

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**  
**ETEC. ZONA LESTE**  
**NOVOTEC DESENVOLVIMENTO DE SISTEMAS**

**BRUNO SANTOS PORTUGAL**

**LISTA 15: Projeto Final BD II – Parte III BD\_TodaBella**

**São Paulo**

**2023**

**Bruno Santos Portugal**

**LISTA 15: Projeto Final BD II – Parte III**  
**BD\_TodaBella**

Trabalho de encerramento do bimestre referente à disciplina de Banco de Dados, sob a orientação da professora Edna Pittner, abrangendo todos os conceitos e aprendizados adquiridos ao longo do ano.

**São Paulo**

**2023**

## Sumário

1.	LOJA TODA BELLA .....	4
1.1.	MER apresentado por Melissa: .....	5
1.2.	Estrutura das tabelas criada pela equipe de Melissa: .....	7
1.3.	Realização de inserções de dados nas tabelas do banco de dados. ....	10
1.4.	Select's .....	13
2.	PRogramação WEB (TOda bella).....	16
2.1.	Interface de apresentação.....	17
2.2.	Tela de Login .....	18
2.3.	Interface de gerenciamento.....	19
2.4.	Cadastro de informações .....	20
2.5.	Listagem de dados .....	21
2.6.	Consultas Especificas: .....	22

## 1. LOJA TODA BELLA

Há muitos anos, em uma movimentada rua da cidade, nasceu a "Toda Bella", uma pequena loja de beleza. Sua fundadora, Isabela, tinha uma paixão pelo mundo da beleza e desejava criar um espaço onde as pessoas pudessem encontrar produtos de alta qualidade para realçar sua beleza natural. O que começou como uma pequena boutique de beleza logo se transformou em um local de referência para os amantes da estética e cuidados pessoais.

Conforme o tempo passou, Toda Bella conquistou uma clientela fiel, graças ao compromisso com a qualidade, atendimento personalizado e uma seleção diversificada de produtos de beleza. No entanto, Isabela percebeu que, para manter a loja competitiva e continuar atendendo às necessidades dos clientes, era necessário modernizar a maneira como gerenciavam seus estoques, promoções e registros de clientes.

Foi aí que surgiu a ideia de criar um banco de dados personalizado para a Toda Bella. Isabela contratou uma equipe de desenvolvedores de software que criaram um sistema de banco de dados sob medida para a loja. Este sistema permitia que a

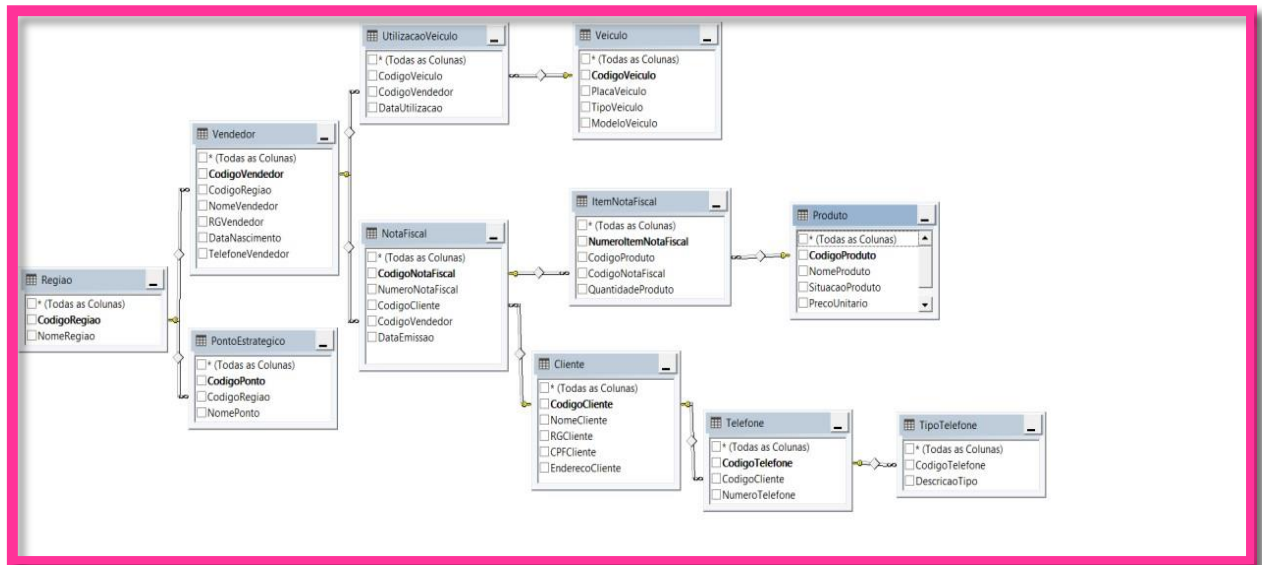
Toda Bella gerenciasse seu inventário de produtos, acompanhasse as preferências dos clientes e lançasse promoções direcionadas de forma eficiente.

Com o novo sistema de banco de dados, a Toda Bella viu um aumento significativo na eficiência operacional. Eles puderam otimizar seus estoques, garantindo que os produtos mais populares sempre estivessem disponíveis. Além disso, o sistema permitiu que eles criassem um programa de fidelidade, no qual os clientes recebiam ofertas personalizadas com base em seus históricos de compras.



### 1.1. MER apresentado por Melissa:

Melissa iniciou sua jornada na Toda Bella, realizando reuniões com Isabela e a equipe para entender as necessidades da loja. Com esse entendimento, ela elaborou o Modelo de Entidade-Relacionamento (MER), detalhando todos os aspectos do negócio, incluindo produtos, clientes, promoções e estoque.



