

UNIVERSIDADE ESTADUAL PAULISTA “JULIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS E TECNOLOGIA
CIÊNCIA DA COMPUTAÇÃO

BANCO DE DADOS I
Ronaldo Celso Messias Correia

E-COMMERCE DE JOGOS USADOS

Gabriel Cecon Carlsen
David Junior Rodrigues
Pedro Henrique Zago Costa

PRESIDENTE PRUDENTE

2021

ÍNDICE

1) ESPECIFICAÇÃO DO PROBLEMA.....	
2) ESQUEMA CONCEITUAL.....	
<i>i. Modelo Entidade Relacionamento (MER)</i>	
3) ESQUEMA RELACIONAL.....	
4) NORMALIZAÇÃO.....	
5) ESPECIFICAÇÕES DE CONSULTAS.....	
<i>i. Álgebra relacional</i>	
<i>ii. SQL</i>	
6) IMPLEMENTAÇÃO DA BASE DE DADOS.....	
<i>i. Criação da Base de Dados</i>	
<i>ii. Alteração da Base de Dados</i>	
<i>iii. Alimentação Inicial da Base de Dados</i>	
7) PROTÓTIPO (INTERFACES)	

ESPECIFICAÇÃO DO PROBLEMA

Empresa: DPG games.

Objetivo: Modelagem realizada para a resolução de um problema do mundo real: altos preços de jogos no Brasil. Jogos eletrônicos recentemente lançados lucram valores incompatíveis com a realidade da maioria dos brasileiros, em grande escala desenvolvidos e produzidos fora do país, sofrem com a conversão dólar-real e acréscimos de carga tributária de aproximadamente 72% do valor original (IPI, ICMS, Confins, PIS, ISSQN). De forma contrária, jogos mais antigos são vendidos como relíquias, pela sua nostalgia e baixa quantidade no mercado, resultando em preços semelhantes aos lançamentos “triple A”, ou seja, o brasileiro médio encontra-se em uma “sinuca de bico”. Uma das soluções mais procuradas é a compra de jogos usados por parte dos gamers, todavia nos encontramos em um novo problema, plataformas eletrônicas de vendas como MercadoLivre, OLX, Marketplace (Facebook) resultam ao consumidor a falta de dinâmica na intermediação comprador/vendedor, gerando burocracia, longos tempos de transporte de mercadoria, assim como custos extras com frete. Nossa solução baseia-se em um aplicativo/e-commerce onde vendedores e compradores cadastram seus dados/localização e anunciam seus produtos, você como comprador busca pelos jogos usados em sua região (raio de quilômetros), entra em contato com o vendedor podendo negociar preço, localidade, formas de identificação (Facebook/Instagram) para confiabilidade, entre outros aspectos. O objetivo principal da nossa aplicação é aproximar, facilitar e desburocratizar a relação de vendedor e comprador referente a jogos usados no Brasil.

