



CENTRO UNIVERSITÁRIO AUGUSTO MOTTA

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Campo Grande - Noturno

Projeto para o módulo de BackEnd

Tema:

Adriel Furtado, Faid A., Júlio César Correia, Lucas Bittencourt, Lucas Tadashi e
Gabryel Fellipe

https://github.com/d3kinho/Projeto_Toca_das_bebidas

Rio de Janeiro, RJ
Setembro/ 2025

Sumário

1. Introdução.....	2
2. Levantamento de Requisitos.....	2
3. Plano de Implementação.....	3
4. Teste Realizados.....	6
5. Considerações.....	6
6. Anexos.....	6

1. Introdução

Este projeto tem por objetivo implementar uma aplicação WEB integrando o FRONTEND com o BACKEND, fazendo uso de banco de dados juntamente com o uso do PHP. Um dos obstáculos encontrados, até o momento, foi a necessidade do uso de um servidor, que foi remediada com o uso do XAMPP, que possibilita um servidor local, que é o necessário para o desenvolver e testar aplicações.

1.1. Objetivo

O objetivo deste projeto trata-se de uma loja virtual de bebidas , onde um usuário pode se cadastrar, ver a descrição de um produto e por fim comprar, visando facilitar e modernizar a compra dos produtos da loja.

2. Levantamento de Requisitos

Requisitos Funcionais (RF)

RF-01. Autenticação de Usuário: A aplicação deve permitir o cadastro de novos usuários e o login de usuários existentes. Usuários comuns podem se cadastrar pelo sistema, enquanto usuários "master" são criados diretamente no banco de dados, via script SQL.

RF-02. Cadastro de Usuário: O formulário de cadastro deve conter campos para nome completo, data de nascimento, sexo, nome materno, CPF, e-mail, telefones (celular e fixo), endereço completo, login, senha e confirmação de senha. Todos os campos, exceto o login e a confirmação de senha, são de preenchimento obrigatório.

- O campo de login deve ter exatamente 6 caracteres alfabéticos.
- A senha deve ser criptografada e ter 8 caracteres alfabéticos.
- O campo CPF deve ter conferência do Dígito Verificador.
- O CEP deve ser validado e, se possível, preencher os demais campos de endereço automaticamente via API.
- O sistema deve redirecionar o usuário para a tela de login após um cadastro bem-sucedido.

RF-02. Login: A tela deve ter campos para login e senha. Após a validação, o sistema deve acionar a tela de segundo fator de autenticação (2FA).

RF-04. Segundo Fator de Autenticação (2FA): Todos os usuários, tanto "master" quanto "comum", terão o segundo fator de autenticação. O sistema deve fazer uma pergunta de segurança aleatória (nome da mãe, data de nascimento

ou CEP) e verificar a resposta com os dados no banco de dados. Após 3 tentativas sem sucesso, o usuário deve ser redirecionado para a tela de login.

RF-05. Gestão de Usuários (Acesso Master):

- Consulta de Usuários: O usuário "master" deve poder visualizar uma lista de todos os usuários.
- Busca: Deve haver um campo de pesquisa onde o usuário "master" pode buscar usuários comuns por uma "substring" (parte do nome).
- Exclusão de Usuários: A tela de consulta deve apresentar um botão de exclusão para cada usuário comum, permitindo que o usuário "master" remova usuários após a confirmação.

RF-06. Alteração de Senha: Somente o usuário comum pode alterar a sua própria senha.

RF-07. Visualização de Logs: O usuário "master" deve poder consultar um log de autenticação, com um filtro de busca por nome ou CPF. O log deve mostrar o dia e a hora, nome do usuário e o 2FA exibido, em ordem cronológica inversa (mais recentes primeiro).

RF-08. Telas de Erro: O sistema deve exibir uma tela de erro quando algo inesperado acontecer, como uma falha na autenticação.

RF-09. Exibição de Informações:

- **Tela principal:** A tela principal deve ter um menu e exibir informações dos produtos, com suas respectivas descrições.
- **Modelo do BD:** A tela do modelo do banco de dados deve ter uma imagem da modelagem utilizada no sistema.