### 1.2. Estrutura de tabelas SQLServe

dbo.Cliente
Colunas
CodigoCliente (PK, int, não nulo)
NomeCliente (varchar(50), nulo)
RGCliente (varchar(20), nulo)
CPFCliente (varchar(20), nulo)
EnderecoCliente (varchar(50), nulo)

dbo.Veiculo
Colunas
CodigoVeiculo (PK, int, não nulo)
PlacaVeiculo (varchar(7), nulo)
TipoVeiculo (varchar(50), nulo)
ModeloVeiculo (varchar(50), nulo)

dbo.UtilizacaoVeiculo
Colunas
CodigoVeiculo (FK, int, não nulo)
CodigoVendedor (FK, int, não nulo)
DataUtilizacao (date, nulo)

dbo.Regiao
Colunas
CodigoRegiao (PK, int, não nulo)
NomeRegiao (varchar(30), nulo)

dbo.ItemNotaFiscal

- Colunas
  - NumeroItemNotaFiscal (PK, int, não nulo)
  - CodigoProduto (FK, int, não nulo)
  - CodigoNotaFiscal (FK, int, não nulo)
  - QuantidadeProduto (int, não nulo)

dbo.PontoEstrategico

- Colunas
  - CodigoPonto (PK, int, não nulo)
  - CodigoRegiao (FK, int, não nulo)
  - NomePonto (varchar(100), nulo)

dbo.Produto

- Colunas
  - CodigoProduto (PK, int, não nulo)
  - NomeProduto (varchar(60), nulo)
  - SituacaoProduto (varchar(7), nulo)
  - PrecoUnitario (decimal(10,2), nulo)
  - QuantidadeEstoque (int, nulo)

dbo.Veiculo

- Colunas
  - CodigoVeiculo (PK, int, não nulo)
  - PlacaVeiculo (varchar(7), nulo)
  - TipoVeiculo (varchar(50), nulo)
  - ModeloVeiculo (varchar(50), nulo)

dbo.NotaFiscal

- Colunas
  - CodigoNotaFiscal (PK, int, não nulo)
  - NumeroNotaFiscal (int, não nulo)
  - CodigoCliente (FK, int, não nulo)
  - CodigoVendedor (FK, int, não nulo)
  - DataEmissao (date, nulo)

dbo.Vendedor

- Colunas
  - CodigoVendedor (PK, int, não nulo)
  - CodigoRegiao (FK, int, não nulo)
  - NomeVendedor (varchar(50), nulo)
  - RGVendedor (varchar(20), nulo)
  - DataNascimento (date, nulo)
  - TelefoneVendedor (varchar(15), nulo)

### 1.3. Estrutura das tabelas criada pela equipe de Melissa:

Logo após a elaboração do Modelo de Entidade-Relacionamento (MER), deu-se início à criação da estrutura do banco de dados. Melissa trabalhou em estreita colaboração com a equipe de desenvolvedores, garantindo que todas as informações essenciais fossem traduzidas de forma eficiente para o sistema.

Eles projetaram as tabelas do banco de dados, estabeleceram chaves primárias e chaves estrangeiras para garantir a integridade dos dados e a eficácia das consultas. Cada aspecto do negócio, desde os detalhes dos produtos, informações dos clientes, até os registros de promoções e estoque, encontrou seu lugar no banco de dados.

#### -- Tabela Produto:

```
CREATE TABLE Produto (
    CodigoProduto INT NOT NULL IDENTITY(1,1),
    NomeProduto VARCHAR(60),
    SituacaoProduto VARCHAR(7) CHECK (SituacaoProduto IN ('Ativo', 'Inativo')),
    PrecoUnitario DECIMAL(10, 2),
    QuantidadeEstoque INT,

    PRIMARY KEY (CodigoProduto) -- Chave primária da tabela.
);
```

#### -- Tabela Cliente:

```
CREATE TABLE Cliente (
    CodigoCliente INT NOT NULL IDENTITY(1,1),
    NomeCliente VARCHAR(50),
    RGCliente VARCHAR(20),
    CPFCliente VARCHAR(20),
    EnderecoCliente VARCHAR(50),

    PRIMARY KEY (CodigoCliente) -- Chave primária da tabela.
);
```

#### -- Tabela Telefone:

```
CREATE TABLE Telefone (
    CodigoTelefone INT NOT NULL IDENTITY(1,1),
    CodigoCliente INT NOT NULL,
    NumeroTelefone VARCHAR(15),

    PRIMARY KEY (CodigoTelefone), -- Chave primária da tabela.
    FOREIGN KEY (CodigoCliente) REFERENCES Cliente (CodigoCliente) -- Chave
    estrangeira para o cliente.
);
```

-- Tabela Tipo Telefone:

```
CREATE TABLE TipoTelefone (
    CodigoTelefone INT NOT NULL, -- Chave estrangeira para associar o tipo de
    telefone ao telefone.
    DescricaoTipo VARCHAR(20), -- Descrição do tipo de telefone.
    FOREIGN KEY (CodigoTelefone) REFERENCES Telefone (CodigoTelefone) -- Chave
    estrangeira para o telefone.);
```

-- Tabela Veículo:

```
CREATE TABLE Veiculo (
    CodigoVeiculo INT NOT NULL IDENTITY(1,1),
    PlacaVeiculo VARCHAR(7),
    TipoVeiculo VARCHAR(50),
    ModeloVeiculo VARCHAR(50),

    PRIMARY KEY (CodigoVeiculo) -- Chave primária da tabela.
);
```