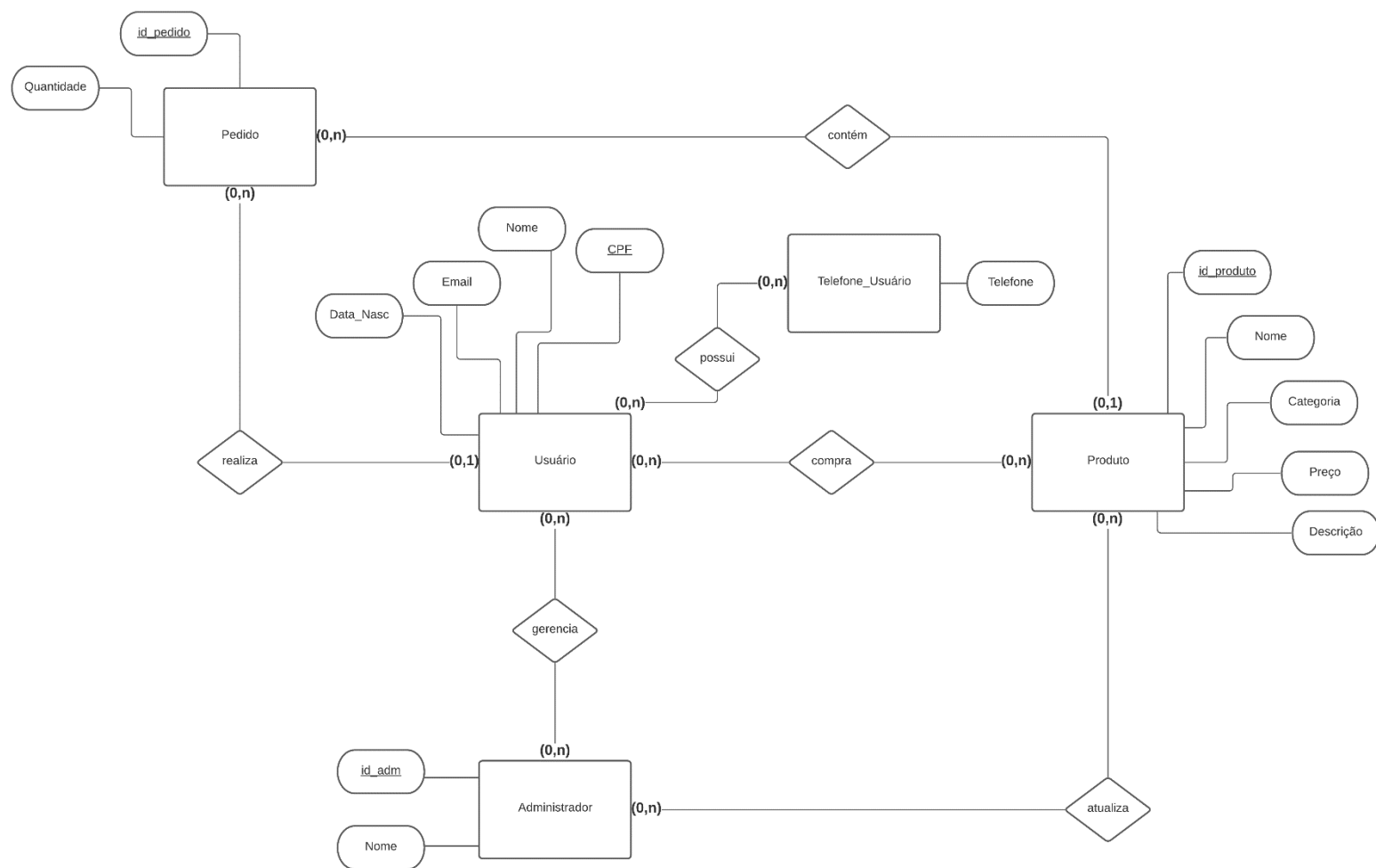
Resumidamente, nossa aplicação baseia-se no conceito de uma rede eletrônica de vendas básica, onde múltiplos usuários poderão adicionar publicações concernentes a jogos usados. Cada publicação deve ter uma imagem, título, resumo e condição, tal publicação pertence a um vendedor que poderá se relacionar com um comprador. Logo temos as relações usuário e publicação, comprador/vendedor e transação, com múltiplos atributos identificadores. Usuários poderão filtrar publicações (jogos usados) por diferentes características.

Obs. As informações disponibilizadas podem sofrer alterações durante o desenvolvimento do projeto.

ESQUEMA CONCEITUAL

Essa seção deve exibir o esquema conceitual (ou seja, o modelo entidade relacionamento) para o problema do mundo real sendo analisado.

Modelo Entidade Relacionamento (MER)



ESQUEMA RELACIONAL

Nessa seção deve ser identificado o conjunto de relações que especificam o banco de dados relacional a ser implementado. Para tanto, deve ser realizado o mapeamento do esquema conceitual para o esquema relacional.

Consiste em um conjunto de oito etapas, bem definidas, mapeando esquema conceitual para o esquema relacional.

1) Analisar cada entidade forte;

Usuário (CPF (PK), Nome, Email, Telefone, Data_Nasc)

Pedido (id_pedido (PK), Quantidade)

Produto (id_produto (PK), Nome, Marca, Preço, Descrição)

Administrador (id_admin (PK), Nome)

2) Analisar cada entidade fraca;

(0 entidades fracas)

3) Regular cardinalidade 1:1;

(0 cardinalidades 1:1)

4) Regular cardinalidade 1:n;

Pedido (id_pedido (PK), id_produto (FK), CPF (FK), Quantidade)

5) Regular cardinalidade n:n

Compra (CPF, id_produto)

Gerencia (CPF, id_admin)

Atualiza (id_admin, id_produto)

Obs. Toda cardinalidade n:n gera agregação.

