

Universidade de Brasília

Departamento de Ciência da Computação



Projeto de Banco de Dados
Marketplace

Autores:

Lucas de Azevedo Juvito 19/0047101

Mateus da Silva Souza 19/0035072

Brasília
27 de outubro de 2021

Conteúdo

1	Introdução	2
2	Diagrama Entidade Relacionamento	3
3	Modelo Relacional	4
4	Consultas em álgebra relacional	4
4.1	Primeira consulta	5
4.2	Segunda consulta	5
4.3	Terceira consulta	6
4.4	Quarta consulta	7
4.5	Quinta consulta	7
5	Avaliação das formas normais	8
5.1	Tabela de Unidades da Federação	8
5.2	Tabela de anúncios	9
5.3	Tabela de vendas	9
5.4	Tabela de comentários	9
5.5	Tabela de usuários	10
6	Diagrama de como é feita aligação entre interface gráfica e camada de persistência	11
7	Diagrama da camada de mapeamento	12
8	Github	12

1 Introdução

Pretendemos realizar um projeto de Banco de Dados implementando um marketplace. Na nossa estrutura, usuários (pessoas física ou jurídicas) poderiam realizar a criação de anúncios de seus produtos, e outros usuários realizarem a compra. Usuários também seriam capazes de comentar sobre os produtos recebidos, ou em que pretendem realizar a compra.

No banco de dados do Mercado Fechado existem usuários que podem ser pessoas físicas ou jurídicas, um usuário tem login e senha, além disso, se for uma pessoa física também tem CPF, nome e data de nascimento, mas, se for uma pessoa jurídica tem CNPJ e nome da empresa.

Ao interagir com a interface gráfica, pessoas logadas precisam de uma chave que prova que ela está logada, e que ela pode acessar páginas que somente pessoas logadas podem acessar.

Usuários podem ter endereços, um endereço tem CEP, unidade da federação, cidade, bairro, quadra, número e um complemento opcional.

O Mercado Fechado funciona a partir de anúncios nos quais tem título, descrição, produto, valor, comentários.

Um produto tem nome, foto, marca, fabricante, ano de fabricação, categoria e descrição interna.

Também é preciso efetivar uma venda de anúncio, salvando a data da venda, o comprador, o vendedor, o anúncio que foi vendido, o valor que foi vendido, o endereço de entrega e a avaliação do cliente.

Na interface gráfica temos um site que na tela inicial são listadas as categorias dos produtos e ao clicar numa destas categoria o usuário é redirecionado para uma tela com os anúncios dos produtos daquela categoria. Também temos nesta página o redirecionamento para o cadastro ou login de usuários.

Se o usuário clicar em um anúncio, ele irá para uma página que mostra as informações necessárias para comprar o anúncio junto com a foto do produto, comentários feitos naquele anúncio e avaliações de clientes que compraram aquele anúncio.

Se um usuário estiver logado, ele poderá criar anúncios; adicionar produtos; comprar anúncios; comentar em anúncios; e acessar seu perfil. No seu perfil, são mostradas suas compras e suas vendas, sendo também possível avaliar suas compras.

2 Diagrama Entidade Relacionamento

Para criar o Diagrama Entidade Relacionamento utilizamos a ferramenta brModelo.

