

euComida

30/09/2025

1. Identificação

Nome	RGA	Turma
<u>Arthur Fernandes Rabuske</u>	<u>202011722002</u>	<u>CC</u>

2. Resumo

O euComida é um sistema de delivery de comida que visa operar de forma eficiente como intermediário entre consumidores e restaurantes. O sistema se concentra em dois atores principais: consumidores e restaurantes.

- **Consumidores:** São usuários que compram comida através da plataforma.
- **Restaurantes:** São parceiros que oferecem seus produtos na plataforma.

O sistema permite que os consumidores visualizem os produtos oferecidos pelos restaurantes, façam pedidos e acompanhem o status de suas compras. O modelo de negócio da empresa se baseia na cobrança de uma taxa de 3% sobre cada venda realizada pelos restaurantes. Essa taxa é cobrada no momento do pagamento do consumidor, proporcionando uma fonte de receita constante para a empresa. O sistema, desenvolvido com base no MySQL, terá duas interfaces: uma para os consumidores realizarem suas compras e outra para os restaurantes gerenciarem seus estabelecimentos e cardápios.

3. Modelo entidade relacionamento

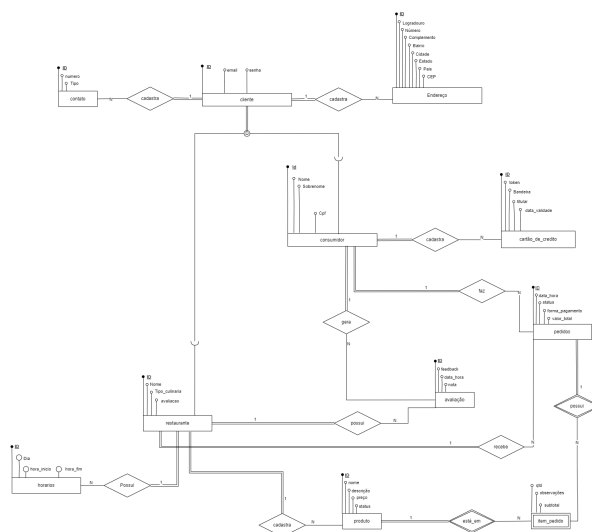


FIGURA 1 - Diagrama entidade relacionamento

- A entidade cliente trata-se da modelagem do usuário do sistema (seja consumidores ou restaurantes). Ela possui ID como chave primária, email e senha. O cliente pode ter zero ou muitos endereços e zero ou muitos contatos.
- A entidade consumidor é uma especialização de cliente. Ela possui CPF, nome e sobrenome como atributos. Um consumidor pode cadastrar zero ou muitos cartões para pagamento. Um consumidor pode gerar zero ou muitas avaliações sobre um restaurante. Um consumidor pode realizar zero ou muitos pedidos.
- A entidade endereço possui ID como chave primária, Logradouro (Ex: Rua, Avenida, etc.), Número, Complemento, Bairro, Cidade, Estado, País, CEP. Um endereço só pode ser cadastrado por um cliente.
- A entidade de contato possui ID como chave primária, número e tipo (ex: Fixo, Móvel..). Um contato só pode ser cadastrado por um cliente.
- A entidade cartão de crédito possui ID como chave primária, um token para substituir informações sensíveis (Ex: Número e CVV), Nome do titular, bandeira (Ex: Visa, Mastercard) e data de validade. Um cartão só pode estar associado a um e somente a um consumidor.
- A entidade restaurante é uma especialização de cliente. Ela possui nome, tipo de culinária e avaliação para armazenar a média das avaliações recebidas. Um restaurante pode registrar muitos horários de funcionamento. Um restaurante pode ter muitas avaliações. Um restaurante pode cadastrar muitos produtos. Um restaurante pode receber muitos pedidos.
- A entidade horarios armazena a hora de funcionamento dos restaurantes. Ela possui ID como chave primária, dia da semana, horário de início e horário de fim do expediente. Um horário só pode ser armazenado por um cliente.
- A entidade de avaliação possui um ID como chave primária, um feedback textual, uma data e hora, e uma nota de 0 a 5. Uma avaliação pode ser gerada por um e somente um consumidor. Uma avaliação deve estar associada a um e somente um restaurante.
- A entidade produtos possui ID, nome, descrição, preço unitário, status (disponível/ indisponível). Um prato pode ser registrado por uma e somente um restaurante. Um prato pode estar em 0 ou muitos itens pedidos.

- A entidade pedidos possui ID, data e hora, status (preparação, trânsito, entregue), forma de pagamento e valor total. Um pedido só pode ser realizado por um e somente por um consumidor. Um pedido está relacionado a uma e somente um restaurante. Um pedido pode ter 0 ou muitos itens.
- A entidade fraca item_pedido possui quantidade, uma observação inserida pelo consumidor e o subtotal. Um item pedido está associado a um e somente um pedido. Um item pedido tem referência a um e somente um prato.