6) Verificar relacionamentos com grau > 2;

(0 relacionamento com grau > 2)

7) Tratar atributos multivalorados;

Telefone_Usuário (CPF, Telefone)

8) Mapeamento de especializações;

(0 especializações)

NORMALIZAÇÃO

Essa seção deve, para cada uma das relações especificadas na seção anterior, identificar se elas se encontram ou não na Terceira Forma Normal (3FN).

Primeira Forma Normal (1FN): Atributos necessitam ser atômicos e monovalorados.

Usuário (CPF (PK), Nome, Email, Data_Nasc)

Pedido (id_pedido (PK), id_produto (FK), CPF (FK), Quantidade)

Produto (id_produto (PK), Nome, Marca, Preço, Descrição)

Administrador (id_admin (PK), Nome)

Telefone_Usuário (CPF (FK), Telefone)

Compra (CPF, id_produto)

Gerencia (CPF, id_admin)

Atualiza (id_admin, id_produto)

Todas as relações estão na 1FN, pois não há nenhum atributo não-atômico e não-monovalorado.

Segunda Forma Normal (2FN): Estar na 1FN e todos os atributos não-chave são funcionalmente dependentes de todas as partes da chave primária.

Todas as tabelas já estão normalizadas segundo a 2FN, pelo fato de não existir nenhuma chave primária composta no Modelo Entidade Relacionamento.

Terceira Forma Normal (3FN): Estar na 2FN e não existir atributos não-chave que sejam dependentes de outros atributos não-chave, ou seja, não existir dependência transitiva.

Todas as tabelas já estão normalizadas segundo a 3FN. Todos os atributos não-chave das relações dependem apenas de sua chave primária (PK), logo não existe dependência transitiva.

Conclusão: De fato, quando a modelagem entidade-relacionamento é bem projetada, é gerado de forma natural relações normalizadas na 3FN. Logo, o esquema conceitual apresenta-se extremamente importante ao projetar-se uma aplicação utilizando conceitos de banco de dados, poupando tempo e atividades que posteriormente mostraram-se desnecessárias (normalização).

ESPECIFICAÇÕES DE CONSULTAS EM ÁLGEBRA E SQL

1) Selecione o cpf e nome dos usuários, onde o nome começa com os caracteres "Ga".

SQL

```
SELECT cpf, nome FROM usuario  
WHERE nome LIKE 'Ga%';
```

ÁLGEBRA

π cpf, nome σ nome LIKE 'Ga%' (usuario)

2) Selecione o nome e preço dos produtos, onde o preço está entre 60.00 e 90.00 reais.

SQL

```
SELECT nome, preco FROM produto  
WHERE preco BETWEEN 60.00 AND 90.00;
```

ÁLGEBRA

π nome, preco σ preco \geq 60.00 \wedge preco \leq 90.00 (produto)

3) Selecione todos os id's dos pedidos.

SQL

```
SELECT id_pedido FROM pedido;
```

ÁLGEBRA

π id_pedido (pedido)

4) Selecione todos os usuários nascidos a partir do dia um de janeiro de mil novecentos e noventa e nove (01/01/1999).

SQL

```
SELECT * FROM usuario  
WHERE data_nasc >= '1999-01-01';
```

ÁLGEBRA

σ data_nasc \geq DATE('1999-01-01') (usuario)

5) Selecione o id e nome de todos os administradores.

