
MODELO LÓGICO DE DADOS - MLD

IDENTIFICAÇÃO DO PROJETO

Projeto

- Máquina de Café - Devine Café

Disciplina do Projeto

- Métodos Formais de Engenharia de Software

Equipe do Projeto

- ALLYSON BRUNO DE FREITAS FERNANDES - 2024012632;
- ANDREY DE OLIVEIRA SABINO - 2020010859;
- GEÍSA MORAIS GABRIEL - 2024012594;
- LÍVIA BEATRIZ MAIA DE LIMA - 2024012596;
- PEDRO DAMIÃO DE OLIVEIRA LUZ - 2021022519.
- KLEBSON DAVI DE SOUZA MAGALHÃES - 2022011458;

HISTÓRICO DE REGISTROS

Versão	Data	Autor	Descrição
{1.0}	19/04/2025	Klebson Davi	Elaboração do documento

SUMÁRIO

1. MODELO LÓGICO DE DADOS DO DEVINE CAFÉ.....	3
2. RESUMO.....	3
3. PESSOAS ENVOLVIDAS.....	3
4. METODOLOGIAS.....	3
5. MODELO LÓGICO DE DADOS.....	5

1. MODELO LÓGICO DE DADOS DO DEVINE CAFÉ

Este documento tem como objetivo descrever o modelo lógico dos dados do Devine Café. Ele descreve a estrutura dos dados de forma rigorosa, mas ainda independente de tecnologias específicas.

2. RESUMO

Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui Digite seu texto aqui.

A realização da especificação formal, foi desenvolvida baseada nos requisitos funcionais que estão registrados na documentação do sistema.

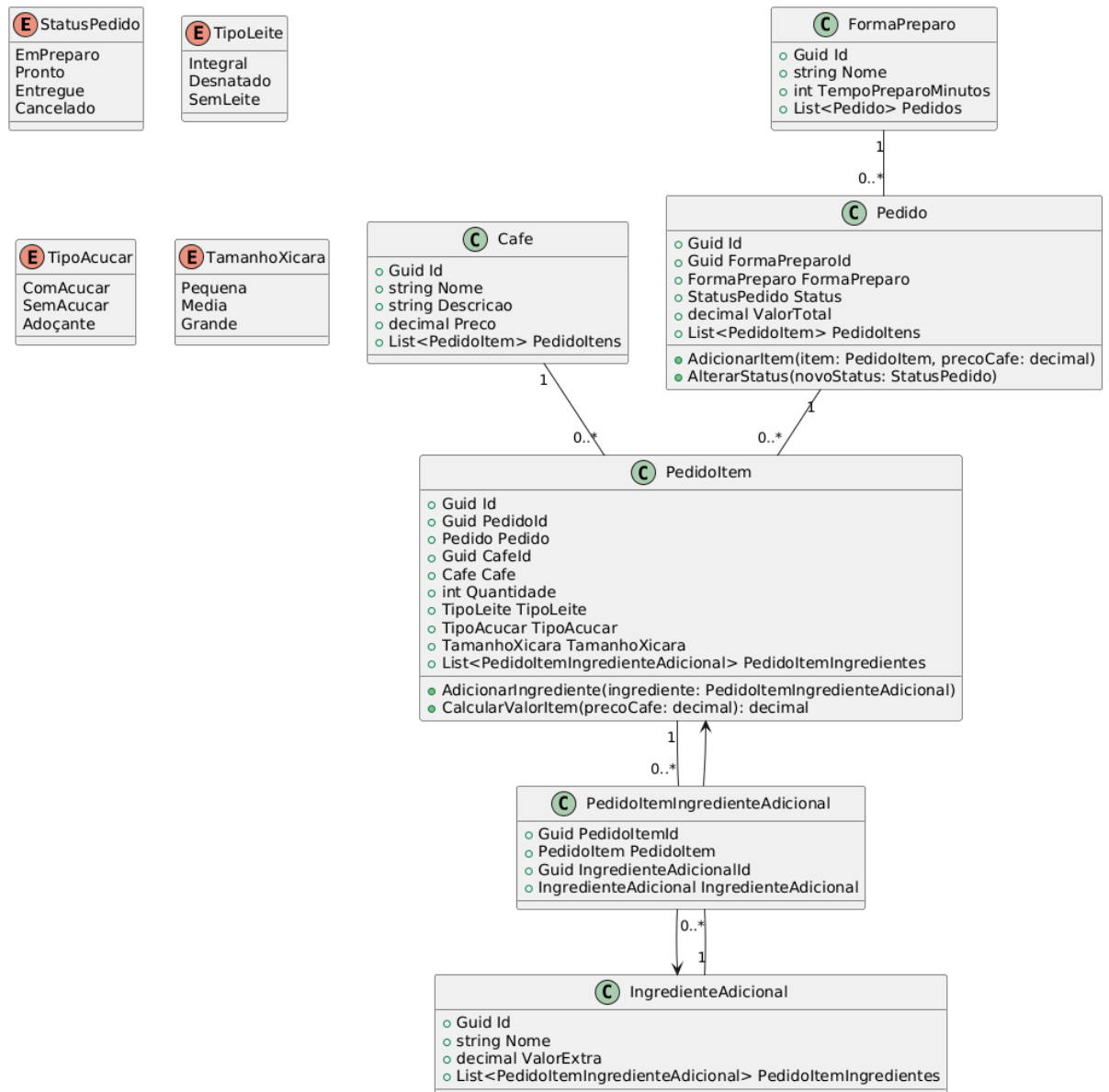
3. PESSOAS ENVOLVIDAS

As pessoas que tiveram envolvimento no desenvolvimento do modelo estão elencadas abaixo:

