter muitos atributos e a leitura poderá ser difícil.

Requisitos:

1. Descrição do objetivo geral do sistema

O sistema tem o propósito de ser uma loja de jogos e itens relacionados a jogos, ou seja, o objetivo principal do sistema é que com as informações das compras dos usuários possamos realizar análises sobre o consumo do usuário, quais os itens mais comprados, os jogos mais comprados, o preço médio de jogos, expansões e etc..

2. Uma descrição mais detalhada do sistema

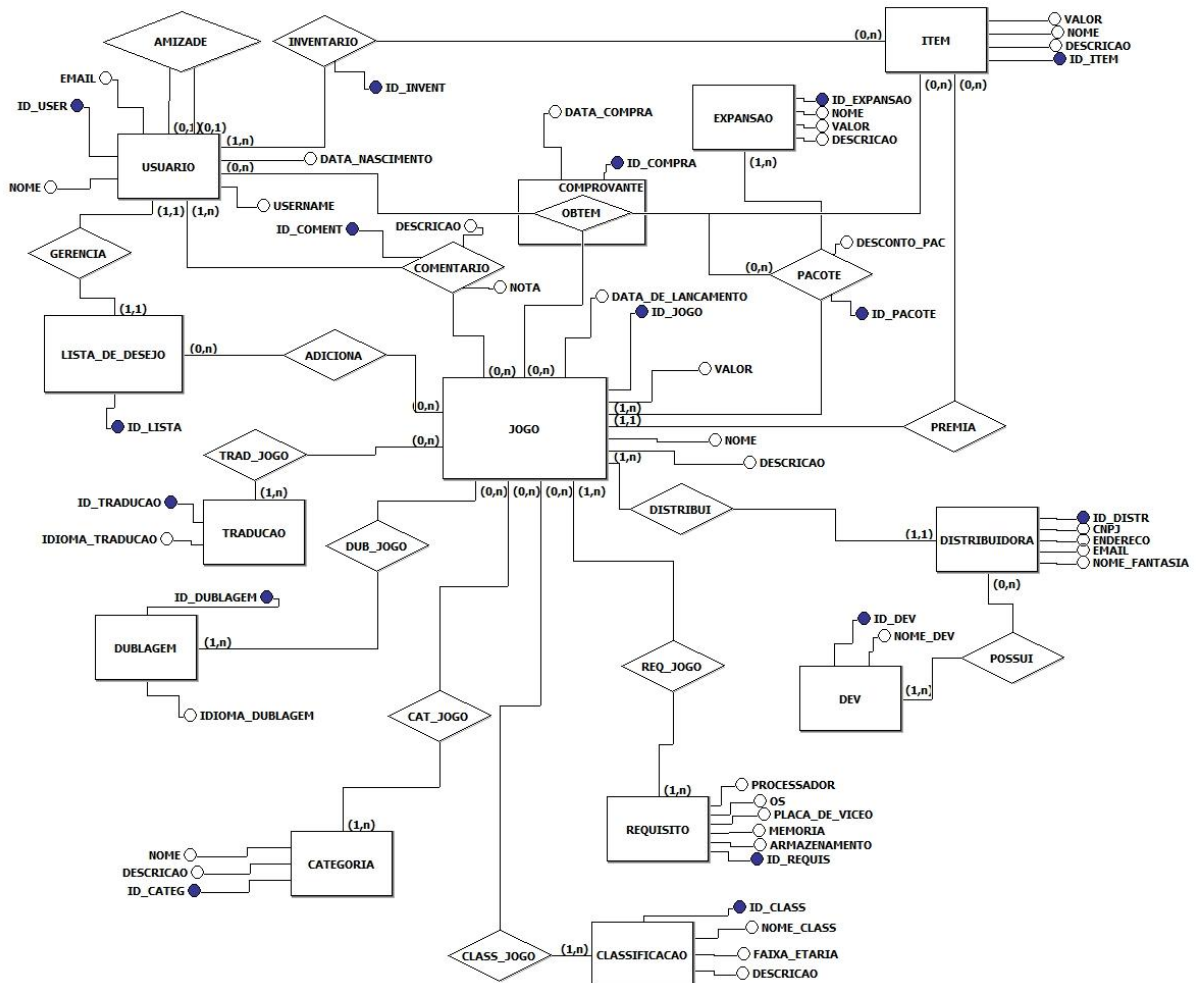
O sistema representado será do funcionamento de uma plataforma online de jogos, onde se pode pesquisar, favoritar e comprar jogos de todas as categorias, classificações, tradução, dublagens disponíveis e requisitos para rodar o jogo. As informações sobre os jogos devem ter seu nome, valor para venda na loja, descrição, data de lançamento assim como ao jogo devemos subentender que este possui ao menos uma das características descritas acima. Entende-se que para a dublagem e a tradução o mais importante é descrever o idioma que se encontra o atributo. Ao classificar um jogo entende-se que é necessário identificá-la apenas por um atributo identificador, no entanto requer-se também o seu nome, faixa etária de quem pode jogar o jogo e descrição sobre o mesmo. A categoria do jogo é identificada por um atributo identificador, descrição e nome. E os requisitos de um jogo estão definidos como um atributo identificador, sistema operacional, processador, placa de vídeo, memória RAM disponível no sistema e memória de armazenamento. Também é possível a compra de uma ou mais expansões para um jogo, ou seja, conteúdo adicional que jogadores podem adquirir para melhorar sua experiência. A expansão tem nome, valor e descrição. E ao menos uma expansão em conjunto com o jogo base, o jogador obtém um pacote, ou seja, ele receberá um valor promocional. Este valor será então pago pelo jogo base e os valores adicionais de expansão. Caso o usuário já obtenha qualquer item que faça parte do pacote promocional, este valor não é incluso.