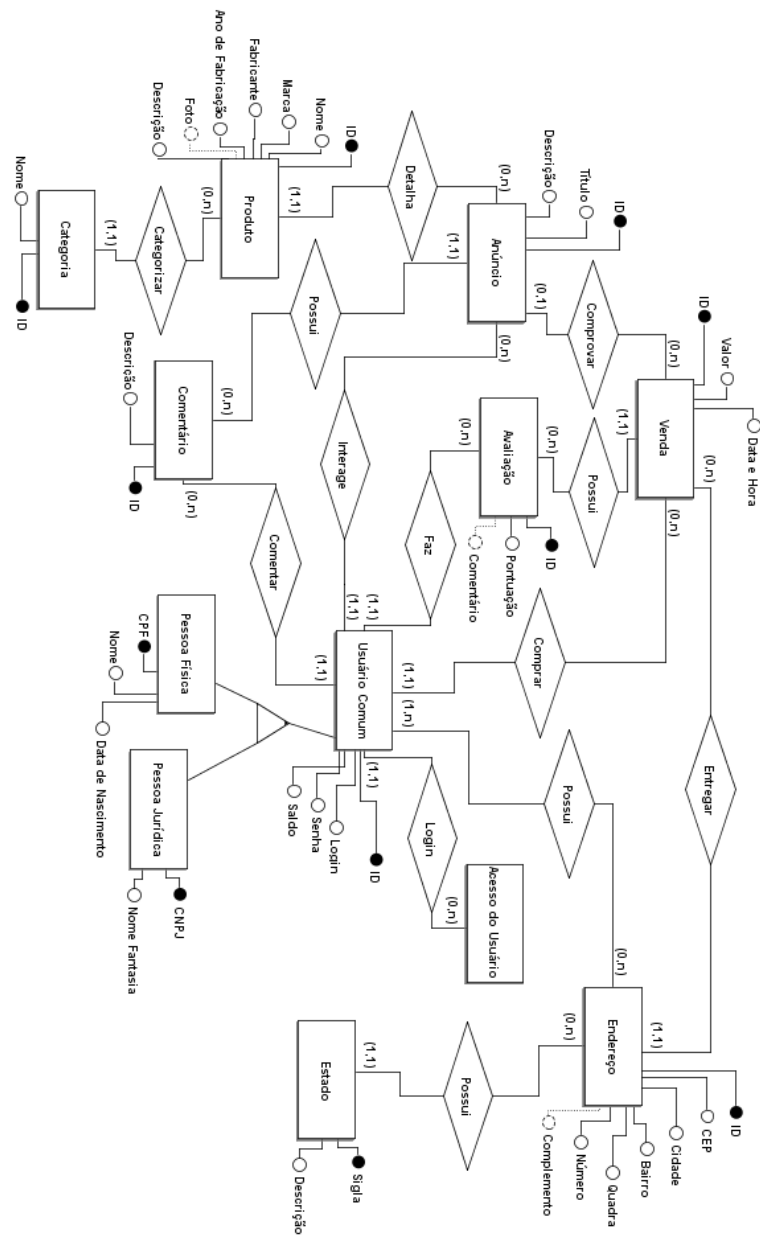
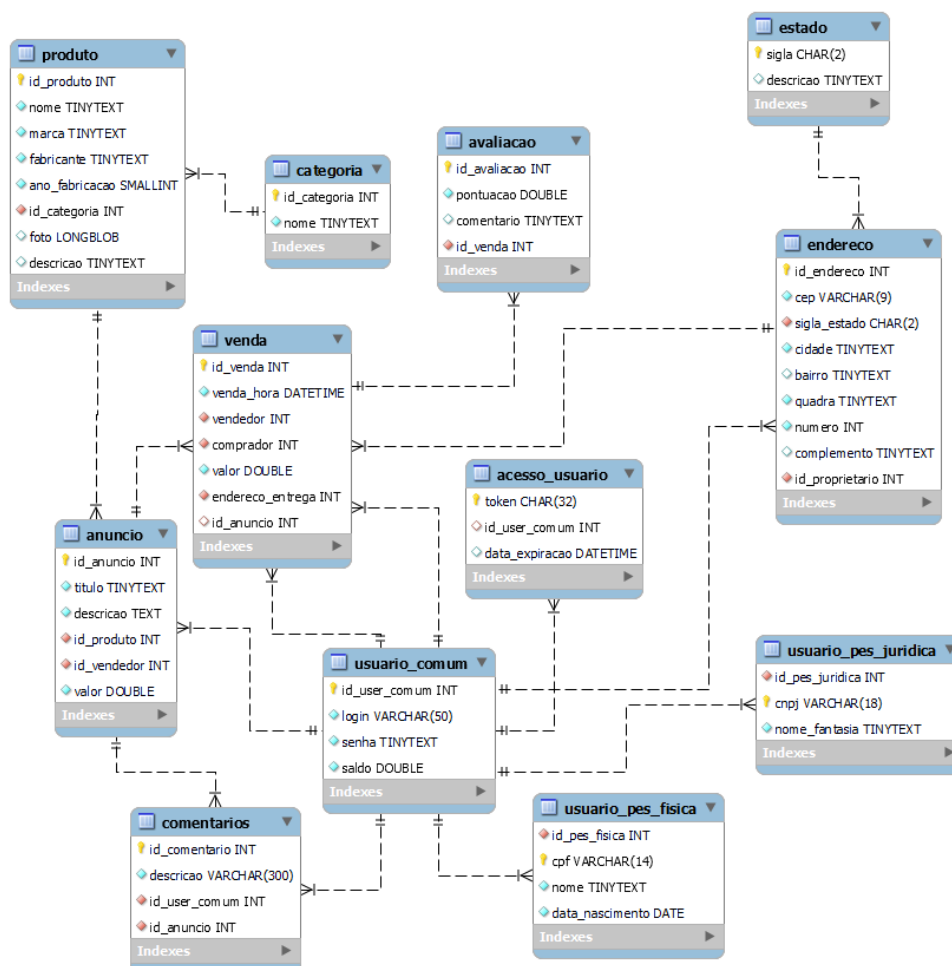