SQL

```
SELECT id_adm, nome FROM administrador;
```

ÁLGEBRA

π id_adm, nome (administrador)

IMPLEMENTAÇÃO DA BASE DE DADOS

Criação da Base de Dados: *Script*, documentado, com os comandos SQL para a criação da base de dados, de acordo com o esquema lógico.

```
create database dpg_games;                                # Criação do banco de dados
use dpg_games;                                           # Selecionar o banco de dados

create table administrador (                             # Criação da tabela administrador
  id_adm integer not null auto_increment primary key,
  nome varchar(40) not null
) ENGINE = innodb;

create table usuario (                                   # Criação da tabela usuário
  cpf varchar(11) not null primary key,
  nome varchar(40) not null,
  email varchar(100) not null,
  id_adm integer not null,
  data_nasc date not null) ENGINE = innodb;

create table telefone_usuario (                          # Criação da tabela de telefones
  cpf varchar(11),
  telefone varchar(20)
) ENGINE = InnoDB;

create table produto (                                   # Criação da tabela produto
  id_produto integer not null auto_increment primary key,
  nome varchar(40) not null,
  categoria varchar(20) not null,
  preco decimal(5,2) not null,
  id_adm integer not null,
  cpf varchar(11) not null,
  descricao text) ENGINE = innodb;

create table pedido (                                    # Criação da tabela pedido
  id_pedido integer not null auto_increment primary key,
  quantidade integer not null,
  id_produto integer not null,
  cpf varchar(11) not null
) ENGINE = innodb;
```

Alteração da Base de Dados: Após a criação da estrutura do nosso banco de dados, modificamos as tabelas para a implementação de suas relações (atribuição das chaves estrangeiras).

```

alter table usuario
add constraint fk_id_adm foreign key (id_adm) references administrador(id_adm);           # RELAÇÃO GERENCIA

alter table telefone_usuario
add constraint fk_cpf foreign key (cpf) references usuario(cpf);                       # RELAÇÃO POSSUI

alter table produto
add constraint fk_cpf foreign key (cpf) references usuario(cpf),                       # RELAÇÃO COMPRA
add constraint fk_id_adm foreign key (id_adm) references administrador(id_adm);        # RELAÇÃO ATUALIZA

alter table pedido
add constraint fk_produto foreign key (id_produto) references produto(id_produto),     # RELAÇÃO CONTÉM
add constraint fk_cpf foreign key (cpf) references usuario(cpf);

```

Alimentação inicial da Base de Dados: *Script*, documentado, com os comandos SQL para a alimentação inicial da base de dados, com no mínimo 5 tuplas por tabela.

```

insert into administrador (nome) values
('GabrielCecon'),
('DavidJr'),
('PedroZago'),
('DavidOno'),
('RonaldoCelso');

insert into usuario (cpf, nome, email, id_adm, data_nasc) values
('45673677881', 'Joe Rogan', 'joerogan@unesp.br', 1, '1967-08-11'),
('12345678910', 'Liam Smith', 'liamsmith@unesp.br', 1, '1999-07-04'),
('34528746781', 'Emma Watson', 'emmawatson@unesp.br', 1, '1975-07-12'),
('74633278571', 'James Lopes', 'jameslopes@unesp.br', 1, '1998-03-22'),
('98364736217', 'Noah Johnson', 'noahjonson@unesp.br', 1, '1997-03-01'),
('32145667542', 'Olivia Brown', 'oliviabrown@unesp.br', 1, '1985-07-31'),
('98574837218', 'Emma Willians', 'emmawillians@unesp.br', 1, '2002-08-07'),
('95809001881', 'Ronnie Coleman', 'ronniecoleman@unesp.br', 1, '1964-05-13'),
('11223344567', 'Jordan B. Peterson', 'jordanbpeterson@unesp.br', 1, '1962-06-12'),
('09899776623', 'Matthew McConaughey', 'matthewmcconaughey@unesp.br', 1, '1969-11-04');

insert into telefone_usuario (cpf, telefone) values
('12345678910', '18994283818'),
('98364736217', '18997922541'),
('32145667542', '1145341273'),
('98574837218', '11976584763'),
('74633278571', '11989776554'),
('34528746781', '15838812389'),
('95809001881', '15984903890'),
('11223344567', '11854839839'),
('45673677881', '11112233445'),
('09899776623', '11890900987');