Dentro do sistema existirão usuários, e esses são definidos pelas informações de email, username, nome completo opcional e data de nascimento. Além disso, o usuário poderá adicionar jogos dentro da sua lista de desejos, podendo assim retirar uma média de preço e preço total de seus interesses. O usuário poderá se relacionar com outros usuários com pedidos de amizade, associando entre si. Ao jogar um jogo, o usuário receberá itens colecionáveis, e estes podem ser negociados na loja. Um item é identificado por valor, nome e descrição. Os itens são armazenados em um inventário pessoal do usuário, que sumariza o valor total de seus itens. Também será possível a avaliação do jogo, deixando uma nota e um comentário, que poderão ser rastreados diretamente ao usuário.

Ao comprar jogos, itens ou pacotes é gerado um comprovante, que relaciona-se ao usuário, jogo e item de forma que seja possível comprar diversos itens, jogos ou pacotes em um mesmo comprovante. No comprovante deve-se existir a data da compra, atributo identificador para o comprovante e o valor da compra. Cada jogo possui uma distribuidora, podendo essa representar diversos jogos, que é identificada pelo nome, CNPJ, endereço e email e, por sua vez, disponibiliza dos devs responsáveis pela criação dos jogos.

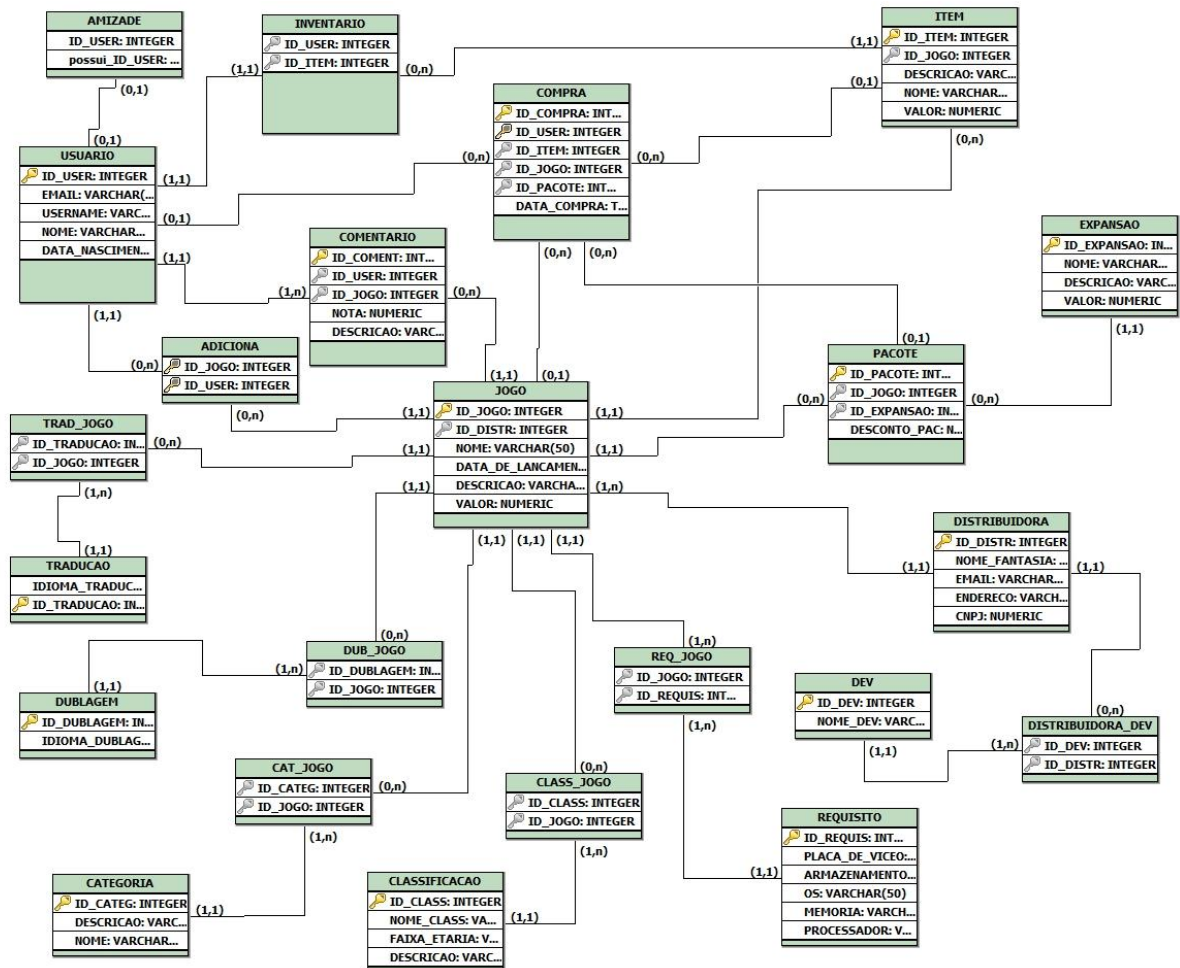
3. Inclusão da modelagem conceitual

Item anexado no arquivo compactado .zip (ModeloConceitual.brM).



4. Inclusão da modelagem lógica

Item anexado no arquivo compactado .zip (ModeloLogico.brM).