-- Tabela Região:

```
CREATE TABLE Regiao (
    CodigoRegiao INT NOT NULL IDENTITY(1,1),
    NomeRegiao VARCHAR(30),

    PRIMARY KEY (CodigoRegiao) -- Chave primária da tabela.
);
```

-- Tabela Vendedor:

```
CREATE TABLE Vendedor (
    CodigoVendedor INT NOT NULL IDENTITY(1,1),
    NomeVendedor VARCHAR(50),
    RGVendedor VARCHAR(20),
    DataNascimento DATE,
    TelefoneVendedor VARCHAR(15),
    CodigoRegiao INT NOT NULL, -- Chave estrangeira para associar o vendedor a uma
    região.

    PRIMARY KEY (CodigoVendedor), -- Chave primária da tabela.
    FOREIGN KEY (CodigoRegiao) REFERENCES Regiao (CodigoRegiao) -- Chave estrangeira
    para a região.
);
```

-- Tabela Ponto Estratégico:

```
CREATE TABLE PontoEstrategico (
    CodigoPonto INT NOT NULL IDENTITY(1,1),
    NomePonto VARCHAR(50),
    CodigoRegiao INT NOT NULL, -- Chave estrangeira para associar o ponto a uma
    região.

    PRIMARY KEY (CodigoPonto), -- Chave primária da tabela.
    FOREIGN KEY (CodigoRegiao) REFERENCES Regiao (CodigoRegiao) -- Chave estrangeira
    para a região.
);
```



-- Tabela Utilização do Veículo:

```
CREATE TABLE UtilizacaoVeiculo (
    DataUtilizacao DATE,
    Codigoveiculo INT NOT NULL, -- Chave estrangeira para associar o veículo.
    Codigovendedor INT NOT NULL, -- Chave estrangeira para associar o vendedor.

    FOREIGN KEY (CodigoVeiculo) REFERENCES Veiculo (CodigoVeiculo), -- Chave
    estrangeira para o veículo.
    FOREIGN KEY (CodigoVendedor) REFERENCES Vendedor (CodigoVendedor) -- Chave
    estrangeira para o vendedor.
);
```

-- Tabela Nota Fiscal:

```
CREATE TABLE NotaFiscal (
    CodigoNotaFiscal INT NOT NULL IDENTITY(1,1),
    NumeroNotaFiscal INT NOT NULL,
    DataEmissao DATE,
    CodigoCliente INT NOT NULL, -- Chave estrangeira para associar o cliente à nota
    fiscal.
    CodigoVendedor INT NOT NULL, -- Chave estrangeira para associar o vendedor à nota
    fiscal.

    PRIMARY KEY (CodigoNotaFiscal), -- Chave primária da tabela.
    FOREIGN KEY (CodigoCliente) REFERENCES Cliente (CodigoCliente), -- Chave
    estrangeira para o cliente.
    FOREIGN KEY (CodigoVendedor) REFERENCES Vendedor (CodigoVendedor) -- Chave
    estrangeira para o vendedor.
);
```

-- Tabela Item da Nota Fiscal:

```
CREATE TABLE ItemNotaFiscal (
    NumeroItemNotaFiscal INT NOT NULL IDENTITY(1,1),
    QuantidadeProduto INT NOT NULL, -- Quantidade do produto na nota fiscal.
    CodigoProduto INT NOT NULL, -- Chave estrangeira para associar o produto ao item.
    CodigoNotaFiscal INT NOT NULL,

    PRIMARY KEY (NumeroItemNotaFiscal), -- Chave primária da tabela.
    FOREIGN KEY (CodigoNotaFiscal) REFERENCES NotaFiscal (CodigoNotaFiscal), -- Chave
    estrangeira para a nota fiscal.
    FOREIGN KEY (CodigoProduto) REFERENCES Produto (CodigoProduto) -- Chave
    estrangeira para o produto.
);
```

#### 1.4. Realização de inserções de dados nas tabelas do banco de dados:

Depois da estrutura do banco de dados estar pronta, o próximo passo foi preencher as tabelas com dados reais. Melissa e sua equipe trabalharam arduamente para realizar os "inserts" em cada tabela, garantindo que todas as informações estivessem corretamente registradas no sistema.

##### -- Inserção de dados na tabela Produto:

```
INSERT INTO Produto (NomeProduto, SituacaoProduto, PrecoUnitario, QuantidadeEstoque)
VALUES
('Sabonete' , 'Ativo' , 2.99 , 200),
('Creme Facial' , 'Ativo' , 15.50 , 5 ),
('Esmalte' , 'Ativo' , 5.99 , 150),
('Secador de Cabelo' , 'Ativo' , 49.99 , 30 ),
('Pincel de Maquiagem' , 'Ativo' , 7.99 , 100),
('Protetor Solar' , 'Inativo' , 12.99 , 80),
('Batom' , 'Ativo' , 9.50 , 60),
('Shampoo' , 'Inativo' , 8.99 , 90),
('Condicionador' , 'Ativo' , 8.99 , 80),
('Máscara de Cílios' , 'Inativo' , 6.50 , 70);
```

##### -- Inserção de dados na tabela Cliente:

```
INSERT INTO Cliente (NomeCliente, RGCliente, CPFCliente, EnderecoCliente)
VALUES
('Pedro Santos' , '7654321' , '123456789' , 'Rua C, 789'),
('Ana Pereira' , '9871234' , '234567890' , 'Av. D, 987'),
('José Silva' , '5432198' , '890123456' , 'Rua E, 456'),
('Marta Oliveira' , '3217890' , '567890123' , 'Av. F, 654'),
('Luiz Fernandes' , '8765432' , '678901234' , 'Rua G, 321');
```