- ALLYSON BRUNO DE FREITAS FERNANDES
- GEISA MORAIS GABRIEL
- KLEBSON DAVI DE SOUZA MAGALHAES

4. METODOLOGIAS

O modelo lógico é desenvolvido com base em um diagrama conceitual, que representa os requisitos de dados de forma abstrata antes da implementação. Dessa forma, utilizamos o diagrama de classes da aplicação para podermos desenvolver o modelo lógico de dados. Nosso diagrama de classes aparece na imagem abaixo.



Antes de apresentar o que serviu de norte para o desenvolvimento, nesse sentido o modelo lógico define:

- Entidades → Tabelas.
- Atributos → Colunas.
- Relacionamentos → Chaves primárias (PK) e estrangeiras (FK).
- Restrições → Regras de integridade (domínios, nulidade, unicidade).

Acompanhemos o exemplo abaixo:

Elemento	Descrição	Exemplo (SQL)
Tabelas	Estruturas que armazenam entidades.	<code>CLIENTE (id, nome, email)</code>
Chave Primária	Identificador único de uma linha.	<code>id INT PRIMARY KEY</code>
Chave Estrangeira	Referência a uma chave primária em outra tabela.	<code>cliente_id INT REFERENCES CLIENTE(id)</code>
Domínios	Tipos de dados e restrições dos atributos.	<code>email VARCHAR(255) NOT NULL UNIQUE</code>
Normalização	Eliminação de redundâncias (1NF, 2NF, 3NF, etc.).	Tabelas divididas para evitar repetição.

O objetivo é traduzir o modelo conceitual para uma estrutura implementável, garantindo consistência através de regras de integridade e otimizar consultas sem vincular a um SGBD específico.

5. MODELO LÓGICO DE DADOS

Primeiro, criamos tabelas auxiliares para simular os enums:

```
CREATE TABLE StatusPedido (  
    nome VARCHAR(20) PRIMARY KEY  
);
```

```
CREATE TABLE TipoLeite (  
    nome VARCHAR(20) PRIMARY KEY  
);
```

```
CREATE TABLE TipoAcucar (  
    nome VARCHAR(20) PRIMARY KEY  
);
```

```
CREATE TABLE TamanhoXicara (  
    nome VARCHAR(20) PRIMARY KEY  
);
```

Em seguida, as principais:

```
CREATE TABLE FormaPreparo (  
    id UUID PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    tempo_preparo_minutos INT NOT NULL  
);
```

```
CREATE TABLE Cafe (  
    id UUID PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    descricao TEXT,  
    preco DECIMAL(10,2) NOT NULL  
);
```

```
CREATE TABLE Pedido (  
    id UUID PRIMARY KEY,  
    forma_preparo_id UUID NOT NULL,  
    status VARCHAR(20) NOT NULL,  
    valor_total DECIMAL(10,2),  
    FOREIGN KEY (forma_preparo_id) REFERENCES FormaPreparo(id),  
    FOREIGN KEY (status) REFERENCES StatusPedido(nome)  
);
```

```
CREATE TABLE Pedidoltem (  
    id UUID PRIMARY KEY,  
    pedido_id UUID NOT NULL,  
    cafe_id UUID NOT NULL,  
    quantidade INT NOT NULL,  
    tipo_leite VARCHAR(20) NOT NULL,  
    tipo_acucar VARCHAR(20) NOT NULL,  
    tamanho_xicara VARCHAR(20) NOT NULL,  
    FOREIGN KEY (pedido_id) REFERENCES Pedido(id),  
    FOREIGN KEY (cafe_id) REFERENCES Cafe(id),  
    FOREIGN KEY (tipo_leite) REFERENCES TipoLeite(nome),  
    FOREIGN KEY (tipo_acucar) REFERENCES TipoAcucar(nome),  
    FOREIGN KEY (tamanho_xicara) REFERENCES TamanhoXicara(nome)  
);
```

```
CREATE TABLE IngredienteAdicional (  
  id UUID PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  valor_extra DECIMAL(10,2) NOT NULL  
);  
  
CREATE TABLE PedidoItemIngredienteAdicional (  
  pedido_item_id UUID NOT NULL,  
  ingrediente_adicional_id UUID NOT NULL,  
  PRIMARY KEY (pedido_item_id, ingrediente_adicional_id),  
  FOREIGN KEY (pedido_item_id) REFERENCES PedidoItem(id),  
  FOREIGN KEY (ingrediente_adicional_id) REFERENCES IngredienteAdicional(id)  
);
```

Métodos como AdicionarIngrediente e CalcularValorItem não aparecem no modelo lógico porque são implementações da lógica de negócio, não dados.