RF-010. Funcionalidades de Acessibilidade

- PDF: Inserir um botão para baixar a lista de usuários em formato PDF.
- Contraste: A aplicação deve prover uma barra com a opção de troca de contraste (fundo escuro/fonte clara e vice-versa).
- Tamanho da Fonte: A aplicação deve permitir o aumento e a diminuição do tamanho das fontes.

Requisitos não funcionais (RNF)

RNF-01. Identidade Visual e Design:

- O site deve usar uma identidade visual clara.
- Após o login, o login do usuário deve ser exibido no canto superior direito de todas as telas.
- O sistema deve exibir um menu e uma opção de logout em todas as telas após a autenticação.
- Mensagens de feedback ao usuário, como em formulários, devem usar modais ou toasts para uma estética elegante, evitando alerts. A identidade visual deve ser consistente em todos os feedbacks.

RNF-02. Tecnologia:

- A linguagem de programação para o Back-End deve ser PHP.
- O banco de dados a ser utilizado é o MySQL.
- O sistema deve ser responsivo e pode usar frameworks de front-end como Materialize, Bootstrap ou outros.

RNF-02. Segurança: A senha do usuário deve ser criptografada.

RNF-04. Arquitetura: O site deve ter um conjunto de páginas interconectadas.

3. Desenvolvimento

Sprint 1: Estrutura e Back-End Essencial

Objetivo: Configurar o ambiente e desenvolver a base da aplicação, incluindo o sistema de autenticação.

Backlog:

- **Configuração do Ambiente:** Instalar e configurar o ambiente de desenvolvimento (XAMPP com PHP e MySQL).
- **Modelagem e Banco de Dados:** Criar o script SQL para as tabelas de usuário, logs, e as tabelas necessárias, seguindo o modelo ER.
- **Lógica de Cadastro e Login:** Implementar a lógica de back-end para o cadastro de usuários. Isso inclui a validação dos campos, como CPF e CEP com API , e a criptografia da senha.
- **Lógica de Login:** Implementar a funcionalidade que valida o login e a senha no banco de dados.
- **Usuário Master:** Inserir o usuário master via script SQL no banco de dados.

Sprint 2: Autenticação e Funções Básicas

Objetivo: Configurar o ambiente e desenvolver a base da aplicação, incluindo o sistema de autenticação.

Backlog:

- **Implementação do 2FA:** Criar a lógica para a tela de Segundo Fator de Autenticação (2FA). A geração da pergunta deve ser aleatória e a resposta deve ser validada com o banco de dados. O sistema deve redirecionar para a tela de login após 3 tentativas sem sucesso.
- **Implementação do 2FA:** Desenvolver a funcionalidade de logout.
- **Lógica de Alteração de Senha:** Criar o sistema que permite ao usuário comum alterar a própria senha.

Sprint 3: Funcionalidades do Perfil Master

Objetivo: Desenvolver todas as funcionalidades exclusivas do perfil master.

Backlog:

- **Controle de Acesso:** Implementar o gerenciamento de sessão de usuário para controlar o acesso de acordo com o perfil.
- **Consulta de Usuários:** Criar a lógica de back-end para a tela de Consulta de Usuário, que exibe a lista de todos os usuários.
- **Busca:** Desenvolver a funcionalidade de pesquisa que permite ao usuário master buscar por uma substring no nome dos usuários comuns.
- **Exclusão de Usuários:** Implementar a lógica de exclusão, incluindo a confirmação para remover o usuário comum.
- **Tela de Logs:** Criar a lógica para a página de logs de autenticação, permitindo a consulta por nome, CPF ou todos, exibindo os dados em ordem mais recente.

Sprint 4: Acessibilidade, Testes e Entrega

Objetivo: Desenvolver todas as funcionalidades exclusivas do perfil master.

Backlog:

- Implementar a funcionalidade para baixar a lista de usuários em PDF
- Desenvolver o mecanismo de troca de contraste e de aumento/diminuição do tamanho das fontes.
- **Testes e Validação:** Realizar testes de ponta a ponta em todas as funcionalidades, garantindo que o sistema é responsivo e que não há erros.