##### -- Inserção de dados na tabela Telefone:

```
INSERT INTO Telefone (CodigoCliente, NumeroTelefone)
VALUES
(3, '555-2468'),
(4, '555-7890'), (5,
'555-9876'),
(1, '555-1111'),
(2, '555-2222');
```

##### -- Inserção de dados na tabela TipoTelefone:

```
INSERT INTO TipoTelefone (CodigoTelefone, DescricaoTipo)
VALUES
(3, 'Comercial' ),
(4, 'Celular Trabalho' ),
(5, 'Recado' ),
(1, 'Celular Pessoal' ),
(2, 'Residencial' );
```

```

-- Inserção de dados na tabela Veiculo:
INSERT INTO Veiculo (PlacaVeiculo, TipoVeiculo, ModeloVeiculo)
VALUES
('DEF456', 'Carro', 'SUV'      ), ('UVW123',
'Moto' , 'Custom'  ),
('GHI789', 'Carro', 'Hatchback'),
('JKL012', 'Moto' , 'Touring'  ),
('MNO345', 'Carro', 'Sedan'    );

-- Inserção de dados na tabela Regiao:
INSERT INTO Regiao (NomeRegiao)
VALUES
('Oeste'   ),
('Leste'   ),
('Centro'  ), ('Noroeste'),
('Sudoeste');

-- Inserção de dados na tabela Ponto Estratégico com os principais locais de cada
região (adicionais):
INSERT INTO PontoEstrategico (CodigoRegiao, NomePonto)
VALUES
(3, 'Avenida Paulista, Museu de Arte de São Paulo (MASP), Parque Trianon, Shopping
Paulista'),
(4, 'Parque Ibirapuera, Museu de Arte Moderna de São Paulo (MAM), Jardim Botânico'),
(5, 'Pátio do Colégio, Catedral da Sé, Mercado Municipal de São Paulo'),
(1, 'Parque Villa-Lobos, Shopping Villa-Lobos, Estádio Allianz Parque'),
(2, 'Parque Burle Marx, Praça Panamericana, Jockey Club de São Paulo'),
(3, 'Parque da Independência, Museu do Ipiranga, Sesc Ipiranga'),
(4, 'Parque Villa-Lobos, Tomie Ohtake Cultural Institute, Shopping Eldorado'),
(5, 'Bairro da Liberdade, Teatro Municipal de São Paulo, Pinacoteca do Estado'),
(1, 'Parque do Povo, Shopping Iguatemi, Cidade Jardim'),
(2, 'Parque do Povo, Parque Alfredo Volpi, Shopping Cidade Jardim'),
(5, 'Parque do Carmo, Shopping Aricanduva, Sesc Itaquera');

-- Inserção de dados na tabela Vendedor:
INSERT INTO Vendedor (CodigoRegiao, NomeVendedor, RGVendedor, DataNascimento,
TelefoneVendedor)
VALUES
(3, 'Rita Soares'   , '6543210', '1988-04-30', '555-3333'),
(4, 'Paulo Martins' , '4321098', '1992-07-12', '555-4444'), (5,
'Isabel Lima'      , '2109876', '1995-10-15', '555-5555'),
(1, 'Fernando Costa', '7890123', '1980-02-25', '555-6666'),
(2, 'Lúcia Santos'  , '0987654', '1987-11-05', '555-7777');

-- Inserção de dados na tabela UtilizacaoVeiculo:
INSERT INTO UtilizacaoVeiculo (CodigoVeiculo, CodigoVendedor, DataUtilizacao)
VALUES
(3, 3, '2023-10-27'), (4,
4, '2023-10-28'),
(5, 5, '2023-10-29'), (1,
1, '2023-10-30'),
(2, 2, '2023-10-31');

```

-- Inserção de dados na tabela Nota Fiscal:

```
INSERT INTO NotaFiscal (NumeroNotaFiscal, CodigoCliente, CodigoVendedor, DataEmissao)
VALUES
(1003, 3, 3, '2023-10-27'),
(1004, 4, 4, '2023-10-28'),
(1005, 5, 5, '2023-10-29'),
(1006, 1, 1, '2023-10-30'),
(1007, 2, 2, '2023-10-31'),
(1008, 2, 3, '2023-11-1'),
(1009, 1, 3, '2023-11-2');
```

-- Inserção de dados na tabela Item da Nota Fiscal:

```
INSERT INTO ItemNotaFiscal (CodigoProduto, CodigoNotaFiscal, QuantidadeProduto) VALUES
(3, 1003, 4),
(4, 1004, 1),
(5, 1005, 2),
(1, 1006, 5),
(2, 1007, 3),
(6, 1008, 2),
(8, 1009, 6);
```

## 1.5. Select's :

Com a conclusão da fase de preenchimento de dados, Melissa e sua equipe deram início a uma etapa crucial do desenvolvimento: a implementação de algoritmos de seleção específicos. Eles se debruçaram sobre as necessidades específicas do sistema, procurando aprimorar a eficiência na busca e recuperação de informações.