Figura 1: DER utilizando o brModelo

3 Modelo Relacional

Para criar o Modelo Relacional do banco de dados do projeto nós utilizamos a ferramenta de modelagem disponível no MySQL Workbench.



Modelo Relacional utilizando o MySQL Workbench

4 Consultas em álgebra relacional

Nessa seção faremos 5 consultas ao banco de dados usando álgebra relacional considerando o conjunto de dados que foram inseridos na sua criação, estes dados podem ser verificados no arquivo SQL disponibilizado no Github que será apresentado na seção 8.

4.1 Primeira consulta

Aqui nós vamos realizar uma consulta para visualizar as avaliações de vendas de anúncios feitas por usuários.

$$\begin{aligned}
 A_1 &\leftarrow avaliacao \bowtie_{avaliacao.id_venda=venda.id_venda} venda \\
 B_1 &\leftarrow A_1 \bowtie_{comprador=id_user_comum} usuario_comum \\
 C_1 &\leftarrow B_1 \bowtie_{venda.id_anuncio=anuncio.id_anuncio} anuncio \\
 &\pi_{login,pontuacao,titulo,comentario}(C_1)
 \end{aligned}$$

login	pontuacao	titulo	comentario
user2	3	PS5 muito barato	Caro
user1	3.5	Placa de vídeo para turbinar os seus video jogos	Deu bom
user1	4	Livro do LOL	É o esperado. LOL
user3	5	Fita antiga do Homem Aranha 2	Produto incrível
user4	0.5	IPHONE 13 PRO único no Brazil	Não gostei

Figura 2: Resultado

4.2 Segunda consulta

Nesta consulta queremos listar os endereços de um usuário específico (com ID 1), renomeando as colunas do resultado para melhorar a visualização.

$$\begin{aligned}
 A_2 &\leftarrow endereco \bowtie_{sigla=sigla_estado} estado \\
 B_2 &\leftarrow A_2 \bowtie_{id_proprietario=id_user_comum} usuario_comum \\
 C_2 &\leftarrow \sigma_{id_user_comum=1}(B_2) \\
 D_2 &\leftarrow \pi_{login,cep,descricao,cidade,bairro,quadra,numero}(C_2) \\
 \rho_{(Usuário,CEP,Unidade da Federação,Cidade,Bairro,Quadra,Número)}(D_2)
 \end{aligned}$$

Usuário	CEP	Unidade da Federação	Cidade	Bairro	Quadra	Número
user1	72910-000	Goiás	Águas Lindas de Goiás	Perola 1	quadra 9	900

Figura 3: Resultado

4.3 Terceira consulta

Para esta consulta precisamos de algumas extensões da álgebra relacional para poder ordenar (τ) e agrupar (γ) expressões. Nós vamos listar as categorias que mais têm produtos anunciados e ordenar isso em ordem decrescente, como no banco há somente um anúncio por categoria na sua criação o resultado não será muito expressivo, contudo ao adicionarmos mais anuncios comprovamos seu funcionamento.

$$\begin{aligned} A_3 &\leftarrow \sigma_{\text{anuncio.id_produto}=\text{produto.id_produto} \text{ AND } \text{produto.id_categoria}=\text{categoria.id_categoria}} \\ &\quad (\text{anuncio} \times \text{produto} \times \text{categoria}) \\ B_3 &\leftarrow \gamma_{\text{id_categoria}, \text{COUNT}(\text{id_anuncio})}(A_3) \\ C_3 &\leftarrow \pi_{\text{categoria.nome}, \text{COUNT}(\text{id_anuncio}) \rightarrow \text{qnt_anuncio}}(B_3) \\ &\quad \tau_{\text{qnt_anuncio} \downarrow}(C_3) \end{aligned}$$