5. Inclusão do script DDL obtido a partir do item 4

```
CREATE TABLE DUBLAGEM (  
  ID_DUBLAGEM SERIAL PRIMARY KEY,  
  IDIOMA_DUBLAGEM VARCHAR(20)  
);
```

```
CREATE TABLE CLASSIFICACAO (  
  ID_CLASS SERIAL PRIMARY KEY,  
  NOME_CLASS VARCHAR(20),  
  FAIXA_ETARIA VARCHAR(4),  
  DESCRICAO VARCHAR(200)  
);
```

```
CREATE TABLE DISTRIBUIDORA (  
  ID_DISTR SERIAL PRIMARY KEY,  
  NOME_FANTASIA VARCHAR(30),  
  EMAIL VARCHAR(50) UNIQUE,  
  ENDERECO VARCHAR(50),  
  CNPJ NUMERIC UNIQUE  
);
```

```
CREATE TABLE ITEM (  
  ID_ITEM SERIAL PRIMARY KEY,  
  ID_JOGO INTEGER,  
  DESCRICAO VARCHAR(200),  
  NOME VARCHAR(50),  
  VALOR NUMERIC  
);
```

```
CREATE TABLE USUARIO (  
  ID_USER SERIAL PRIMARY KEY,  
  EMAIL VARCHAR(100) UNIQUE,  
  USERNAME VARCHAR(20),  
  NOME VARCHAR(50),  
  DATA_NASCIMENTO TIMESTAMP  
);
```

```
CREATE TABLE CATEGORIA (  
  ID_CATEG SERIAL PRIMARY KEY,  
  DESCRICAO VARCHAR(200),
```



```
NOME VARCHAR(30)
);
```

```
CREATE TABLE JOGO (
ID_JOGO SERIAL PRIMARY KEY,
ID_DISTR INTEGER,
NOME VARCHAR(50),
DATA_DE_LANCAMENTO TIMESTAMP,
DESCRICAO VARCHAR(500),
VALOR NUMERIC,
FOREIGN KEY(ID_DISTR) REFERENCES DISTRIBUIDORA (ID_DISTR) ON UPDATE
CASCADE ON DELETE SET NULL
);
```

```
CREATE TABLE REQUISITO (
ID_REQUIS SERIAL PRIMARY KEY,
PLACA_DE_VIDEO VARCHAR(50),
ARMAZENAMENTO VARCHAR(100),
OS VARCHAR(50),
MEMORIA VARCHAR(50),
PROCESSADOR VARCHAR(50)
);
```

```
CREATE TABLE EXPANSAO (
ID_EXPANSAO SERIAL PRIMARY KEY,
NOME VARCHAR(50),
DESCRICAO VARCHAR(200),
VALOR NUMERIC
);
```

```
CREATE TABLE TRADUCAO (
IDIOMA_TRADUCAO VARCHAR(20),
ID_TRADUCAO SERIAL PRIMARY KEY
);
```

```
CREATE TABLE DEV (
ID_DEV SERIAL PRIMARY KEY,
NOME_DEV VARCHAR(50)
);
```

```
CREATE TABLE AMIZADE (  
  ID_USER INTEGER,  
  possui_ID_USER INTEGER  
);
```

```
CREATE TABLE COMPRA (  
  ID_USER INTEGER,  
  ID_ITEM INTEGER NULL,  
  ID_JOGO INTEGER NULL,  
  ID_PACOTE INTEGER NULL,  
  ID_COMPRA SERIAL UNIQUE,  
  DATA_COMPRA TIMESTAMP,  
  PRIMARY KEY(ID_COMPRA),  
  FOREIGN KEY(ID_USER) REFERENCES USUARIO (ID_USER) ON UPDATE  
  CASCADE ON DELETE SET NULL,  
  FOREIGN KEY(ID_ITEM) REFERENCES ITEM (ID_ITEM) ON UPDATE CASCADE ON  
  DELETE SET NULL,  
  FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
  ON DELETE SET NULL  
);
```

```
CREATE TABLE LISTA_DE_DESEJOS (  
  ID_JOGO INTEGER,  
  ID_USER INTEGER,  
  PRIMARY KEY(ID_JOGO,ID_USER),  
  FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
  ON DELETE CASCADE,  
  FOREIGN KEY(ID_USER) REFERENCES USUARIO (ID_USER) ON UPDATE  
  CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE DISTRIBUIDORA_DEV (  
  ID_DEV INTEGER,  
  ID_DISTR INTEGER,  
  FOREIGN KEY(ID_DEV) REFERENCES DEV (ID_DEV) ON UPDATE CASCADE ON  
  DELETE CASCADE,  
  FOREIGN KEY(ID_DISTR) REFERENCES DISTRIBUIDORA (ID_DISTR) ON UPDATE  
  CASCADE ON DELETE CASCADE  
);
```

```

CREATE TABLE INVENTARIO (
ID_USER INTEGER,
ID_ITEM INTEGER,
FOREIGN KEY(ID_USER) REFERENCES USUARIO (ID_USER) ON UPDATE
CASCADE ON DELETE SET NULL,
FOREIGN KEY(ID_ITEM) REFERENCES ITEM (ID_ITEM) ON UPDATE CASCADE ON
DELETE CASCADE
);