Para otimizar a performance, decidiram incorporar algoritmos de seleção avançados que pudessem lidar com grandes conjuntos de dados de maneira eficiente. Utilizando técnicas de indexação e estratégias de busca otimizadas, a equipe conseguiu reduzir significativamente o tempo necessário para recuperar informações cruciais do banco de dados.

```
-- Listar todos os pontos estratégicos de cada região.
SELECT Regiao.NomeRegiao, PontoEstrategico.NomePonto
FROM Regiao
INNER JOIN PontoEstrategico
ON Regiao.CodigoRegiao = PontoEstrategico.CodigoRegiao;
```

NomeRegiao	NomePonto
Centro	Avenida Paulista, Museu de Arte de São Paulo (MAS...
Noroeste	Parque Ibirapuera, Museu de Arte Moderna de São ...
Sudoeste	Pátio do Colégio, Catedral da Sé, Mercado Municipal...
Oeste	Parque Villa-Lobos, Shopping Villa-Lobos, Estádio Al...
Leste	Parque Burle Marx, Praça Panamericana, Jockey Clu...
Centro	Parque da Independência, Museu do Ipiranga, Sesc ...
Noroeste	Parque Villa-Lobos, Tomie Ohtake Cultural Institute, ...
Sudoeste	Bairro da Liberdade, Teatro Municipal de São Paulo,...
Oeste	Parque do Povo, Shopping Iguatemi, Cidade Jardim
Leste	Parque do Povo, Parque Alfredo Volpi, Shopping Cid...
Sudoeste	Parque do Carmo, Shopping Aricanduva, Sesc Itaque...

```
-- Listar os nomes das regiões cadastradas.
SELECT NomeRegiao FROM Regiao;
```

NomeRegiao
Oeste
Leste
Centro
Noroeste
Sudoeste

```
--Listar todos os vendedores e quais veículos que eles utilizaram no último mês.
SELECT Vendedor.NomeVendedor, Veiculo.PlacaVeiculo, UtilizacaoVeiculo.DataUtilizacao
FROM Vendedor
INNER JOIN UtilizacaoVeiculo
ON Vendedor.CodigoVendedor = UtilizacaoVeiculo.CodigoVeiculo
INNER JOIN Veiculo
ON UtilizacaoVeiculo.CodigoVeiculo = Veiculo.CodigoVeiculo
WHERE UtilizacaoVeiculo.DataUtilizacao <= DATEADD(MONTH, 0, GETDATE())
ORDER BY Vendedor.NomeVendedor;
```

NomeVendedor	PlacaVeiculo	DataUtilizacao
Fernando Costa	JKL012	2023-10-28
Isabel Lima	GHI789	2023-10-27
Lúcia Santos	MNO345	2023-10-29
Paulo Martins	UVW123	2023-10-31
Rita Soares	DEF456	2023-10-30

```
--Listar todos os vendedores responsáveis por cada região.
SELECT Regiao.NomeRegiao, Vendedor.NomeVendedor
FROM Regiao
INNER JOIN Vendedor
ON Regiao.CodigoRegiao = Vendedor.CodigoRegiao
ORDER BY Regiao.CodigoRegiao;
```

NomeRegiao	NomeVendedor
Oeste	Fernando Costa
Leste	Lúcia Santos
Centro	Rita Soares
Noroeste	Paulo Martins
Sudoeste	Isabel Lima

```
--Todos os produtos que ainda não foram vendidos.
SELECT NomeProduto
FROM Produto
WHERE CodigoProduto NOT IN (SELECT CodigoProduto FROM ItemNotaFiscal)
ORDER BY NomeProduto;
```

NomeProduto
Batom
Condicionador
Máscara de Cílios
Shampoo

```
--Todos os produtos vendidos por um determinado Vendedor
SELECT DISTINCT Produto.NomeProduto
FROM Produto
INNER JOIN ItemNotaFiscal
ON Produto.CodigoProduto = ItemNotaFiscal.CodigoProduto
INNER JOIN NotaFiscal
ON ItemNotaFiscal.CodigoNotaFiscal = NotaFiscal.CodigoNotaFiscal
INNER JOIN Vendedor ON NotaFiscal.CodigoVendedor = Vendedor.CodigoVendedor
WHERE Vendedor.NomeVendedor = 'Rita Soares';
```

--O comando Select Distinct é usado para apenas retornar valores diferentes.  
 --Dentro de uma tabela, uma coluna pode tem registros iguais, e algumas vezes  
 --só queremos valores únicos.

NomeProduto
Sabonete

```
-- Todos os vendedores que venderam um determinado <produto>.
SELECT DISTINCT Vendedor.NomeVendedor
FROM Vendedor
INNER JOIN NotaFiscal
ON Vendedor.CodigoVendedor = NotaFiscal.CodigoVendedor
INNER JOIN ItemNotaFiscal
ON NotaFiscal.CodigoNotaFiscal = ItemNotaFiscal.CodigoNotaFiscal
INNER JOIN Produto
ON ItemNotaFiscal.CodigoProduto = Produto.CodigoProduto
WHERE Produto.NomeProduto = 'Sabonete';
```

NomeVendedor
Rita Soares

```
-- Listar o histórico de utilização de um determinado <veículo>.
SELECT Veiculo.PlacaVeiculo, Veiculo.ModeloVeiculo, Vendedor.NomeVendedor,
UtilizacaoVeiculo.DataUtilizacao
FROM Vendedor
INNER JOIN UtilizacaoVeiculo
ON Vendedor.CodigoVendedor = UtilizacaoVeiculo.CodigoVendedor
INNER JOIN Veiculo
ON UtilizacaoVeiculo.CodigoVeiculo = Veiculo.CodigoVeiculo
WHERE Veiculo.PlacaVeiculo = 'JKL012';
```

PlacaVeiculo	ModeloVeiculo	NomeVendedor	DataUtilizacao
JKL012	Touring	Fernando Costa	2023-10-28

```
--Listar a quantidade de itens de cada nota fiscal:
SELECT CodigoNotaFiscal, COUNT(*) AS QuantidadeItens
FROM ItemNotaFiscal
GROUP BY CodigoNotaFiscal;
```

CodigoNotaFiscal	QuantidadeItens
1	1
2	1
3	1
4	1
5	1
6	1

## **2. PROGRAMAÇÃO WEB (TODA BELLA)**

Empolgada com os resultados alcançados até o momento, Melissa reuniu sua equipe para discutir uma surpresa especial para Isadora, a cliente que havia sido uma parceira essencial desde o início do projeto. Queriam não apenas entregar um banco de dados robusto, mas também proporcionar uma experiência mais acessível e interativa para Isadora e sua equipe.