nome	qnt_anuncio
CONSOLES	1
COMPONENTES PARA PC	1
LIVROS	1
FILMES	1
CELULARES	1

Figura 4: Resultado com o conjunto original de dados no banco

nome	qnt_anuncio
LIVROS	5
CONSOLES	3
CELULARES	2
COMPONENTES PARA PC	1
FILMES	1

Figura 5: Resultado após a inserção de alguns anúncios

4.4 Quarta consulta

Nesta consulta é listado os anúncios de uma empresa a partir do seu CNPJ que nesse caso é 16.780.211/0001-24

$$\begin{aligned}
A_4 &\leftarrow anuncio \bowtie_{anuncio.id_produto=produto.id_produto} produto \\
B_4 &\leftarrow A_4 \bowtie_{produto.id_categoria=categoria.id_categoria} categoria \\
C_4 &\leftarrow B_4 \bowtie_{id_vendedor=id_pes_juridica} usuario_pes_juridica \\
D_4 &\leftarrow \sigma_{cnpj="16.780.211/0001-24"}(C_4) \\
\pi_{nome_fantasia \rightarrow Empresa, titulo \rightarrow Anuncio, produto.nome \rightarrow Produto, categoria.nome \rightarrow Categoria}(D_4)
\end{aligned}$$

Empresa	Anuncio	Produto	Categoria
Locadora MaxFilmes	Fita antiga do Homem Aranha 2	Homem Aranha 2	FILMES

Figura 6: Resultado

4.5 Quinta consulta

Usaremos as operações para ordenar (τ) e agrupar (γ) expressões novamente. Dessa vez vamos listar os produtos que mais foram vendidos.

$$\begin{aligned}
A_5 &\leftarrow venda \bowtie_{venda.id_anuncio=anuncio.id_anuncio} anuncio \\
B_5 &\leftarrow A_5 \bowtie_{anuncio.id_produto=produto.id_produto} produto \\
C_5 &\leftarrow \gamma_{id_produto, COUNT(id_produto)}(B_5) \\
D_5 &\leftarrow \pi_{produto.nome, COUNT(id_produto) \rightarrow qnt_vendida}(C_5) \\
&\tau_{qnt_vendida} \downarrow (D_5)
\end{aligned}$$

nome	qnt_vendida
Playstation 5	1
RTX 3090	1
Reinos de Runeterra	1
Homem Aranha 2	1
Iphone 13 PRO	1

Figura 7: Resultado com o conjunto original de dados no banco

nome	qnt_vendida
Iphone 13 PRO	6
Playstation 5	4
RTX 3090	2
Reinos de Runeterra	1
Homem Aranha 2	1

Figura 8: Resultado após a inserção de algumas vendas

5 Avaliação das formas normais

Nesta seção iremos avaliar 3 formas normais em 5 tabelas do nosso banco de dados. A primeira forma normal (1FN) diz que o valor de uma coluna de uma tabela é indivisível. Uma tabela está na segunda forma normal (2FN) se ela está na (1FN) e todo atributo do complemento de uma chave candidata é totalmente funcionalmente dependente daquela chave. E uma tabela está na terceira forma normal (3FN) se ela também estiver na (2FN) e se, e somente se todos os atributos não-chave forem dependentes não-transitivos da chave primária.

5.1 Tabela de Unidades da Federação

Figura 9: estado

sigla	descricao
AC	Acre
AL	Alagoas
AM	Amazonas
AP	Amapá
BA	Bahia
CE	Ceará
DF	Distrito Federal
ES	Espírito Santo
GO	Goiás
MA	Maranhão
MG	Minas Gerais
MS	Mato Grosso do Sul
MT	Mato Grosso
PA	Pará

Como não há atributos multi-valorados, esta tabela se encontra na 1FN.

O atributo não-chave (descricao) é totalmente funcionalmente dependente da chave (sigla), e por também estar na 1FN, se encontra na 2FN.

Por último, temos que todos os atributos não-chave não definem outros atributos não-chave, e a tabela está na 1FN e 2FN. Logo, temos que a tabela está na 3FN.