```

```

CREATE TABLE COMENTARIO (
ID_COMENT SERIAL PRIMARY KEY,
ID_USER INTEGER,
ID_JOGO INTEGER,
NOTA NUMERIC,
DESCRICAO VARCHAR(200),
FOREIGN KEY(ID_USER) REFERENCES USUARIO (ID_USER) ON UPDATE
CASCADE ON DELETE CASCADE,
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE
ON DELETE CASCADE
);

```

```

CREATE TABLE PACOTE (
ID_PACOTE SERIAL PRIMARY KEY,
ID_JOGO INTEGER,
ID_EXPANSAO INTEGER,
DESCONTO_PAC NUMERIC,
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE
ON DELETE SET NULL,
FOREIGN KEY(ID_EXPANSAO) REFERENCES EXPANSAO (ID_EXPANSAO) ON
UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE CLASS_JOGO (
ID_CLASS INTEGER,
ID_JOGO INTEGER,
FOREIGN KEY(ID_CLASS) REFERENCES CLASSIFICACAO (ID_CLASS) ON UPDATE
CASCADE ON DELETE CASCADE,
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE
ON DELETE CASCADE
);

```

```

CREATE TABLE TRAD_JOGO (

```

```
ID_TRADUCAO INTEGER,  
ID_JOGO INTEGER,  
FOREIGN KEY(ID_TRADUCAO) REFERENCES TRADUCAO (ID_TRADUCAO) ON  
UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE DUB_JOGO (  
ID_DUBLAGEM INTEGER,  
ID_JOGO INTEGER,  
FOREIGN KEY(ID_DUBLAGEM) REFERENCES DUBLAGEM (ID_DUBLAGEM) ON  
UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE CAT_JOGO (  
ID_CATEG INTEGER,  
ID_JOGO INTEGER,  
FOREIGN KEY(ID_CATEG) REFERENCES CATEGORIA (ID_CATEG) ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE REQ_JOGO (  
ID_JOGO INTEGER,  
ID_REQUIS INTEGER,  
FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO) ON UPDATE CASCADE  
ON DELETE CASCADE,  
FOREIGN KEY(ID_REQUIS) REFERENCES REQUISITO (ID_REQUIS) ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```