Foi decidido, então, criar uma aplicação web personalizada, intuitiva e amigável, que permitiria a Isadora acessar e gerenciar os dados de forma mais fácil e eficiente. A aplicação seria uma extensão do sistema de banco de dados, oferecendo funcionalidades adicionais e uma interface aprimorada.

Melissa e sua equipe mergulharam de cabeça no desenvolvimento da aplicação web, utilizando tecnologias modernas para garantir uma experiência de usuário fluida e responsiva.

Durante o processo de desenvolvimento, Melissa manteve uma comunicação próxima com Isadora, compartilhando os avanços e ajustando a aplicação de acordo com as preferências e feedback da cliente. A colaboração estreita fortaleceu ainda mais o relacionamento entre as duas equipes e garantiu que a aplicação atendesse plenamente às expectativas de Isadora.

Finalmente, o dia da apresentação chegou. Melissa e sua equipe organizaram uma reunião virtual para mostrar a Isadora a nova aplicação web. Compartilharam a tela e guiaram Isadora por meio das diversas funcionalidades, destacando como a aplicação simplificaria as tarefas diárias, proporcionando uma visão mais clara e acessível dos dados.

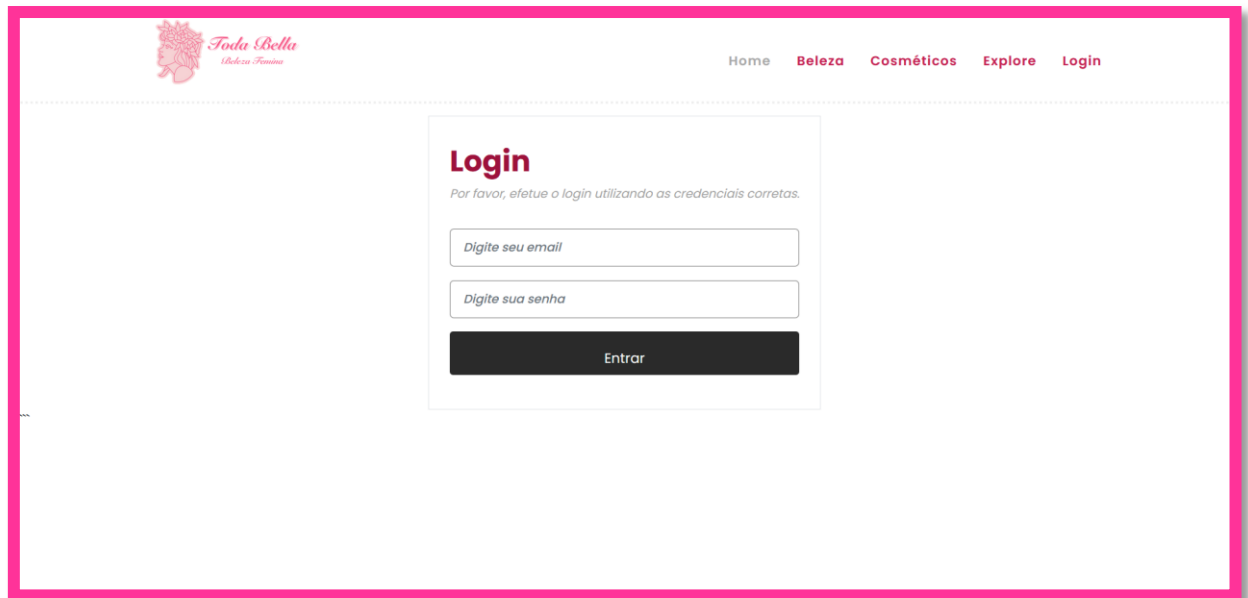


## 2.1. Interface de apresentação:



Optamos por desenvolver uma tela com cores simples, visando aprimorar a experiência do usuário da melhor maneira possível. Para isso, a equipe dedicou-se não apenas à escolha cuidadosa das cores, mas também à implementação de elementos visuais intuitivos e funcionais. Cada detalhe foi meticulosamente pensado para garantir que a navegação fosse fluida e agradável, proporcionando ao usuário não apenas informações claras, mas também uma estética visualmente atraente. Essa abordagem minuciosa não apenas elevou a qualidade estética da interface, mas também contribuiu para a usabilidade, criando uma experiência integrada e agradável para todos os usuários.

## 2.2. Tela de Login:



The screenshot shows a web application interface for 'Toda Bella'. At the top left is a logo with a pink flower and the text 'Toda Bella' and 'Beleza Feminina'. To the right is a navigation menu with links: 'Home', 'Beleza', 'Cosméticos', 'Explore', and 'Login'. The 'Login' link is highlighted in pink. Below the navigation bar is a central login form. The form has a title 'Login' in bold pink text, followed by a subtitle 'Por favor, efetue o login utilizando as credenciais corretas.' in a smaller font. There are two input fields: the first is labeled 'Digite seu email' and the second is labeled 'Digite sua senha'. Below these fields is a dark grey button with the text 'Entrar' in white.