```

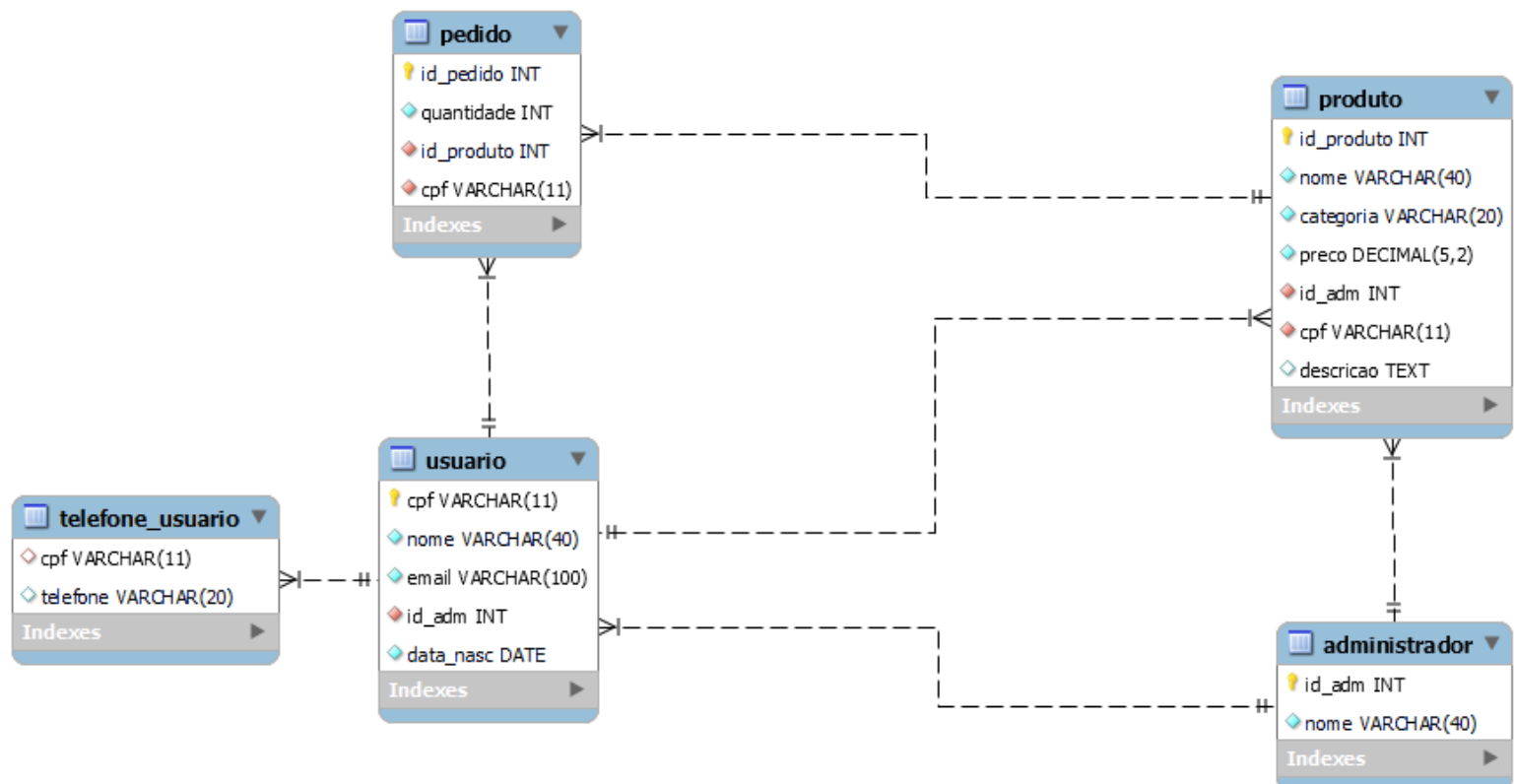
```

insert into produto (nome, categoria, preco, id_adm, cpf, descricao) values
('God of War', 'Santa Monica', 60.00, 1, '12345678910', 'Deus bom de guerra' ),
('Mortal Kombat 11', 'Nether Realm', 120.00, 1, '12345678910', 'Lutinha com excesso de sangue'),
('The Last Of Us 2', 'Naughty Dog', 200.00, 1, '12345678910', 'O segundo melhor jogo da história'),
('Grand Theft Auto V', 'Rock Star', 50.00, 1, '12345678910', 'Revolucionário'),
('Red Dead Redemption II', 'Rock Star', 150.00, 1, '12345678910', 'O terceiro melhor jogo da história'),