4. Modelo relacional

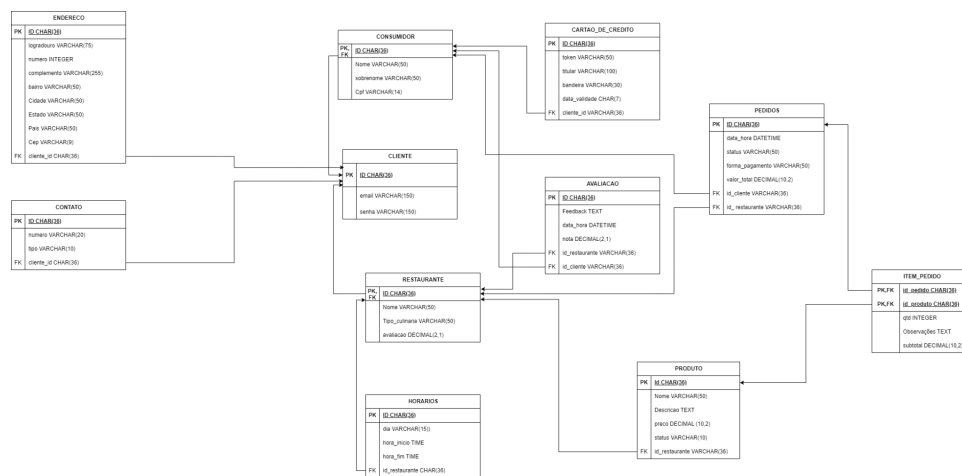


FIGURA 2 - Modelo relacional

Entidade	Função	Chave primária	Chave estrangeira	Cardinalidade
Cliente	Armazena os dados comuns a todos os usuários do sistema	ID	Sem	Sem
Consumidor	Armazena os dados dos consumidores	ID(Referencia vinda da tabela cliente)	ID (Referencia vinda da tabela cliente)	Um consumidor está associado a um cliente. Um cliente só pode estar associado a zero ou um consumidor

Restaurante	Armazena os dados dos restaurantes	ID (Referencia vinda da tabela cliente)	ID (Referencia vinda da tabela cliente)	Um restaurante está associado a um cliente. Um cliente só pode estar associado a zero ou um restaurante
Endereço	Armazena os dados de endereços	ID	cliente_id (Referencia vinda da tabela cliente)	Um endereço pode ser registrado por apenas um cliente. Um cliente pode registrar muitos endereços
Cartão de crédito	Armazenar os dados dos cartões de crédito dos consumidores.	ID	consumidor_id (faz referência a ID em consumidor)	Um cartão de crédito só pode ser registrado por um consumidor. Um consumidor pode registrar muitos cartões de crédito
contato	Armazena informações de contato	ID	cliente_id (Faz referência ao ID na tabela cliente)	Um contato está associado a um cliente. Um cliente só pode estar associado a zero ou um contato
Produto	Armazenar os dados dos produtos oferecidos pelos restaurantes.	ID	id_restaurante (Faz referência a ID em restaurante).	Um produto está associado a um restaurante. Um restaurante pode registrar zero ou muitos produtos.
Pedidos	Armazenar os pedidos realizados pelos consumidores.	ID	id_consumidor (faz referência a ID em consumidor); id_restaurante (faz referência a	Um consumidor pode fazer vários pedidos, e um restaurante pode receber vários pedidos

			ID em restaurante)	
Item pedido	Relaciona os produtos aos pedidos.	id_pedido (referência a ID em pedido) e id_produto (referência a ID em produto)	id_pedido (referência a ID em pedidos) e id_produto (referência a ID em produto)	Um pedido pode conter múltiplos produtos e um produto pode estar em múltiplos pedidos
Avaliação	Armazenar avaliações dos consumidores sobre os restaurantes.	ID	id_consumidor (referência a ID em consumidor) e. id_restaurante (referência a ID em restaurante).	Um consumidor pode fazer várias avaliações e um restaurante pode receber várias avaliações

5. Requisitos funcionais

consumidores:

- O consumidor pode se cadastrar na plataforma.s
- O consumidor pode realizar login na plataforma.
- O consumidor pode registrar endereços de entrega. s
- O consumidor pode editar seus endereços de entrega. n
- O consumidor pode registrar contatos. s
- O consumidor pode editar seus contatos. n
- O consumidor pode registrar cartão de crédito como método de pagamento. n
- O consumidor pode editar seu cartão inserido. n
- O consumidor pode visualizar todos os restaurantes cadastrados na plataforma. n
- O consumidor pode visualizar os produtos oferecidos em um restaurante. n
- O consumidor pode realizar pedidos para os restaurantes. n
- O consumidor pode visualizar seus pedidos realizados. n
- O consumidor pode realizar o pagamento de um pedido. n
- O consumidor pode avaliar um restaurante. n