Até o momento presente, limitamos a funcionalidade do site exclusivamente ao gerenciamento de banco de dados. Contudo, vislumbramos a possibilidade de expansão em breve, transformando-o em uma plataforma completa que não apenas gerencia dados, mas também possibilita a realização de compras online. Esta perspectiva representa um passo adiante no desenvolvimento do site, ampliando sua utilidade para além do âmbito do gerenciamento de informações, abrindo as portas para uma experiência mais abrangente e interativa para os usuários. Estamos animados com as oportunidades que essa potencial implementação trará, oferecendo uma solução mais abrangente e integrada aos nossos usuários.

E possível acessar o banco de dados adicionando as seguintes credencias:

**Email:** admin@example.com;

**Senha:** senha123.

### 2.3. Interface de gerenciamento :



Ao inserir as credenciais corretas, temos acesso a uma interface que possibilita a manipulação de diversas funções do banco de dados. Nesse espaço, é possível realizar ações como cadastrar novos itens, visualizar listas e efetuar consultas específicas, proporcionando um controle mais abrangente sobre as informações armazenadas. Essa tela se revela como uma ferramenta central para a administração do banco de dados, oferecendo a capacidade de realizar operações diversas de forma eficiente e personalizada.

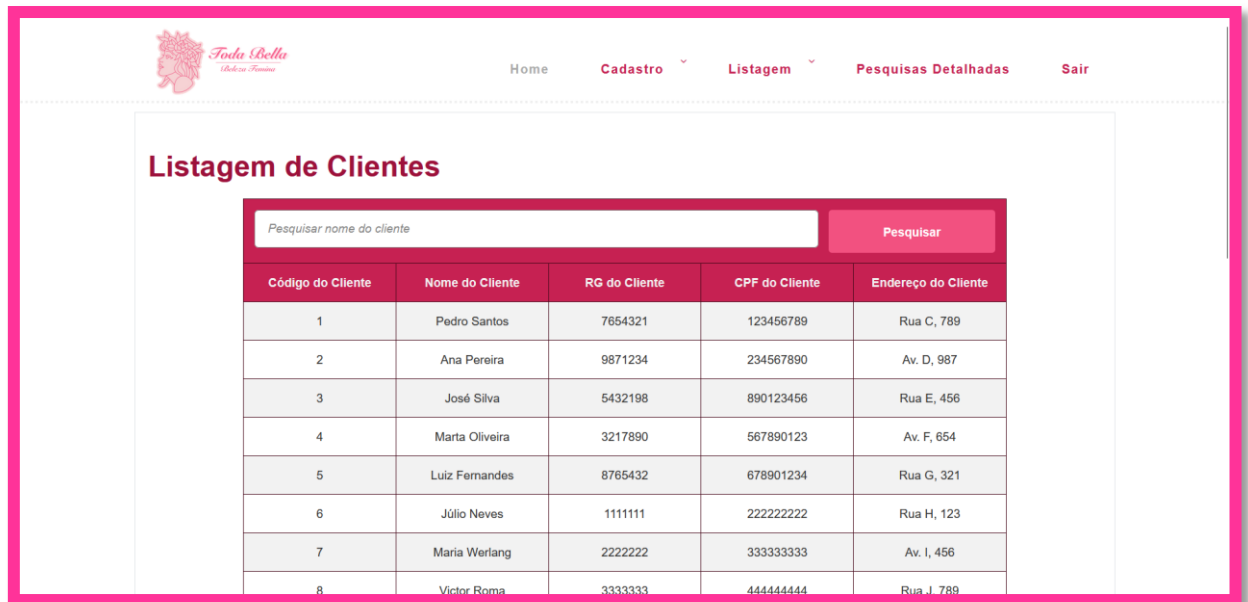
## 2.4. Cadastro de informações :



The image shows a web application interface with a pink header. On the left is a logo for 'Toda Bella' with the tagline 'Bela e Simples'. On the right are navigation links: 'Home', 'Cadastro' (with a dropdown arrow), 'Listagem' (with a dropdown arrow), 'Pesquisas Detalhadas', and 'Sair'. The main content area features a form titled 'Cadastro de Cliente' in bold pink text. The form contains five input fields, each preceded by a label: 'Nome do Cliente:', 'RG do Cliente:', 'CPF do Cliente:', 'CEP do Cliente:', and 'Endereço do Cliente:'. The entire form is enclosed in a light gray border.

Este é um exemplo representativo de registro que podemos incluir em nossa base de dados. Trata-se de uma entrada que ilustra as informações essenciais a serem armazenadas, e estamos abertos a personalizar e adicionar campos adicionais conforme necessário. Este cadastro serve como um modelo inicial, mas podemos aprimorá-lo considerando as particularidades e requisitos específicos do sistema, garantindo assim uma abordagem mais abrangente e adaptada às necessidades da nossa aplicação.