('Batman: Return to Arkham', 'Warner Bros', 40.00, 1, '12345678910', 'Bilionário órfão'),
('Cyberpunk 2077', 'CD Project Red', 170.00, 1, '12345678910', 'Quase bugou para inserir kkk'),
('Spider-Man', 'Insomniac Games', 200.00, 1, '12345678910', 'Vai teia !!!'),
('Doom Eternal', 'Bethesda', 130.00, 1, '12345678910', 'Nostalgia');

insert into pedido (quantidade, id_produto, cpf) values
(1, 1, '12345678910'),
(1, 2, '12345678910'),
(1, 3, '12345678910'),
(1, 4, '12345678910'),
(1, 5, '12345678910');

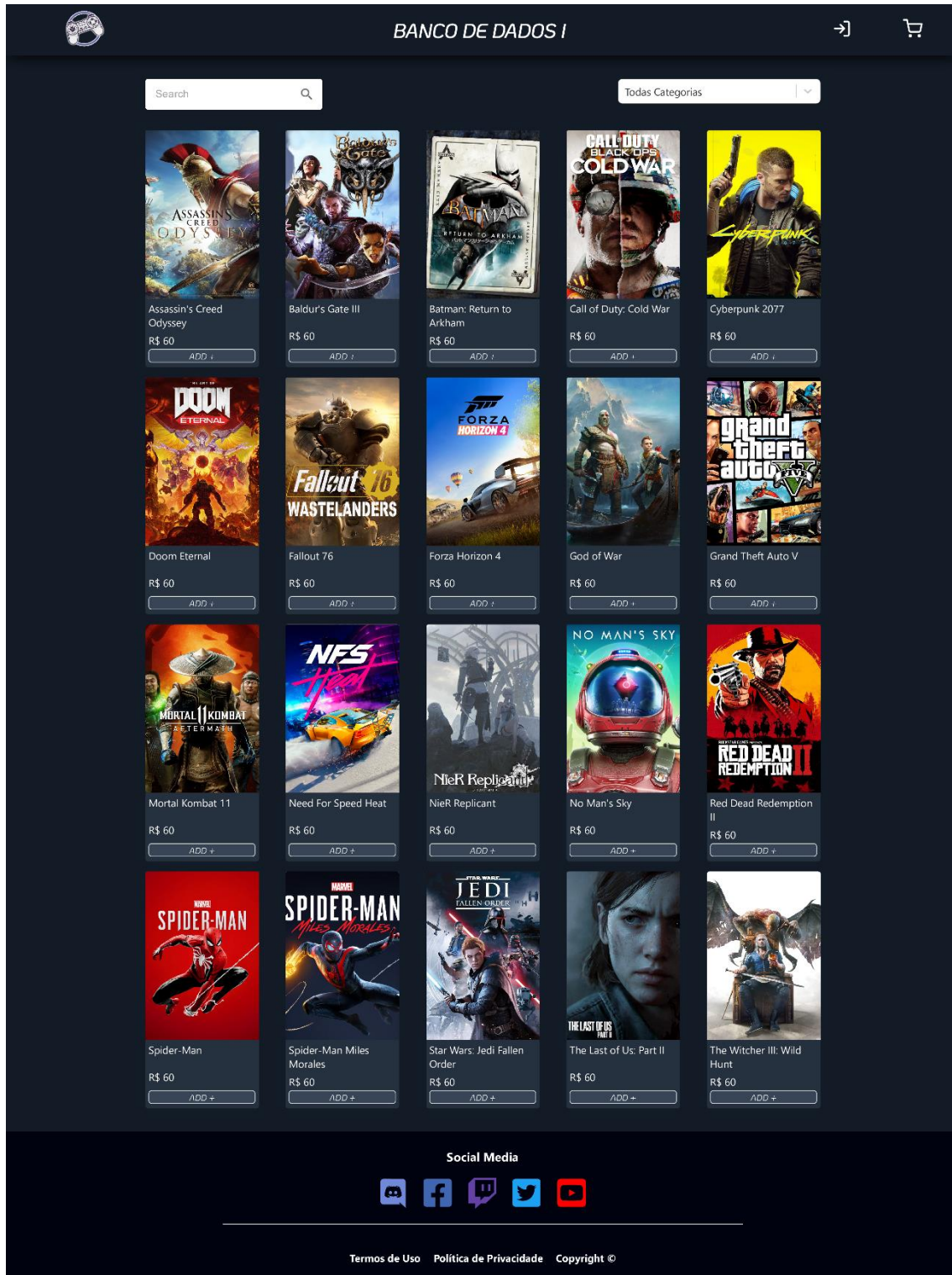
```