```
ALTER TABLE ITEM ADD FOREIGN KEY(ID_JOGO) REFERENCES JOGO (ID_JOGO)  
ON UPDATE CASCADE ON DELETE CASCADE;  
ALTER TABLE COMPRA ADD FOREIGN KEY(ID_PACOTE) REFERENCES PACOTE  
(ID_PACOTE) ON UPDATE CASCADE ON DELETE SET NULL;
```

6. Gráficos relacionados às consultas

Os gráficos foram gerados após as operações de Insert, Update e Delete no Banco de dados

***Os gráficos de pizza podem mudar de cor, mas mantém os dados ***

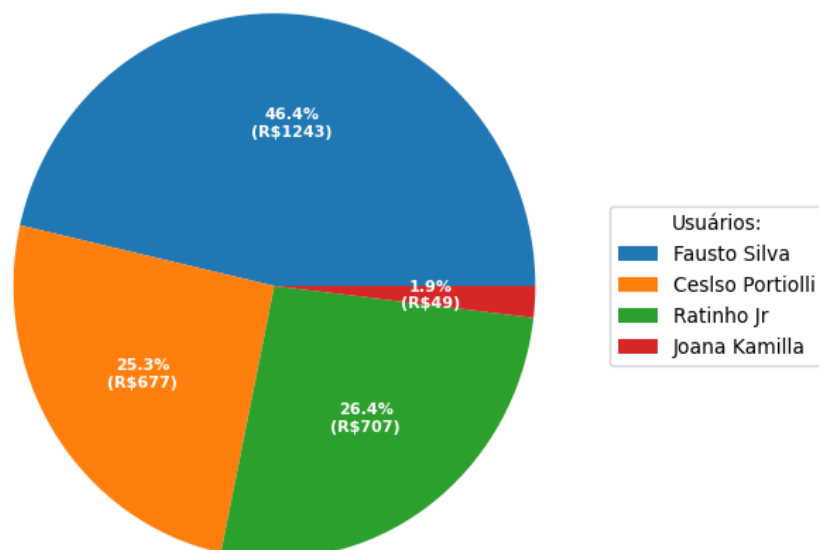
Consulta 1:

O objetivo da consulta 1 é relacionar os usuários com o preço médio de seus inventários, ou seja, a média do valor de todos os seus itens colecionáveis. Relacionando então três tabelas.

O resultado então fica assim (considerando consulta após update/delete)

	nome character varying (50)	valorinvent numeric
1	Joana Kamilla	50.0000000000000000
2	Ceslso Portioli	677.2000000000000000
3	Ratinho Jr	707.2000000000000000
4	Fausto Silva	1243.0000000000000000

Preço Médio dos Inventários