Documentação e Entrega:

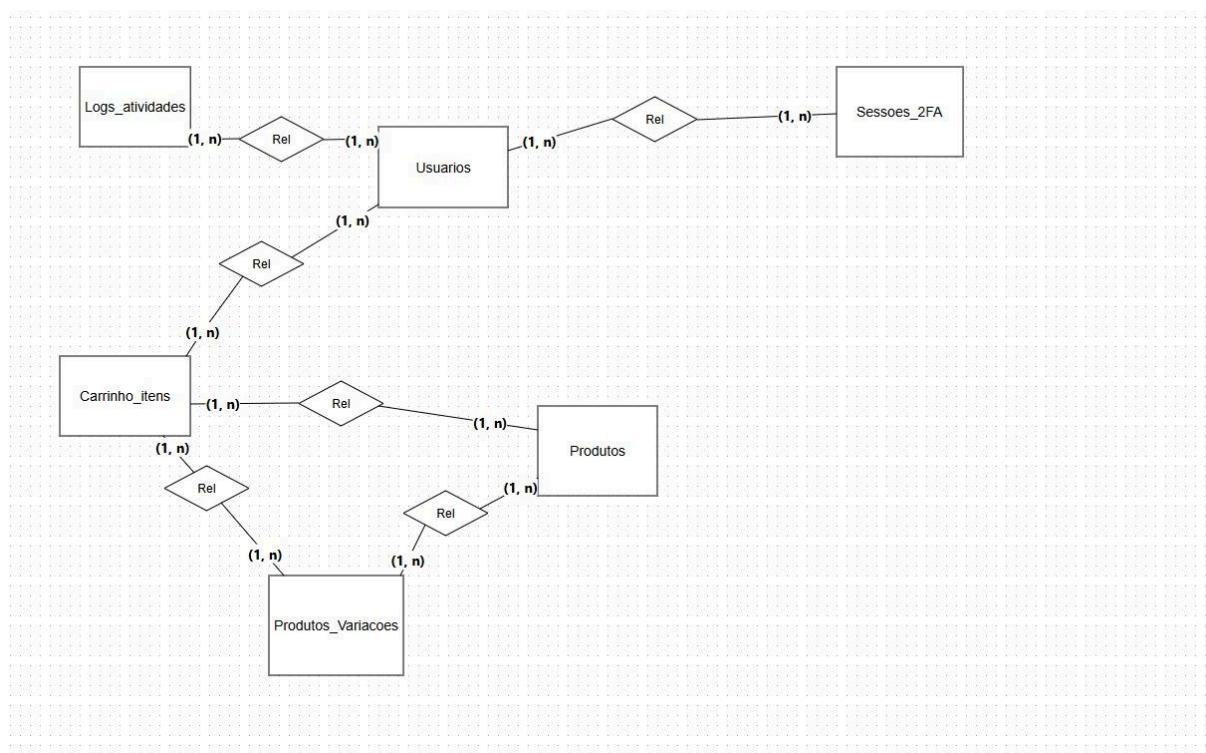
- Finalizar a documentação do projeto, incluindo o modelo ER e o link do GitHub
- Preparar a apresentação do projeto.

4. Teste Realizados

5. Considerações Finais

6. Anexos

6.1. Anexo A: Modelo DER



6.2. Anexo B: Prints de telas

Monitoramento de Acessos

Voltar ao Painel Sair

Histórico de Entradas e Saídas

Baixar Relatório em PDF

Data/Hora	Usuário	Login	Ação	IP
21/11/2025 22:10:00	Jorge	adriel	Login (2FA)	:1
21/11/2025 22:09:07	teste	teste	Logout	:1
21/11/2025 22:08:54	teste	teste	Login	:1
21/11/2025 22:08:43	teste	teste	Logout	:1
21/11/2025 22:08:30	teste	teste	Login	:1
21/11/2025 22:07:26	Jorge	adriel	Logout	:1
21/11/2025 15:25:00	Jorge	adriel	Login (2FA)	:1
21/11/2025 15:23:46	Jorge	adriel	Logout	:1
21/11/2025 15:21:58	Jorge	adriel	Login (2FA)	:1
21/11/2025 15:21:39	Jorge	adriel	Logout	:1
21/11/2025 15:00:19	Jorge	adriel	Login (2FA)	:1
21/11/2025 14:59:03	Jorge	adriel	Logout	:1
21/11/2025 14:57:46	Jorge	adriel	Login (2FA)	:1