5.2 Tabela de anúncios

Figura 10: anuncio

id_anuncio	titulo	descricao	id_produto	id_vendedor	valor
1	PS5 muito barato	O console mais potente da última geração	1	1	10000
2	Placa de vídeo para turbinar os seus video jogos	A melhor placa de vídeo já feita para os meros ...	2	2	32000
3	Livro do LOL	LOL LOL LOL	3	3	50
4	Fita antiga do Homem Aranha 2	Edição de colecionador rabiscada pelo diretor d...	4	4	70
5	IPHONE 13 PRO único no Brazil	Mais atual celular android que você terá em sua...	5	5	13000

Como não há atributos multi-valorados, esta tabela se encontra na 1FN.

Os atributo não-chave (titulo, descricao, id_produto, id_vendedor, valor) são totalmente funcionalmente dependentes da chave (id_anuncio), e devido a tabela também estar na 1FN, podemos dizer que esta se encontra na 2FN.

Por último, temos que todos os atributos não-chave não definem outros atributos não-chave, e a tabela está na 1FN e 2FN. Logo, temos que a tabela está na 3FN.

5.3 Tabela de vendas

Figura 11: venda

id_venda	venda_hora	vendedor	comprador	valor	endereço_entrega	id_anuncio
1	2021-10-23 17:08:02	1	2	10000	3	1
2	2021-10-23 17:08:03	2	1	32000	3	2
3	2021-10-23 17:08:03	3	1	50	3	3
4	2021-10-23 17:08:03	4	3	70	3	4
5	2021-10-23 17:08:03	5	4	13000	3	5

Como não há atributos multi-valorados, esta tabela se encontra na 1FN.

Os atributo não-chave (descricao) é totalmente funcionalmente dependente da chave (sigla), e por também estar na 1FN, se encontra na 2FN.

Por último, temos que todos os atributos não-chave não definem outros atributos não-chave, e está na 1FN e 2FN. Logo, temos que a tabela está na 3FN.

5.4 Tabela de comentários

Figura 12: comentario

id_comentario	descricao	id_user_comum	id_anuncio
1	Mas tá muito bar...	10	1
2	É seguro isso ai m...	9	1
3	Ainda estou inseg...	8	1
4	Não gostei, fui ta...	7	1
5	Produto é falseta...	6	1

Como não há atributos multi-valorados, esta tabela se encontra na 1FN.

Os atributo não-chave (descricao) é totalmente funcionalmente dependente da chave (sigla), e por também estar na 1FN, se encontra na 2FN.

Por último, temos que todos os atributos não-chave não definem outros atributos não-chave, e está na 1FN e 2FN. Logo, temos que a tabela está na 3FN.

5.5 Tabela de usuários

Figura 13: usuario_comum

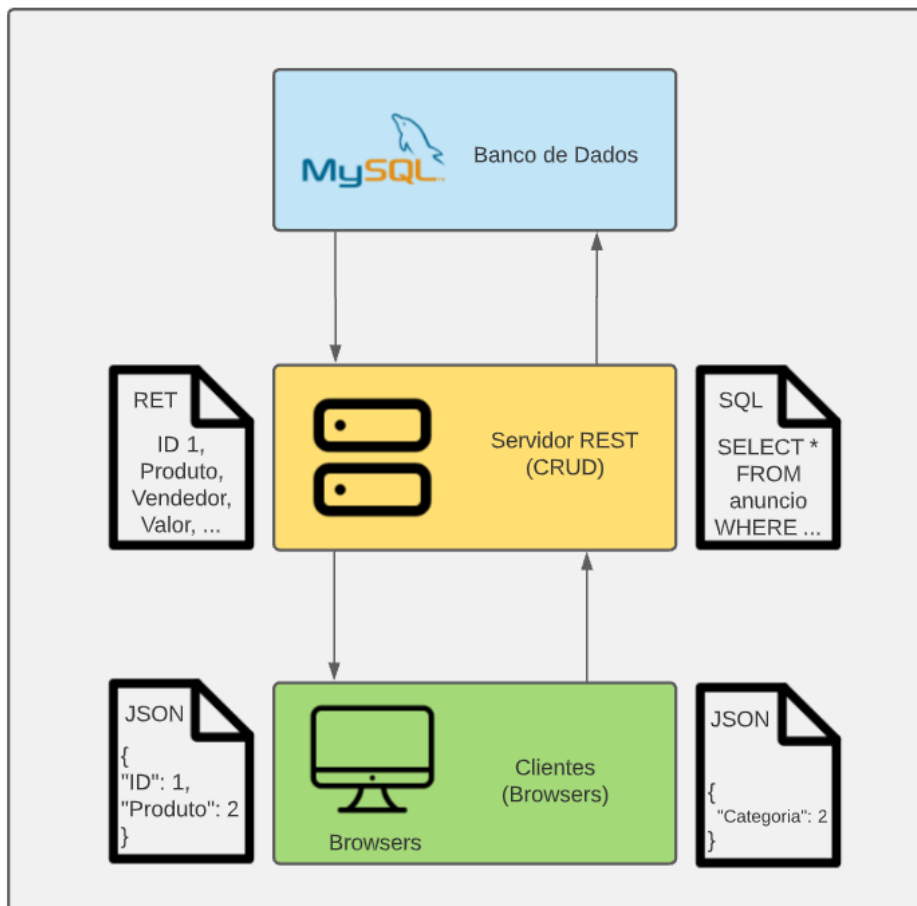
id_user_comum	login	senha
1	user1	123
2	user2	123
3	user3	123
4	user4	123
5	user5	123
6	user6	123
7	user7	123
8	user8	123
9	user9	123
10	user10	123