## 2.5. Listagem de dados:



**Listagem de Clientes**

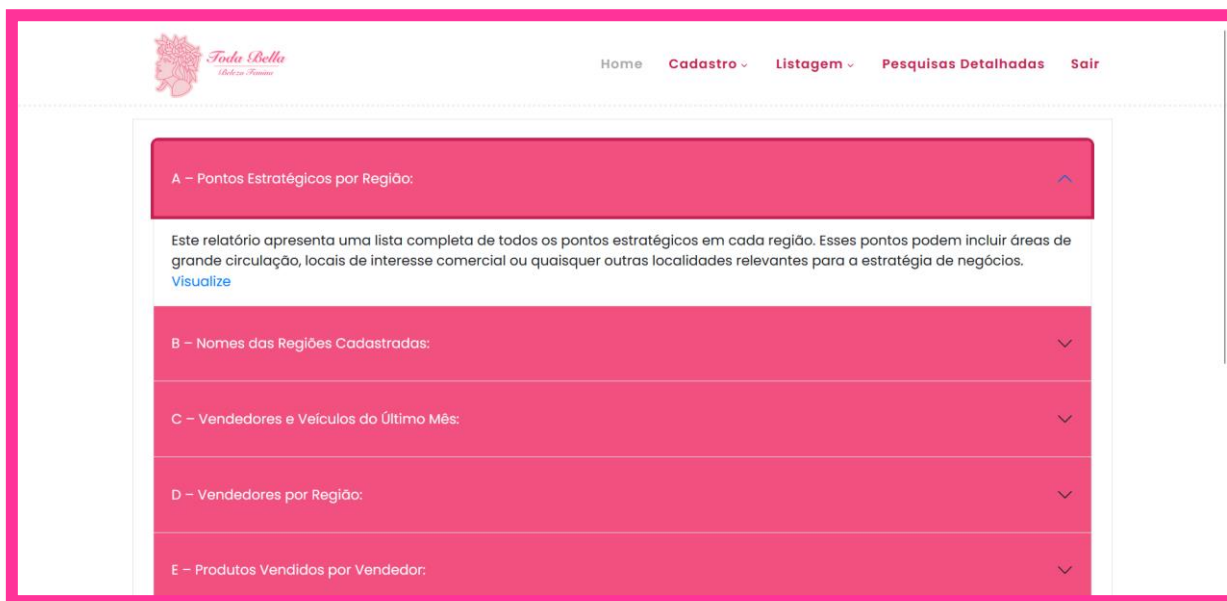
Pesquisar nome do cliente Pesquisar

Código do Cliente	Nome do Cliente	RG do Cliente	CPF do Cliente	Endereço do Cliente
1	Pedro Santos	7654321	123456789	Rua C, 789
2	Ana Pereira	9871234	234567890	Av. D, 987
3	José Silva	5432198	890123456	Rua E, 456
4	Marta Oliveira	3217890	567890123	Av. F, 654
5	Luiz Fernandes	8765432	678901234	Rua G, 321
6	Júlio Neves	1111111	222222222	Rua H, 123
7	Maria Werlang	2222222	333333333	Av. I, 456
8	Víctor Roma	3333333	444444444	Rua J, 789

Neste ponto, é possível visualizar a lista de clientes, permitindo a pesquisa por meio do nome do cliente para uma consulta mais direcionada. Essa funcionalidade oferece uma maneira eficiente de localizar e visualizar informações específicas relacionadas aos clientes cadastrados. Ao digitar o nome desejado, o sistema proporciona uma resposta precisa, facilitando a identificação e visualização dos detalhes pertinentes a cada cliente. Essa capacidade de pesquisa personalizada representa uma ferramenta valiosa para explorar e analisar a relação de clientes em nosso banco de dados.

## 2.6. Consultas Específicas:

Aqui podemos visualizar algumas consultas pré-definidas.



Ao clicar na opção "Visualizar", é possível acessar detalhadamente os resultados da pesquisa que foi predefinida. Essa funcionalidade permite uma análise mais aprofundada das informações relacionadas à pesquisa específica realizada. Os dados apresentados de maneira detalhada oferecem uma visão abrangente e organizada, permitindo aos usuários explorar com facilidade os resultados da pesquisa predefinida. Essa abordagem torna o processo de visualização mais intuitivo e eficaz, proporcionando uma experiência mais rica e informativa para os usuários que desejam examinar detalhes específicos dos resultados encontrados.

Região	Ponto Estratégico
Oeste	Parque do Povo, Shopping Iguatemi, Cidade Jardim
Oeste	Parque do Ibirapuera
Oeste	Aeroporto de Congonhas
Oeste	Hospital Santa Paula
Oeste	Museu do Ipiranga
Oeste	Praça das Crianças
Oeste	Estação Santo Amaro
Oeste	Hospital Albert Einstein
Oeste	Praça Charles Miller

Assim concluímos mais um projeto com êxito. Estou emocionado em compartilhar que o site, resultado desse esforço conjunto, estará disponível na forma de um arquivo anexado à tarefa. Gostaria de expressar meu profundo agradecimento à professora Edna Pittner por não apenas transmitir conhecimentos técnicos, mas também por ser uma fonte constante de inspiração. Foi graças à sua paixão pelo universo de bancos de dados e programação web que descobri meu próprio entusiasmo por essas áreas.

Agradeço sinceramente pelos ensinamentos valiosos que a senhora proporcionou não só a mim, mas a todos os meus colegas de classe. Seu comprometimento e habilidades pedagógicas tornaram o aprendizado uma jornada envolvente e enriquecedora para todos nós. Sua maestria em ensinar vai além do simples repasse de informações; ela inspira o desejo de aprender e explorar mais profundamente.

Este projeto marca não apenas o encerramento de uma etapa, mas também o início de uma jornada contínua em busca de conhecimento e aprimoramento. Estou grato por ter tido a oportunidade de ser orientado por uma professora tão dedicada e competente como a senhora.

Atenciosamente,

Bruno Portugal