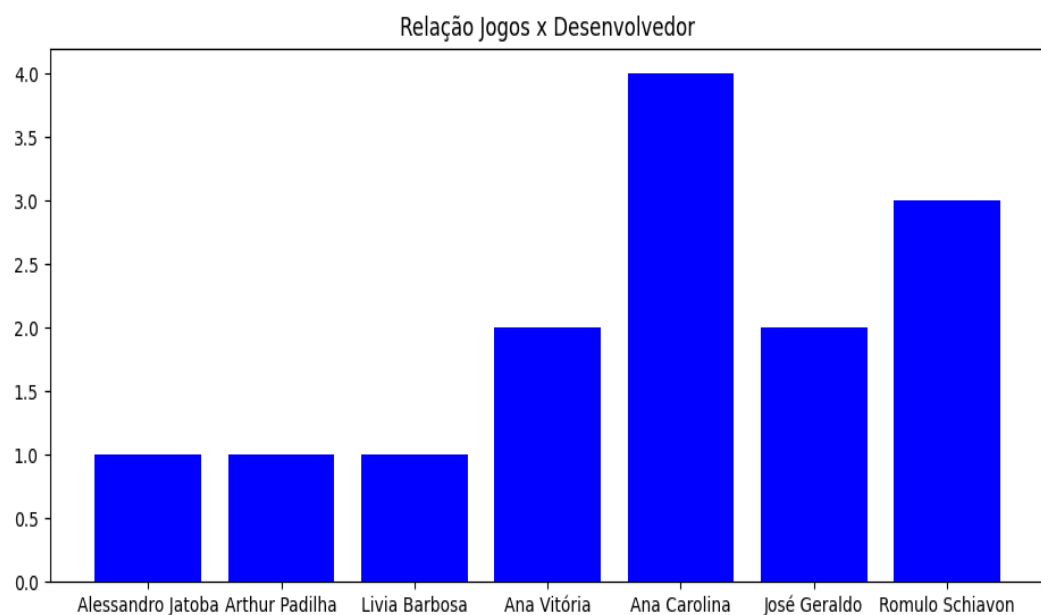


Consulta 2:

O objetivo da consulta 1 é relacionar os desenvolvedores com a quantidade de jogos que cada um fez , ou seja, a soma dos jogos que cada distribuidora fez com os desenvolvedores relacionados a ela.

O resultado então fica assim (considerando consulta após update/delete)

	quant_jogos bigint	desenvolvedor character varying (50)
1	2	Alessandro Jatoba
2	1	Arthur Padilha
3	1	Livia Barbosa
4	2	Ana Vitória
5	4	Ana Carolina
6	2	José Geraldo
7	3	Romulo Schiavon



Consulta 3:

O objetivo da consulta 3 é relacionar a lista de desejos dos usuários, e o seu valor. Podemos obter então o valor total que os usuários da aplicação VAPOR pretendem gastar no futuro.

O resultado então fica assim (considerando consulta após update/delete)

	nomes character varying (50)	valorfuturo numeric
1	Ceslso Portioli	450
2	Fausto Silva	1130
3	Kleber de Paula Pedra	180
4	Maria das Dores	380
5	Ratinho Jr	500
6	Zeca Margo	1000

Preço total da Lista de Desejos