Restaurantes:

- Os restaurantes podem se cadastrar na plataforma. s
- Os restaurantes podem fazer login na plataforma. s
- Os restaurantes podem cadastrar produtos no seu cardápio. s
- Os restaurantes podem editar seus produtos. s
- Os restaurantes podem receber pedidos. s
- Os restaurantes podem visualizar os pedidos recebidos. s
- Os restaurantes podem receber avaliações do consumidor. s
- O restaurante pode registrar um endereço. s
- Os restaurantes podem editar o endereço. n
- Os restaurantes podem registrar contatos. s
- Os restaurantes podem editar os contatos; n
- Os restaurantes podem adicionar seus horários de funcionamento; s

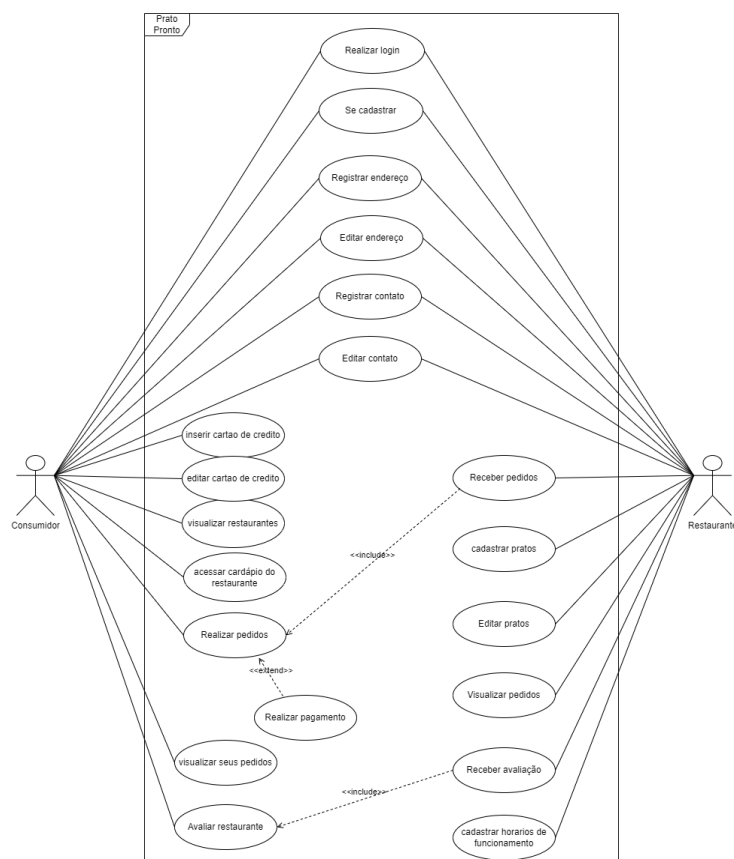


FIGURA 3 - Diagrama de casos de uso

6. Procedimentos armazenados e gatilhos

Esta aplicação utiliza procedimentos armazenados e gatilhos para maximizar a automação dos processos em mySql. Abaixo estão as situações onde tais tecnologias são necessárias:

Procedimentos armazenados:

- **Calcula a nota média de um restaurante:** Procedimento armazenado que calcule a média das notas recebidas de um restaurante e armazene o valor na tabela.
- **Calcular o valor total do pedido:** Procedimento que dado um pedido calcule o valor total e armazene na tabela.
- **Calcular subtotais de um item pedido:** Procedimento que calcule o subtotal de um item pedido e armazene na tabela.

Gatilhos:

- Gatilho na tabela **item_pedido** que atualize preço total na tabela pedido toda vez que for adicionado ou removido uma tupla na tabela.
- Gatilho na tabela **avaliações** para calcular a nota média de um restaurante toda vez que for adicionado ou removido uma tupla na tabela.
- Gatilho na tabela **item_pedido** para que não de para adicionar em um pedido um item que estiver inativo.

Front End

O front-end da aplicação é desenvolvido utilizando o framework de desenvolvimento web Bootstrap. A aplicação é dividida em dois módulos, consumidor e restaurante:

- O módulo de consumidor permite aos usuários navegar por restaurantes, visualizar cardápios e fazer pedidos.
- O módulo de restaurante é voltado para a gestão, permitindo aos restaurantes cadastrarem seus produtos, gerenciarem pedidos.

Back End

O backend da aplicação é desenvolvido com Flask. Para o Banco de dados foi utilizado MySQL.