Painel Master

Voltar ao Site Sair

Gerenciador de Produtos

Ver Logs Gerenciar Usuários Novo Produto

Imagen	Nome	Categoria	Preço Base	Ações
	Água 1,5L	Bebida	R\$ 5,00	<button>Editar</button> <button>Excluir</button>
	Água com gás	Bebida	R\$ 4,00	<button>Editar</button> <button>Excluir</button>
	Água sem gás	Bebida	R\$ 3,00	<button>Editar</button> <button>Excluir</button>
	Coca-Cola 1,5L	Bebida	R\$ 10,00	<button>Editar</button> <button>Excluir</button>
	Coca-Cola Zero 1,5L	Bebida	R\$ 10,00	<button>Editar</button> <button>Excluir</button>
	Del Valle Manga	Bebida	R\$ 7,00	<button>Editar</button> <button>Excluir</button>
	Del Valle Maracujá	Bebida	R\$ 7,00	<button>Editar</button> <button>Excluir</button>
	Del Valle Pêssego	Bebida	R\$ 7,00	<button>Editar</button> <button>Excluir</button>
	Del Valle Uva	Bebida	R\$ 7,00	<button>Editar</button> <button>Excluir</button>

The screenshot shows a product catalog titled "Bebida" (Drink) with four items listed:

- Água 1,5L (Water 1,5L) with a "COMPRAR" (Buy) button and a water bottle image.
- Água com gás (Carbonated Water) with a "COMPRAR" button and a water bottle image.
- Água sem gás (Non-carbonated Water) with a "COMPRAR" button and a water bottle image.
- Coca-Cola 1,5L (Coca-Cola 1,5L) with a "COMPRAR" button and a Coca-Cola bottle image.

The screenshot shows a table titled "Todos os Usuários Registrados" (All Registered Users) with the following data:

ID	Nome	Email	Login	CPF	Celular	Perfil	Ações
1	Admin Master	admin@tocadasbebidas.com	admin	000.000.000-00	(00) 00000-0000	master	<button>Editar</button> <button>Excluir</button>
3	Jorge	123@hotmail.com	adriel	123.456.789-10	(21) 00000-0000	master	<button>Editar</button> <button>Excluir</button>
4	teste	mocotto@gmail.com	teste	123.456.631-23	(21) 00000-0000	comum	<button>Editar</button> <button>Excluir</button>

The screenshot shows a modal window titled "Sua Conta" (Your Account) with the message "Segurança Ativa ✓" (Two-factor authentication active). It states: "A autenticação de dois fatores está ATIVADA. Sua conta está protegida com perguntas de segurança (Mae, Nascimento, CEP)." (Two-factor authentication is active. Your account is protected with security questions (Mother, Birthdate, ZIP Code)). There are "Desativar 2FA" (Disable 2FA) and "Sair" (Logout) buttons.

6.3. Anexo C: Tabelas

```

CREATE DATABASE IF NOT EXISTS toca_das_bebidas
CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

USE toca_das_bebidas;

CREATE TABLE IF NOT EXISTS usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    nome VARCHAR(255) NOT NULL,
    data_nascimento DATE NOT NULL,
    sexo ENUM('masculino', 'feminino', 'outro') NOT NULL,
    nome_materno VARCHAR(255) NOT NULL,
    cpf VARCHAR(14) NOT NULL UNIQUE,
    celular VARCHAR(15) NOT NULL,
    telefone_fixo VARCHAR(14) NULL,
    endereco TEXT NOT NULL,
    login VARCHAR(100) NOT NULL UNIQUE,
    senha_hash VARCHAR(255) NOT NULL,
    perfil ENUM('comum', 'master') NOT NULL DEFAULT
    'comum',
    autenticacao_2fa BOOLEAN NOT NULL DEFAULT FALSE, --
Coluna nova
    cep VARCHAR(10) NULL, -- Coluna nova
    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ;

CREATE TABLE IF NOT EXISTS sessoes_2fa (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    token VARCHAR(64) NOT NULL UNIQUE,
    pergunta_tipo ENUM('mae', 'nascimento', 'cep') NOT
NULL,
    tentativas TINYINT DEFAULT 0,
    expires_at DATETIME NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id) ON
DELETE CASCADE
) ;

CREATE TABLE IF NOT EXISTS produtos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,