Esquema MYSQL da Base de Dados: Esquema MYSQL criado a partir da base de dados especificada acima.



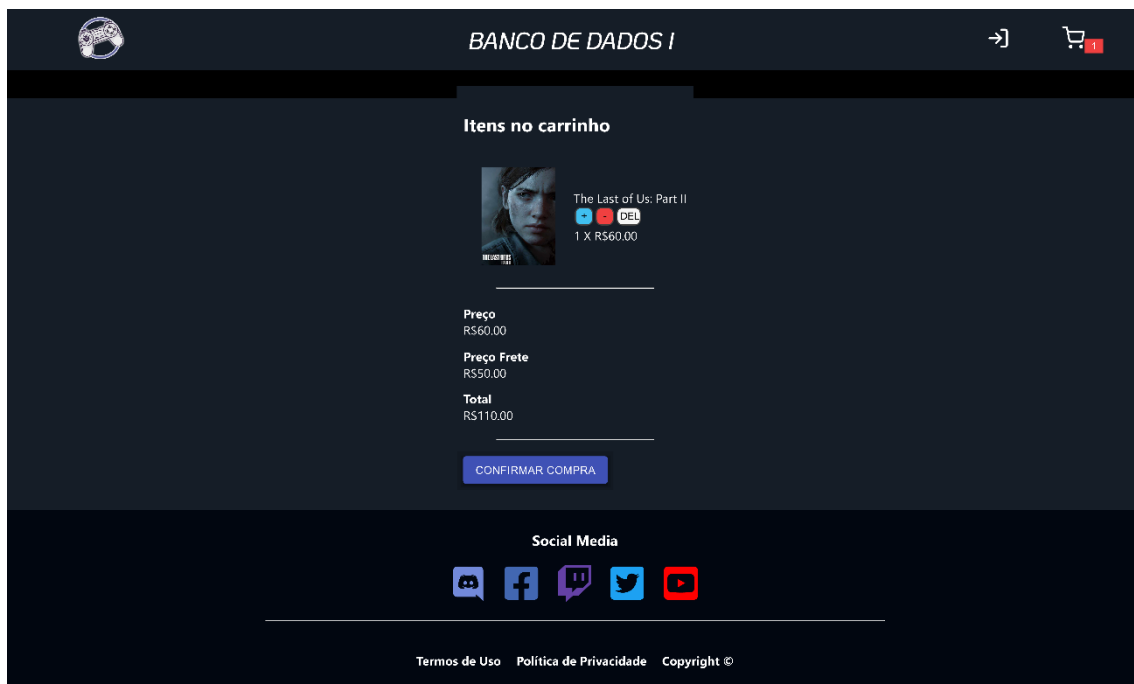
PROTÓTIPO (INTERFACES)

HOME



Apresentação dos produtos com sistema de filtragem e busca alfabética.

CARRINHO



Resumo dos produtos requeridos pelo usuário.

DADOS DO USUÁRIO

Dados do usuário para a confirmação da compra.

CONFIRMAÇÃO



Confirmação do pedido do usuário.

TERMOS DE USO



Termos de uso DPG Games.

POLÍTICA DE PRIVACIDADE



Políticas de privacidade DPG Games.

De forma geral, todas as páginas contêm:

- **HEADER:** Logo da empresa, título e os ícones de *login* e *carrinho*;
- **MAIN:** Conteúdo específico para cada página;
- **FOOTER:** Links para a social media, termos de uso e política de privacidade da nossa empresa.

PROJETO SEMESTRAL

BANCO DE DADOS 1