Como não há atributos multi-valorados, esta tabela se encontra na 1FN.

Os atributo não-chave (nome e data.nascimento) são totalmente funcionalmente dependentes da chave (cpf), e por também estar na 1FN, se encontra na 2FN.

Por último, temos que todos os atributos não-chave não definem outros atributos não-chave, e está na 1FN e 2FN. Logo, temos que a tabela está na 3FN.

6 Diagrama de como é feita a ligação entre interface gráfica e camada de persistência



Exemplo de pedido, com o cliente requisitando anúncios para uma determinada categoria, o servidor obtendo dados no BD, e os retornando.

7 Diagrama da camada de mapeamento

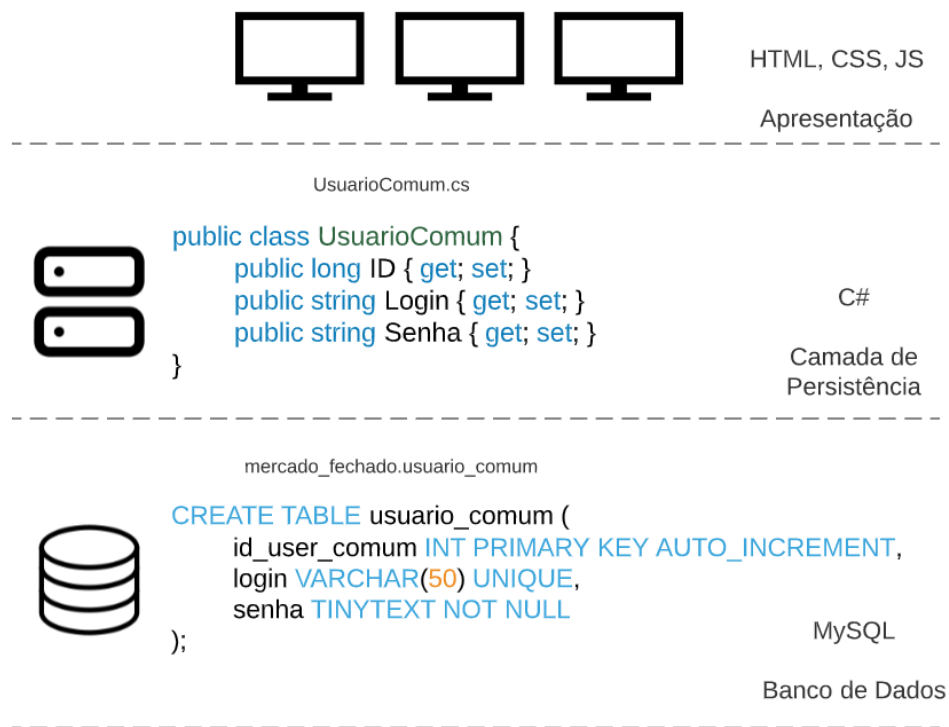


Figura 14: Resultado

8 Github

O desenvolvimento do script SQL que gerou o banco de dados e todos os códigos do projeto estão disponíveis no [GitHub](https://github.com/LucasJuvito/MercadoFechado) e pode ser acessado em <https://github.com/LucasJuvito/MercadoFechado>.

O SGBD utilizado para esse projeto foi o MYSQL, para a camada de persistência foi usado C# e para a interface gráfica um site básico feito em HTML, CSS e JavaScript. Para a comunicação entre servidor e cliente, é utilizado arquitetura REST.