```

```

descricao TEXT,
preco DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
categoria VARCHAR(50),
imagem_url VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS produto_variacoes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    produto_id INT NOT NULL,
    descricao VARCHAR(100) NOT NULL,
    preco DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (produto_id) REFERENCES produtos(id) ON
DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS carrinho_itens (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    produto_id INT NOT NULL,
    variacao_id INT NULL,
    quantidade INT NOT NULL,
    preco_unitario_salvo DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON
DELETE CASCADE,
    FOREIGN KEY (produto_id) REFERENCES produtos(id) ON
DELETE CASCADE,
    FOREIGN KEY (variacao_id) REFERENCES
produto_variacoes(id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS logs_autenticacao (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ip_address VARCHAR(45),
    sucesso BOOLEAN,
    tipo_autenticacao VARCHAR(20) DEFAULT 'login', --
Adicionado para diferenciar login normal de 2FA
    tentativas_2fa INT DEFAULT 0, -- Adicionado para
logar tentativas
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id) ON
DELETE SET NULL
);

```

```

CREATE TABLE IF NOT EXISTS logs_atividades (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    acao VARCHAR(50) NOT NULL, -- Ex: 'Login', 'Logout'
    data_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ip VARCHAR(45),
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON
DELETE CASCADE
);

-- INSERÇÃO DE DADOS PADRÃO --

INSERT INTO usuarios
    (email, nome, data_nascimento, sexo, nome_materno,
    cpf, celular, endereco, login, senha_hash, perfil)
VALUES
(
    'admin@tocadasbebidas.com',
    'Admin Master',
    '2000-01-01',
    'outro',
    'admin',
    '000.000.000-00',
    '(00) 00000-0000',
    'Toca das Bebidas',
    'admin',
    '$2y$10$S.G7QzY.t9J0m8W.V8/1BuwP.EiY7G.gJtO.S4t.Zg4w/8K.X1uS.'
, -- Senha 'Admin123'
    'master'
);

-- CERVEJA
INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Heineken', 'Cerveja', 0.00,
'assets/img/Brejas/heineken.jpg');
SET @last_id = LAST_INSERT_ID();
INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 6.00), (@last_id, '6
unidades', 35.00), (@last_id, '12 unidades', 60.00);

INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Stella Artois', 'Cerveja', 0.00,
'assets/img/Brejas/stella artois.jpg');

```

```

        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 4.50), (@last_id, '6
unidades', 27.00), (@last_id, '12 unidades', 48.00);

        INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Budweiser', 'Cerveja', 0.00,
'assets/img/Brejas/budweiser.jpg');
        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 1.99), (@last_id, '6
unidades', 10.00), (@last_id, '12 unidades', 20.00);

        INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Corona', 'Cerveja', 0.00,
'assets/img/Brejas/corona.jpg');
        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 4.00), (@last_id, '6
unidades', 24.00), (@last_id, '12 unidades', 44.00);

        INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Imperio', 'Cerveja', 0.00,
'assets/img/Brejas/imperio.jpg');
        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 4.00), (@last_id, '6
unidades', 24.00), (@last_id, '12 unidades', 44.00);

        INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Becks', 'Cerveja', 0.00,
'assets/img/Brejas/becks.jpg');
        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 6.00), (@last_id, '6
unidades', 35.00), (@last_id, '12 unidades', 60.00);

-- CATEGORIA: LICOR
        INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('43', 'Licor', 0.00, 'assets/img/LICOR/43.webp');
        SET @last_id = LAST_INSERT_ID();
        INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 118.00), (@last_id, '3
unidades', 340.00), (@last_id, '5 unidades', 531.00);

```

```

    INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Cointreau', 'Licor', 0.00,
'assets/img/LICOR/cointreau.webp');
    SET @last_id = LAST_INSERT_ID();
    INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 125.00), (@last_id, '3
unidades', 348.00), (@last_id, '5 unidades', 548.00);

    INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Amarula', 'Licor', 0.00,
'assets/img/LICOR/amarula.webp');
    SET @last_id = LAST_INSERT_ID();
    INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 129.00), (@last_id, '3
unidades', 355.00), (@last_id, '5 unidades', 565.00);

    INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Ballena Morango', 'Licor', 0.00,
'assets/img/LICOR/BallenaMorango.webp');
    SET @last_id = LAST_INSERT_ID();
    INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 175.00), (@last_id, '3
unidades', 380.00), (@last_id, '5 unidades', 610.00);

    INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Amarula Coffe', 'Licor', 0.00,
'assets/img/LICOR/amarulaCoffe.webp');
    SET @last_id = LAST_INSERT_ID();
    INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, '1 unidade', 129.00), (@last_id, '3
unidades', 355.00), (@last_id, '5 unidades', 565.00);

    INSERT INTO produtos (nome, categoria, preco, imagem_url)
VALUES ('Amarula Raspberry', 'Licor', 0.00,
'assets/img/LICOR/amarulaRaspberry.webp');
    SET @last_id = LAST_INSERT_ID();
    INSERT INTO produto_variacoes (produto_id, descricao,
preco) VALUES (@last_id, 'P', 129.00), (@last_id, 'M',
355.00), (@last_id, 'G', 560.00);

-- WHISKY
    INSERT INTO produtos (nome, preco, categoria, imagem_url)
VALUES

```

```

        ('Old Parr', 143.00, 'Whisky', 'assets/img/Whisky/Old
Parr.webp'),
        ('Johnnie walker Black', 158.00, 'Whisky',
'assets/img/Whisky/Johnnie walker Black.webp'),
        ('Jack Daniels', 138.00, 'Whisky',
'assets/img/Whisky/Jack Daniels.webp');

-- BEBIDA
INSERT INTO produtos (nome, preco, categoria, imagem_url)
VALUES
        ('Coca-Cola 1,5L', 10.00, 'Bebida',
'assets/img/bebidas/coca 1,5L.png'),
        ('Coca-Cola Zero 1,5L', 10.00, 'Bebida',
'assets/img/bebidas/coca 1,5L 0.png'),
        ('Guaraná Antarctica 1,5L', 9.00, 'Bebida',
'assets/img/bebidas/guarana 1,5L.png'),
        ('Guaraná Zero 1,5L', 9.00, 'Bebida',
'assets/img/bebidas/guarana 1,5L 0.png'),
        ('Del Valle Uva', 7.00, 'Bebida', 'assets/img/bebidas/del
valle uva.png'),
        ('Del Valle Pêssego', 7.00, 'Bebida',
'assets/img/bebidas/del valle pessego.png'),
        ('Del Valle Manga', 7.00, 'Bebida',
'assets/img/bebidas/del valle manga.png'),
        ('Del Valle Maracujá', 7.00, 'Bebida',
'assets/img/bebidas/del valle maracuja.png'),
        ('Tônica', 5.00, 'Bebida',
'assets/img/bebidas/tonica.png'),
        ('Schweppes', 5.00, 'Bebida',
'assets/img/bebidas/schweppes.png'),
        ('H2O', 6.00, 'Bebida', 'assets/img/bebidas/h20.png'),
        ('Limoneto', 6.00, 'Bebida',
'assets/img/bebidas/limoneto.png'),
        ('Água com gás', 4.00, 'Bebida', 'assets/img/bebidas/agua
c gas.png'),
        ('Água sem gás', 3.00, 'Bebida', 'assets/img/bebidas/agua
s gas.png'),
        ('Guaracamp', 2.00, 'Bebida',
'assets/img/bebidas/guaracamp.png'),
        ('Água 1.5L', 5.00, 'Bebida', 'assets/img/bebidas/agua
1.5L.png');

```